Information Security Management Handbook

Fifth Edition

Edited by
Harold F. Tipton, CISSP
Micki Krause, CISSP

AUERBACH

Information Security Management Handbook

Fifth Edition

OTHER INFORMATION SECURITY BOOKS FROM AUERBACH

Asset Protection and Security Management Handbook
POA Publishing
ISBN: 0-8493-1603-0

Building a Global Information Assurance Program
Raymond J. Curts and Douglas E. Campbell
ISBN: 0-8493-1368-6

Building an Information Security Awareness Program
Mark B. Desman
ISBN: 0-8493-0116-5

Critical Incident Management
Alan B. Sterneckert
ISBN: 0-8493-0010-X

Cyber Crime Investigator's Field Guide
Bruce Middleton
ISBN: 0-8493-1192-6

Cyber Forensics: A Field Manual for Collecting, Examining, and Preserving Evidence of Computer Crimes
Albert J. Marcella, Jr. and Robert S. Greenfield
ISBN: 0-8493-0955-7

The Ethical Hack: A Framework for Business Value Penetration Testing
James S. Tiller
ISBN: 0-8493-1609-X

The Hacker's Handbook: The Strategy Behind Breaking into and Defending Networks
Susan Young and Dave Aitel
ISBN: 0-8493-0888-7

Information Security Architecture: An Integrated Approach to Security in the Organization
Jan Killmeyer Tudor
ISBN: 0-8493-9988-2

Information Security Fundamentals
Thomas R. Peltier
ISBN: 0-8493-1957-9

Information Security Management Handbook, 5th Edition
Harold F. Tipton and Micki Krause
ISBN: 0-8493-1997-8

Information Security Policies, Procedures, and Standards: Guidelines for Effective Information Security Management
Thomas R. Peltier
ISBN: 0-8493-1137-3

Information Security Risk Analysis
Thomas R. Peltier
ISBN: 0-8493-0880-1

Information Technology Control and Audit
Fredrick Gallegos, Daniel Manson, and Sandra Allen-Senft
ISBN: 0-8493-9994-7

Investigator's Guide to Steganography
Gregory Kipper
0-8493-2433-5

Managing a Network Vulnerability Assessment
Thomas Peltier, Justin Peltier, and John A. Blackley
ISBN: 0-8493-1270-1

Network Perimeter Security: Building Defense In-Depth
Cliff Riggs
ISBN: 0-8493-1628-6

The Practical Guide to HIPAA Privacy and Security Compliance
Kevin Beaver and Rebecca Herold
ISBN: 0-8493-1953-6

A Practical Guide to Security Engineering and Information Assurance
Debra S. Herrmann
ISBN: 0-8493-1163-2

The Privacy Papers: Managing Technology, Consumer, Employee and Legislative Actions
Rebecca Herold
ISBN: 0-8493-1248-5

Public Key Infrastructure: Building Trusted Applications and Web Services
John R. Vacca
ISBN: 0-8493-0822-4

Securing and Controlling Cisco Routers
Peter T. Davis
ISBN: 0-8493-1290-6

Strategic Information Security
John Wylder
ISBN: 0-8493-2041-0

Surviving Security: How to Integrate People, Process, and Technology, Second Edition
Amanda Andress
ISBN: 0-8493-2042-9

A Technical Guide to IPSec Virtual Private Networks
James S. Tiller
ISBN: 0-8493-0876-3

Using the Common Criteria for IT Security Evaluation
Debra S. Herrmann
ISBN: 0-8493-1404-6

AUERBACH PUBLICATIONS

www.auerbach-publications.com
To Order Call: 1-800-272-7737 • Fax: 1-800-374-3401
E-mail: orders@crcpress.com

Information Security Management Handbook

Fifth Edition

Edited by
Harold F. Tipton, CISSP
Micki Krause, CISSP

AUERBACH PUBLICATIONS

A CRC Press Company
Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Information security management handbook / Harold F. Tipton, Micki Krause, editors.—5th ed.
p. cm.
Includes bibliographical references and index.
ISBN 0-8493-1997-8 (alk. paper)
1. Computer security—Management—Handbooks, manuals, etc. 2. Data protection—Handbooks, manuals, etc. I. Tipton, Harold F. II. Krause, Micki.

QA76.9.A25I54165 2003
658′.0558—dc22 2003061151

Visit the CRC Press Web site at www.crcpress.com

Auerbach is an imprint of CRC Press LLC

No claim to original U.S. Government works
International Standard Book Number 0-8493-1997-8
Library of Congress Card Number 2003061151
Printed in the United States of America 1 2 3 4 5 6 7 8 9 0
Printed on acid-free paper

© 2004 by CRC Press LLC

Table of Contents

2 TELECOMMUNICATIONS, NETWORK, AND INTERNET SECURITY

Section 2.1 Communications and Network Security

Section 2.2 Internet/Intranet/Extranet

Section 2.3 E-mail Security

Section 2.4 Secure Voice Communications

Section 2.5 Network Attacks and Countermeasures

3 INFORMATION SECURITY MANAGEMENT

Section 3.1 Security Management Concepts and Principles

Section 3.2 Change Control Management

Section 3.3 Data Classification

Section 3.4 Risk Management

Section 3.5 Employment Policies and Practices

Section 3.6 Risk Management

Section 3.7 Security Awareness Training

Section 3.8 Security Management Planning

4 APPLICATION PROGRAM SECURITY

Section 4.1 APPLICATION ISSUES

Section 4.2 Databases and Data Warehousing

Section 4.3 Systems Development Controls

Section 4.4 Malicious Code

Contributors

Thomas Akin, CISSP, has worked in information security for almost a decade. He is the founding director of the Southeast Cybercrime Institute, where he also serves as chairman for the Institute's Board of Advisors. He is an active member of the Georgia Cybercrime Task Force where he heads up the Task Force's Education committee. Thomas also works with Atlanta's ISSA, InfraGard, and HTCIA professional organizations. He has published several articles on Information Security and is the author of *Hardening Cisco Routers.* He developed Kennesaw State University's highly successful UNIX and Cisco training programs and, in addition to his security certifications, is also certified in Solaris, Linux, and AIX; is a Cisco Certified Academic Instructor (CCAI), and is a Certified Network Expert (CNX). He can be reached at takin@kennesaw.edu.

Mandy Andress, CISSP, SSCP, CPA, CISA, is Founder and President of ArcSec Technologies, a security consulting firm specializing in product/technology analysis. Before starting ArcSec Technologies, Mandy worked for Exxon, USA and several Big 5 accounting firms, including Deloitte & Touche and Ernst & Young. After leaving the Big 5, Mandy became Director of Security for Privada, Inc., a privacy start-up in San Jose. At Privada, Mandy helped develop security policies, secure network design, develop Firewall/VPN solutions, increase physical security, secure product design, and periodic network vulnerability testing. Mandy has written numerous security product and technology reviews for various computer trade publications. A member of the Network World Global Test Alliance, she is also a frequent presenter at conferences, including Networld+Interop, Black Hat, and TISC. Mandy holds a BBA in accounting and an MS in MIS from Texas A&M University. She is the author of *Surviving Security, 2nd Edition* (Auerbach Publications, 2003).

Jim Appleyard is a senior security consultant with the IBM Security and Privacy Services consulting practice. With 33 years of technical and management experience in information technology, he specializes in enterprise-wide information security policies and security architecture design. He has specific expertise in developing information security policies, procedures, and standards; conducting business impact analysis; performing enterprisewide security assessments; and designing data classification and security awareness programs.

David W. Baker is a member of the CVE Editorial Board. As a Lead INFOSEC Engineer in MITRE's Security and Information Operations Division, he has experience in deployment and operation of large-scale intrusion detection systems, critical infrastructure protection efforts, and digital forensics research. A member of the American Academy of Forensic Sciences, Baker holds a bachelor's degree from The State University of New York, and a Master of Forensic Science degree from George Washington University.

Dencho N. Batanov is with the school of Advanced Technologies at the Asian Institute of Technology in Pathumthani, Thailand.

John Berti, CISSP, is a Senior Manager in the Winnipeg Office of Deloitte & Touche LLP's Security Services consulting practice. John has extensive experience in information security including E-business security controls, network security reviews, intrusion and penetration testing, risk analysis, policy development, security awareness, and information security assurance programs. John has over 18 years of Information Security experience and is presently a Senior Lead Instructor for $(ISC)^2$, the organization responsible for worldwide CISSP certification of Information Security professionals. John is also an invited lecturer at some of the largest security conferences and has provided expert witness testimony and technical forensic assistance for various

law enforcement agencies in Canada. John also possesses extensive investigative experience in dealing with various information security-related incidents for a large telecommunications company in Manitoba, relating to computer and toll fraud crimes.

Chuck Bianco, FTTR, CISA, CISSP, is an IT Examination Manager for the Office of Thrift Supervision in Dallas, Texas. He has represented his agency on the IT Subcommittee of the FFIEC. Bianco has experienced more than 600 IT examinations, participated in six IT symposia, written OTS' original Disaster Recovery Bulletin, and led the Interagency Symposium resulting in SP–5. He was awarded the FFIEC Outstanding Examiner Award for significant contributions, and received two Department of the Treasury Awards for Outstanding Performance.

Christina M. Bird, Ph.D., CISSP, is a senior security analyst with Counterpane Internet Security in San Jose, California. She has implemented and managed a variety of wide-area-network security technologies, such as firewalls, VPN packages and authentication systems; built and supported Internet-based remote access systems; and developed, implemented, and enforced corporate IS security policies in a variety of environments. Tina is the moderator of the Virtual Private Networks mailing list, and the owner of "VPN Resources on the World Wide Web," a highly regarded vendor neutral source of information about VPN technology. Tina has a BS in physics from Notre Dame and an MS and Ph.D. in astrophysics from the University of Minnesota.

Steven F. Blanding, CIA, CISA, CSP, CFE, CQA, was, when his contributions were written, the Regional Director of Technology for Arthur Andersen, based in Houston, Texas. Steve has 25 years of experience in the areas of financial auditing, systems auditing, quality assurance, information security, and business resumption planning for large corporations in the consulting services, financial services, manufacturing, retail electronics, and defense contract industries. Steve earned a BS in accounting from Virginia Tech and an MS in business information systems from Virginia Commonwealth University.

David Bonewell, CISSP, CISA, is a chief security architect with Teradata, Cincinnati, Ohio.

Kate Borten, CISSP, a nationally recognized expert in health information security and privacy, is president of The Marblehead Group. She has over 20 years at Harvard University teaching hospitals, health centers, and physician practices; as information security head at Massachusetts General Hospital, and Chief Information Security Officer at CareGroup in Boston. She is a frequent speaker at conferences sponsored by AHIMA, AMIA, CHIM, CHIME, CPRI, and HIMSS, and an advisor and contributor to "Briefings on HIPAA."

Dan M. Bowers, CISSP, is a consulting engineer, author, and inventor in the field of security engineering.

Thomas J. Bray, CISSP, is a Principal Security Consultant with SecureImpact. He has more than 13 years of information security experience in banking, information technology, and consulting. Tom can be reached at tjbray@secureimpact.com. SecureImpact is a company dedicated to providing premier security consulting expertise and advice. SecureImpact has created its information and network service offerings to address the growing proliferation of security risks being experienced by small to mid-sized companies. Information about SecureImpact can be obtained by visiting www.secureimpact.com.

Allen Brusewitz, CISSP, CBCP, has more than 30 years of experience in computing in various capacities, including system development, EDP auditing, computer operations, and information security. He has continued his professional career leading consulting teams in cyber-security services with an emphasis on E-commerce security. He also participates in business continuity planning projects and is charged with developing that practice with his current company for delivery to commercial organizations.

Graham Bucholz is a computer security research for the U.S. government in Baltimore, Maryland.

Carl Burney, CISSP, is a Senior Internet Security Analyst with IBM in Salt Lake City, Utah.

Ken Buszta, CISSP, is Chief Information Security Officer for the City of Cincinnati, Ohio, and has more than ten years of IT experience and six years of InfoSec experience. He served in the U.S. Navy's intelligence community before entering the consulting field in 1994. Should you have any questions or comments, he can be reached at Infosecguy@att.net.

James Cannady is a research scientist at Georgia Tech Research Institute. For the past seven years he has focused on developing and implementing innovative approaches to computer security in sensitive networks and systems in military, law enforcement, and commercial environments

Ioana V. Carastan, CISSP, is a manager with Accenture's global security consulting practice. She has written security policies, standards, and processes for clients in a range of industries, including financial services, high-tech, resources, and government

Mark T. Chapman, CISSP, CISM, IAM, is the Director of Information Security Solutions for Omni Tech Corporation in Waukesha, Wisconsin. Mark holds an MS in computer science from the University of Wisconsin, Milwaukee, in the area of cryptography and information security. He has published several papers and has presented research at conferences in the United States, Asia, and Europe. He is the author of several security-related software suites, including the NICETEXT linguistic steganography package available at www.nicet-ext.com. Mark is a member of the executive planning committee for the Eastern Wisconsin Chapter of InfraGard. For questions or comments, contact Mark at mark.chapman@omnitechcorp.com.

Steven Christey is the editor of the CVE List and the chair of the CVE Editorial Board. His operational experience is in vulnerability scanning and incident response. His research interests include automated vulnerability analysis of source code, reverse-engineering of malicious executable code, and responsible vulnerability disclosure practices. He is a Principal INFOSEC Engineer in MITRE's Security and Information Operations Division. He holds a BS in computer science from Hobart College.

Samuel Chun, CISSP, is director for a technology consulting firm in the Washington, D.C., area

Anton Chuvakin, Ph.D., GCIA, GCIH, is a senior security analyst with a major information security company. His areas of InfoSec expertise include intrusion detection, UNIX security, forensics, and honeypots. In his spare time, he maintains his security portal, www.infosecure.org.

Douglas G. Conorich, the Global Solutions Manager for IBM Global Service's Managed Security Services, with over 30 years of experience with computer security holding a variety of technical and management positions, has responsibility for developing new security offerings, ensuring that the current offerings are standardized globally, and oversees training of new members of the MSS team worldwide. Mr. Conorich teaches people how to use the latest vulnerability testing tools to monitor Internet and intranet connections and develop vulnerably assessments suggesting security-related improvements. Mr. Conorich is also actively engaged in the research of bugs and vulnerabilities in computer operating systems and Internet protocols and is involved in the development of customized alerts notifying clients of new potential risks to security. He has presented papers at over 400 conferences, has published numerous computer security-related articles on information security in various magazines and periodicals, and has held associate professor positions at several colleges and universities.

Michael J. Corby, CISSP, is Director of META Group Consulting. He was most recently president of QinetiQ Trusted Information Management and prior to that, vice president of the Netigy Global Security Practice, CIO for Bain & Company, and the Riley Stoker division of Ashland Oil. He has more than 30 years of experience in the information security field and has been a senior executive in several leading IT and security consulting organizations. He was a founding officer of $(ISC)^2$, developer of the CISSP program, and was named the first recipient of the CSI Lifetime Achievement Award. A frequent speaker and prolific author, Corby graduated from WPI in 1972 with a degree in electrical engineering

Kellina M. Craig-Henderson, Ph.D., is an Associate Professor of Social Psychology at Howard University in Washington, D.C. Craig-Henderson's work has been supported by grants from the National Science Foundation and the Center for Human Resource Management at the University of Illinois.

Jeffrey Davis, CISSP, has been working in information security for the past ten years. He is currently a senior manager at Lucent Technologies, involved with intrusion detection, anti-virus, and threat assessment. He holds a bachelor's degree in electrical engineering and a master's degree in computer science from Stevens Institute of Technology

Matthew J. Decker, CISSP, CISA, CBCP, has 17 years of professional experience in information security. He has advised private industry and local government on information security issues for the past six years with International Network Services, Lucent Technologies, and KPMG LLP. Prior to this, he devoted two years to the United States Special Operations Command (USSOCOM) as a contractor for Booz Allen Hamilton, and served nine years with the NSA. He earned a BSEE in 1985 from Florida Atlantic University and an MBA in 1998 from Nova Southeastern University. In 1992, the NSA's Engineering and Physical Science Career Panel awarded him Certified Cryptologic Engineer (CCE) stature. A former president of the ISSA Tampa Bay chapter, he is a member of ISSA and ISACA.

David Deckter, CISSP, a manager with Deloitte & Touche Enterprise Risk Services, has extensive experience in information systems security disciplines, controlled penetration testing, secure operating system, application and internetworking architecture and design, risk and vulnerability assessments, and project management. Deckter has obtained ISC^2 CISSP certification. He has performed numerous network security assessments for emerging technologies and electronic commerce initiatives in the banking, insurance, telecommunications, healthcare, and financial services industries, and has been actively engaged in projects requiring HIPAA security solutions.

Gildas Deograt, CISSP, is a CISSP Common Body of Knowledge (CBK) seminar instructor. He has been working in the IT field for more than ten years, with a focus over the past five years on information security. His experience includes network design and implementation, security policy development and implementation, developing security awareness program, network security architecture, assessment and integration, and also firewall deployment. At present, he is an Information System Security Officer for Total Exploration and Production. Before moving to France, he was the Chief Information Security Officer at TotalFinaElf E&P Indonesia and also a board member of the Information System Security Association (ISSA), Indonesia

Sandeep Dhameja, CISSP, is responsible for implementation, management of data, network security, and information security at Morningstar. With more than ten years of IT experience, including five years in information security, Dhameja has held several executive and consulting positions. He is widely published with the IEEE, International Engineering Consortium (IEC), Society of Automotive Engineers (SAE), and at international conferences.

John Dorf, ARM, is a senior manager in the Actuarial Services Group of Ernst & Young. Specializing in insurance underwriting and risk management consulting, John earned his 19 years of experience as a risk manager at several Fortune 500 financial service and manufacturing firms. Before joining Ernst & Young, John was a senior risk manager at General Electric Capital Corporation. John has also held risk management positions at Witco Corporation, National Westminster Bank, and the American Bureau of Shipping. Prior to becoming a risk manager, John spent seven years as an underwriting manager and senior marine insurance underwriter at AIG and Atlantic Mutual. John holds a MBA with a concentration in risk management from the College of Insurance; a BA in Economics from Lehigh University; and an Associate in Risk Management (ARM) designation from the Insurance Institute of America.

Mark Edmead, CISSP, SSCP, TICSA, is president of MTE Software, Inc. (www.mtesoft.com) and has more than 25 years of experience in software development, product development, and network/information systems

security. Fortune 500 companies have often turned to Mark to help them with projects related to Internet and computer security. Mark previously worked for KPMG Information Risk Management Group and IBM's Privacy and Security Group, where he performed network security assessments, security system reviews, development of security recommendations, and ethical hacking. Other projects included helping companies develop secure and reliable network system architecture for their Web-enabled businesses. Mark was managing editor of the *SANS Digest* (Systems Administration and Network Security) and contributing editor to the *SANS Step-by-Step Windows NT Security Guide*. He is co-author of *Windows NT: Performance, Monitoring and Tuning*, and he developed the *SANS Business Continuity/Disaster Recovery Plan Step-by-Step Guide*.

Carl F. Endorf, CISSP, is a senior security analyst for one of the largest insurance and banking companies in the United States. He has practical experience in forensics, corporate investigations, and Internet security.

Vatcharaporn Esichaikul is with the school of Advanced Technologies at the Asian Institute of Technology in Pathumthani, Thailand.

Jeffrey H. Fenton, CBCP, CISSP, is the corporate IT crisis assurance/mitigation manager and technical lead for IT Risk Management and a senior staff computer system security analyst in the Corporate Information Security Office at Lockheed Martin Corporation. He joined Lockheed Missiles and Space Company in Sunnyvale, California, as a system engineer in 1982 and transferred into its telecommunications group in 1985. Fenton completed a succession of increasingly complex assignments, including project manager for the construction and activation of an earthquake-resistant network center on the Sunnyvale campus in 1992, and group leader for network design and operations from 1993 through 1996. Fenton holds a BA in economics from the University of California, San Diego, an MA in economics and an MS in operations research from Stanford University, and an MBA in telecommunications from Golden Gate University. Fenton is also a Certified Business Continuity Planner (CBCP) and a Certified Information Systems Security Professional (CISSP).

Bryan D. Fish, CISSP, is a security consultant for Lucent Technologies in Dallas, Texas. He holds a BS in Computer Engineering and a Master of Computer Science degree with a focus on internetworking and computer system security, both from Texas A&M University. Professional interests include security programs and policies, and applications of cryptography in network security.

Todd Fitzgerald, CISSP, CISA, is the Systems Security Office for United Government Services, LLC, the nation's largest processor of Medicare hospital claims on behalf of the Centers for Medicare and Medicaid Services (CMS). He has over 24 years of broad-based information technology experience, holding senior IT management positions with Fortune 500 and Global Fortune 250 companies. Todd is a board member of the ISSA–Milwaukee Chapter, co-chair on the HIPAA Collaborative of Wisconsin Security Task Force, participant in the CMS/Gartner Security Best Practices Group, and is a frequent speaker and writer on security issues.

Stephen D. Fried, CISSP, is the Director of Global Information Security at Lucent Technologies, leading the team responsible for protecting Lucent's electronic and information infrastructure. Stephen began his professional career at AT&T in 1985 and has progressed through a wide range of technical and leadership positions in such areas as software development, database design, call center routing, computing research, and information security for AT&T, Avaya, and Lucent Technologies. In more recent history, Stephen has developed the information security program for two Fortune 500 companies, leading the development of security strategy, architecture, and deployment while dealing with such ever-changing topics as policy development, risk assessment, technology development and deployment and security outsourcing. He is a Certified Information Systems Security Professional and is also an instructor with the SANS Institute. Stephen holds a BS in Telecommunications Management and an MS in Computer Science.

Ed Gabrys, CISSP, is a senior systems engineer for Symantec Corporation. He was information security manager for People's Bank in Bridgeport, Connecticut

Brian Geffert, CISSP, CISA, is a senior manager for Deloitte & Touche's Security Services Practice and specializes in information systems controls and solutions. Geffert has worked on the development of HIPAA assessment tools and security services for healthcare industry clients to determine the level of security readiness with Health Insurance Portability and Accountability Act of 1996 (HIPAA) regulations. In addition, he has implemented solutions to assist organizations addressing their HIPAA security readiness issues. Finally, Geffert is a Certified Information Systems Security Professional (CISSP) and a Certified Information Systems Auditor (CISA).

Karen Gibbs is a senior data warehouse architect with Teradata, Dayton, Ohio.

Alex Golod, CISSP, is an infrastructure specialist for EDS in Troy, Michigan.

Robert Gray, Ph.D., is currently Chair of the Quantitative Methods and Computer Information Systems Department at Western New England College and has more than 20 years of academic and management experience in the IT field.

Frandinata Halim, CISSP, MCSE, a senior security consultant at ITPro Citra Indonesia, PT, has ample experience and qualifications in providing clients with managed security services, information system security consulting, secure network deployment, and other services. In addition, he is competent and knowledgeable in the use and hardening of the Windows environment, Cisco security devices, the number of IDSs, firewalls, and others, currently holding certifications such as CISSP from the $(ISC)^2$, CCSP, CCDA, and CCNA from Cisco Systems, and MCSE from Microsoft. He obtained his bachelor's degree in electronic engineering from Trisakti University, Jakarta, and his master's degree in information system management from Bina Nusantara University, Jakarta.

Susan D. Hansche, CISSP, is a senior manager for information system security awareness and training at PEC Solutions, based in Fairfax, Virginia. She has designed numerous training courses on information technology and information systems security for both private-sector and government clients. Susan is co-author of the *Official $(ISC)^2$ Guide to the CISSP Exam.* She can be reached via e-mail at susan.hansche@pec.com.

William T. Harding, Ph.D., is Dean of the College of Business Administration and an associate professor at Texas A & M University, in Corpus Christi.

Chris Hare, CISSP, CISA, is an Information Security and Control Consultant with Nortel Networks in Dallas, Texas. His experience encompasses over sixteen years in the computing industry with key positions ranging from application design, quality assurance, system administration/engineering, network analysis, and security consulting, operations and architecture. His management career, coupled with in-depth technical knowledge, provides the foundation to integrate the intricate risks of technology to the ongoing survival of major corporations. Chris periodically shares his knowledge in speaking engagements, published articles, books, and other publications. He has written a number of articles for *Sys Admin* magazine, ranging from system administration and tutorial articles to management and architecture. Chris is now writing for Auerbach's *Data Security Management, Information Security Management Handbook,* and *Data Communication Management*, and is co-author the *Official $(ISC)^2$ Guide to the CISSP Exam.* Chris has taught information security at Algonquin College (Ottawa, Canada) and was one of the original members of the Advisory Council for this program. He frequently speaks at conferences on UNIX, specialized technology and applications, security, and audit.

Jay Heiser, CISSP, is an analyst with the European headquarters of TruSecure. A seasoned professional with fourteen years of security experience, he has helped secure the infrastructures of both major Swiss banks, leading Internet service providers, manufacturers, and the U.S. Department of Defense. He co-authored *Computer Forensics: Incident Response Essentials,* and is currently writing a new handbook on information security. Since 1999, he has been a columnist for *Information Security* magazine where he also serves on the Editorial Advisory Board. He was the first Security Editor for *Java Developers Journal* and has written for

InfoWorld, Network World, Web Techniques, and *The Handbook of Information Security Management*. In demand in both Europe and America for his entertaining and thought-provoking presentations, Mr. Heiser has an MBA in International Management from the American Graduate School of International Management.

Gilbert Held is an award-winning author and lecturer. Gil is the author of over 40 books and 450 technical articles. Some of Gil's recent book titles include *Building a Wireless Office* and *The ABCs of IP Addressing*, published by Auerbach Publications. Gil can be reached via e-mail at gil_held@yahoo.com.

Foster Henderson, CISSP, MCSE, CRP, CNA, is an information assurance analyst for Analytic Services, Inc. (ANSER). He is currently a member of the Network Operations and Security Branch within the federal government, covering a wide range of IA matters.

Kevin Henry, CISA, CISSP, Director–Program Development for $(ISC)^2$ Institute, is a regular speaker at conferences and training seminars worldwide, with frequent requests to provide in-depth training, foundational and advanced information systems security and audit courses, and detailed presentations and workshops on key issues surrounding the latest issues in the information systems security field. Kevin combines over twenty years experience in telecom and consulting engagements for major government and corporate clients with an interesting and comfortable learning style that enhances the understanding, relevance, and practical applications of the subject matter. Kevin graduated from Red River College as a computer programmer/analyst and has an Advanced Graduate Diploma in Management from Athabasca University, where he is currently enrolled in their MBA program with a focus on information technology. Kevin has also had several articles published in leading trade journals and in the *Handbook of Information Security Management*.

Paul A. Henry, MCP+I, MCSE, CCSA, CFSA, CFSO, CISSP, Vice President of CyberGuard Corporation and an information security expert who has worked in the security field for more than 20 years, has provided analysis and research support on numerous complex network security projects in Asia, the Middle East, and North America, including several multimillion dollar network security projects, such as Saudi Arabia's National Banking System and the DoD Satellite Data Project USA. Henry has given keynote speeches at security seminars and conferences worldwide on topics including DDoS attack risk mitigation, firewall architectures, intrusion methodology, enterprise security, and security policy development. An accomplished author, Henry has also published numerous articles and white papers on firewall architectures, covert channel attacks, distributed denial-of-service (DDoS) attacks, and buffer overruns. Henry has also been interviewed by ZD Net, the *San Francisco Chronicle*, the *Miami Herald*, NBC Nightly News, CNBC Asia, and many other media outlets.

Rebecca Herold, CISSP, CISA, FLMI, is Vice President, Privacy Services and Chief Privacy Officer at DelCreo, Inc. Prior to this, she was chief privacy officer and senior security architect for QinetiQ Trusted Information Management, Inc. (Q-TIM). She has more than 13 years of information security experience. Herold was the editor and contributing author for *The Privacy Papers*, released in December 2001. Most recently she was the co-author of *The Practical Guide to HIPAA Privacy and Security Compliance* (Auerbach, 2004). She has also written numerous magazine and newsletter articles on information security topics and has given many presentations at conferences and seminars. Herold can be reached at rebecca@delcreo.com.

Debra S. Herrmann is the ITT manager of security engineering for the FAA Telecommunications Infrastructure program. Her special expertise is in the specification, design, and assessment of secure mission-critical systems. She is the author of *Using the Common Criteria for IT Security Evaluation* and *A Practical Guide to Security Engineering and Information Assurance*, both from Auerbach Publications.

Steven Hofmeyr, Ph.D., chief scientist and founder of Sana Security, Inc., received a Ph.D. in computer science in 1999 from the University of New Mexico (UNM), focusing on immunological approaches to computer security. During his studies, he spent a year at the Artificial Intelligence Lab at MIT. After finishing his Ph.D., he was a postdoctoral researcher at UNM, and closely associated with the Santa Fe Institute for Complexity Studies. Hofmeyr has authored and co-authored many articles published in conference proceedings and peer-

reviewed journals on computer security, immunology, and adaptive computation. He has served on the program committee for the ACM's New Security Paradigms Workshop, and is currently on the program committee for the Artificial Immune Systems workshop at the IEEE World Congress on Computational Intelligence. He can be reached at steve.hofmeyr@sanasecurity.com.

Daniel D. Houser, CISSP, MBA, e-Biz+, is a senior security engineer with Nationwide Mutual Insurance Company

Joost Houwen, CISSP, CISA, is the security manager for Network Computing Services at BC Hydro. He has a diverse range of IT and information security experience.

Patrick D. Howard, CISSP, a Senior Information Security Consultant for the Titan Corporation, has over 31 years experience in security management and law enforcement. He has been performing security certification and accreditation tasks for over 14 years as both a security manager and a consultant from both government and commercial industry perspectives. He has experience with implementing security C&A with the Department of the Army, Nuclear Regulatory Commission, Department of Agriculture, and Department of Transportation, and has been charged with developing C&A and risk management guidance for organizations such as Bureau of the Public Debt, U.S. Coast Guard, State of California, University of Texas Southwestern Medical School, University of Texas Medical Branch, and corporations including John Hancock, BankBoston, Sprint, eSylvan, and Schering–Plough. He has extensive practical experience in implementing programs and processes based on NIST guidance (FIPS Pub 102, SP 800-18, 800-26, 800-30, 800-37, etc.), OMB Circular A-130, Appendix III, and BS 7799/ISO 17799. He has direct working experience in security plan development for complex systems, sensitivity definition, use of minimum security baselines, risk analysis, vulnerability assessment, controls validation, risk mitigation, and documenting certification and accreditation decisions. Mr. Howard has also developed and presented training on all of these processes. He is the author of *Building and Implementing a Security Certification and Accreditation Program* (Auerbach Publications, 2004).

Javed Ikbal, CISSP, works at a major financial services company as Director, IT Security, where he is involved in security architecture, virus/cyber incident detection and response, policy development, and building custom tools to solve problems. A proponent of open-source security tools, he is a believer in the power of Perl.

Sureerut Inmor is with the school of Advanced Technologies at the Asian Institute of Technology in Pathumthani, Thailand. He can be reached at sureerut_earth@hotmail.com.

Carl B. Jackson, CISSP, is Vice President–Enterprise Continuity Planning for DelCreo, Inc., an enterprise risk management company. He is a Certified Information Systems Security Professional (CISSP) with more than 25 years of experience in the areas of continuity planning, information security, and information technology internal control and quality assurance reviews and audits. Prior to joining DelCreo, Inc., he served in the QinetiQ-TIM Corporation and as a Partner with Ernst & Young, where he was the firm's BCP Service Line Leader. Carl has extensive consulting experience with numerous major organizations in multiple industries, including manufacturing, financial services, transportation, healthcare, technology, pharmaceuticals, retail, aerospace, insurance, and professional sports management. He also has extensive industry business continuity planning experience as an information security practitioner, manager in the field of information security and business continuity planning, and as a university-level instructor. He has written extensively and is a frequent public speaker on all aspects of continuity planning and information security. Carl can be reached at 1+ 936-328-3663 or by e-mail at carl@delcreo.com.

Martin Johnson is senior manager, Information Systems Assurance & Advisory Services, with Ernst & Young LLP.

Sudhanshu Kairab, CISSP, CISA, is an information security consultant with a diverse background, including security consulting, internal auditing, and public accounting across different industries. His recent projects include security assessments and development of security policies and procedures

Ray Kaplan, CISSP, CISA, CISM, Qualified BS7799 Auditor Credentials and CHSP (Certified HIPAA Security Professional), is an information security consultant with Ray Kaplan and Associates in Minneapolis, Minnesota. He has been a consultant and a frequent writer and speaker in information security for over two decades

Christopher King, CISSP, is a security consultant with Greenwich Technology Partners, Chelmsford, Massachusetts.

Walter S. Kobus, Jr., CISSP, is Vice President, Security Consulting Services, with Total Enterprise Security Solutions, LLC. He has over 35 years of experience in information systems with 15 years experience in security, and is a subject matter expert in several areas of information security, including application security, security management practice, certification and accreditation, secure infrastructure, and risk and compliance assessments. As a consultant, he has an extensive background in implementing information security programs in large environments. He has been credited with the development of several commercial software programs in accounting, military deployment, budgeting, marketing, and several IT methodologies in practice today in security and application development.

Bryan T. Koch, CISSP, holds a BS in psychology, Michigan State University. He began his career as an operating systems developer in academic and scientific settings. He has been involved in the field of IT–Security for almost 20 years, starting as an outgrowth of his effort to connect Cray Research to the Internet — he was asked to create (1988) and lead (through 1995) the company's information security program. Since leaving Cray Research, his focus has been the effectiveness of information security programs in high-threat environments such as electronic commerce. Currently he is responsible for the security of RxHub, a healthcare information technology company.

Joe Kovara, CTP and Principal Consultant of Certified Security Solutions, Inc**.,** has more than 25 years in the security and IT industries with extensive experience in all aspects of information security, operating systems and networks, as well as in the development and practical application of new technologies to a wide variety of applications and markets. Joe holds patents on self-configuring computer systems and networks. Prior to joining CSS in 2001, Joe was CTO of CyberSafe Corporation. Joe was a key contributor to CyberSafe's growth to over 250 employees in three countries, including three acquisitions and venture funding of over $100M. He was the prime mover in bringing several enterprise-security products to market and deploying them in mission-critical Fortune 100 environments, with product and services revenues totaling more than $25M. Prior to CyberSafe, Joe was a principal with the security-consulting firm of Kaplan, Kovara & Associates.

Micki Krause, CISSP, has held positions in the information security profession for the past 20 years. She is currently the Chief Information Security Officer at Pacific Life Insurance Company in Newport Beach, California, where she is accountable for directing the Information Protection and Security Program enterprise-wide. Micki has held several leadership roles in industry-influential groups including the Information Systems Security Association (ISSA) and the International Information System Security Certification Consortium (ISC)^2 and is a long-term advocate for professional security education and certification. In 2003, Krause received industry recognition as a recipient of the "Women of Vision" award given by *Information Security* magazine. In 2002, Krause was honored as the second recipient of the Harold F. Tipton Award in recognition of sustained career excellence and outstanding contributions to the profession. She is a reputed speaker, published author, and co-editor of the *Information Security Management Handbook* series.

David C. Krehnke, CISSP, CISM, IAM, is a Principal Information Securit Analyst for Northrop Grumman Information Technology in Raleigh, North Carolina. He has more than 30 years experience in assessment and implementation of information security technology, policy, practices, procedures, and protection mechanisms in support of organizational objectives for various federal agencies and government contractors. Krehnke has also served the (ISC)^2 organization as a board member, vice president, president, and program director responsible for test development.

Mollie E. Krehnke, CISSP, IAM, is a Principal Information Security Analyst for Northrop Grumman Information Technology in Raleigh, North Carolina. She has served as an information security consultant for more than 15 years.

Kelly J. "KJ" Kuchta, CPP, CFE, is President of Forensics Consulting Solutions, in Phoenix. Formerly an area leader for Meta Security Group and Ernst & Young's Computer Forensics Services Group in Phoenix, Arizona. He is an active member of the High Technology Crime Investigation Association (HTCIA), Association of Certified Fraud Examiners (ACFE), Computer Security Institute (CSI), International Association of Financial Crime Investigators Association (IACFCI), and the American Society of Industrial Security (ASIS). He currently serves on the board of the ASIS Information Technology Security Council.

Ross A. Leo, CISSP, an information security professional for over 23 years, with experience in a broad range of enterprises, currently is the Director of Information Systems, and Chief Information Security Officer at the University of Texas Medical Branch/Correctional Managed Care Division in Galveston, Texas. He has worked internationally as a systems analyst and engineer, IT auditor, educator, and security consultant for companies including IBM, St. Luke's Episcopal Hospital, Computer Sciences Corporation, Coopers & Lybrand, and Rockwell International. Recently, he was the Director of IT Security Engineering and Chief Security Architect for Mission Control at the Johnson Space Centre. His professional affiliations include (ISC)^2, ASIS, HCCO, and is a member of the IT Security Curriculum Development and Advisory Board for Texas State Technical College. Mr. Leo attended graduate school at the University of Houston, and undergraduate school at Southern Illinois University. He is the editor of the *HIPAA Program Reference Handbook* (Auerbach Publications, 2004).

Ian Lim, CISSP, a senior consultant in Accenture's global security consulting practice, has defined and deployed security architectures for Fortune 100 companies, as well as contributed to Accenture's Global Privacy and Policy Framework. Ian graduated from the University of California at Irvine with a degree in Information Computer Science and a minor in English

David A. Litzau, CISSP, with a foundation in electronics and audio/visual, moved into the computer sciences in 1994. David has been teaching information security in San Diego for the past six years

David MacLeod, Ph.D., CISSP, is the chief information security officer for The Regence Group, based in Portland, Oregon. He holds a Ph.D. in computer science, has 23 years of experience in information technology, and is accredited by ISC^2 as a CISSP. He is also accredited by the Healthcare Information Management and Systems Society (HIMSS) as a Certified Professional in Healthcare Information Management Systems (CPHIMS). MacLeod has worked in a variety of industries, including government, retail, banking, defense contracting, emerging technologies, biometrics, physical security, and healthcare. He is a member of the organizing committee for the Health Sector Information Sharing and Analysis Center (ISAC), part of the Critical Infrastructure Protection activities ordered by Presidential Decision Directive 63

Franjo Majstor, CISSP, CCIE, is a senior technical consultant with Cisco Systems, Inc., in Brussels, Belgium. He focuses on security products, features, and solutions across technologies and is involved as a trusted adviser in the design of major security networking-related projects in Europe, the Middle East, and Africa.

Robert A. Martin is the leader of Common Vulnerabilities and Exposures (CVE) Compatibility efforts and a member of MITRE's Open Vulnerability Assessment Language (OVAL) team. As a principal engineer in

MITRE's Information Technologies Directorate, his work focuses on the interplay of cyber-security, critical infrastructure protection, and software engineering technologies and practices. A member of the ACM, AFCEA, NDIA, and the IEEE, Martin holds a bachelor's degree and a master's degree in electrical engineering from Rensselaer Polytechnic Institute and an MBA from Babson College.

Bruce R. Matthews, CISSP, has been managing embassy technical security programs for U.S. government facilities worldwide for over 15 years. He is a Security Engineering Officer with the U.S. Department of State, Bureau of Diplomatic Security, and is currently on a three-year exchange program with the British Government. With the British, Bruce is examining a wide range of technical security issues and how they impact on IT security. As part of his work, he also conducts vulnerability assessments, IT security investigations and forensic analysis. In previous assignments, Bruce was head of the Department of State IT security training program and Chairman of the Security Standards Revision Committee for the Overseas Security Policy Board (OSPB). Bruce, who has been published in magazines such as *Information Security* and *State,* is the author of *Video Surveillance and Security Applications: A Manager's Guide to CCTV* (Auerbach Publications, 2004).

George G. McBride, CISSP, is the Senior Manager of Lucent Technologies' Global Risk Assessment and Penetration Testing group in Holmdel, New Jersey, and has worked in the network security industry for more than six years. George has spoken at conferences worldwide on topics such as penetration testing, risk assessments, and open source security tools. He has consulted to numerous Fortune 100 companies on projects including network architecture, application vulnerability assessments, and security organization development. George has a Bachelor's degree in electronic engineering and a master's degree in software engineering.

Samuel C. McClintock is a Principal Security Consultant with Litton PRC, Raleigh, North Carolina

Lowell Bruce McCulley, CISSP, has more than 30 years of professional experience in the information systems industry. His security credentials are complemented by an extensive background in systems development engineering, primarily focused on critical systems, along with experience in production operations, training, and support roles.

Laurie Hill McQuillan, CISSP, has been a technology consultant for 25 years, providing IT support services to commercial and federal government organizations. McQuillan is vice president of KeyCrest Enterprises, a national security consulting company. She has a Master's degree in technology management and teaches graduate-level classes on the uses of technology for research and the impact of technology on culture. She is treasurer of the Northern Virginia Chapter of the Information Systems Security Association (ISSA) and a founding member of CASPR, an international project that plans to publish Commonly Accepted Security Practices and Recommendations. She can be contacted at LMcQuillan@KeyCrest.com.

Dorsey Morrow, JD, CISSP, is operations manager and general counsel for the International Information Systems Security Certification Consortium, Inc. (ISC)2. He earned a BS degree in computer science and an MBA with an emphasis in information technology. He has served as general counsel to numerous information technology companies and also served as a judge. He is licensed to practice in Alabama, Massachusetts, the 11th Federal Circuit, and the U.S. Supreme Court.

William Hugh Murray, CISSP, is an executive consultant for TruSecure Corporation and a senior lecturer at the Naval Postgraduate School, has more than fifty years experience in information technology and more than thirty years in security. He serves as secretary of $(ISC)^2$ and is an advisor on the Board of Directors of the New York Metropolitan Chapter of ISSA. During more than twenty-five years with IBM his management responsibilities included development of access control programs, advising IBM customers on security, and the articulation of the IBM security product plan. He is the author of the IBM publication, *Information System Security Controls and Procedures.* Mr. Murray has made significant contributions to the literature and the practice of information security. He is a popular speaker on such topics as network security architecture, encryption, PKI, and secure electronic commerce. He is a founding member of the International Committee

to establish the "Generally Accepted System Security Principles" (GASSP) as called for in the National Research Council's Report, *Computers at Risk*. He is a founder and board member of the Colloquium on Information System Security Education (CISSE). He has been recognized as a founder of the systems audit field and by *Information Security* as a Pioneer in Computer Security. In 1987 he received the Fitzgerald Memorial Award for leadership in data security. In 1989 he received the Joseph J. Wasserman Award for contributions to security, audit and control. In 1995 he received a Lifetime Achievement Award from the Computer Security Institute. In 1999 he was enrolled in the ISSA Hall of Fame in recognition of his outstanding contribution to the information security community.

Judith M. Myerson is a systems architect and engineer, and also a freelance writer. She is the editor of *Enterprise Systems Integration, 2nd Edition,* and the author of *The Complete Book of Middleware* and numerous articles, white papers, and reports. In addition to software engineering, her areas of interest include middleware technologies, enterprisewide systems, database technologies, application development, network management, distributed systems, component-based technologies, and project management. You can contact her at jmyerson@bellatlantic.net.

K. Narayanaswamy, Ph.D., Chief Technology Officer and co-founder, Cs3, Inc., is an accomplished technologist who has successfully led the company's research division since inception. He was the principal investigator of several DARPA and NSF research projects that have resulted in the company's initial software product suite, and leads the company's current venture into DDoS and Internet infrastructure technology. He has a Ph.D. in computer science from the University of Southern California.

Matunda Nyanchama, Ph.D., CISSP, is a Senior Advisor, Information Security Analytics at the Bank of Montreal Financial Group. Dr. Nyanchama has held a number of professional security positions, including working as a senior security consultant at Ernst & Young; Director of Security Architecture at Intellitactics Inc., a Canadian security software company; and Telecommunications Engineer at the Kenya Posts & Telecommunications Corporation, Kenya. Dr. Nyanchama has published a number of security management papers and is interested in information protection as a risk management, and information security metrics. Dr. Nyanchama holds masters and doctoral degrees in computer science from the University of Western Ontario in Canada, and an undergraduate electrical engineering degree from the University of Nairobi, Kenya.

Will Ozier, president and founder of OPA Inc. – The Integrated Risk Management Group (OPA), is an expert in risk assessment and contingency planning, with broad experience consulting to Fortune 500 companies and government agencies at all levels. Prior to founding OPA, Ozier held key technical and management positions with leading firms in the manufacturing, financial, and consulting industries. Since then Ozier conceived, developed, and now directs the marketing and evolution of the expert risk analysis and assessment package, BDSS. He chaired the ISSA Information Valuation Committee, which developed and released the ISSA *Guideline for Information Valuation*, and he now chairs the International Information Security Foundation's (IISF) Committee to develop Generally Accepted System Security Principles (GASSP). He consulted to the President's Commission on Critical Infrastructure Protection (PCCIP). He was principal author of The IIA's *Information Security Management: A Call to Action for Corporate Governance*. Ozier is an articulate author and spokesman for information security who has published numerous articles and has presented many talks and seminars in the United States and abroad to a wide variety of audiences.

Keith Pasley, CISSP, is a security professional with over 20 years experience designing and building security architectures for both commercial and federal government. Keith has authored papers and taught security classes and currently working as a regional security practice director.

Ralph Spencer Poore, CISSP, CISA, CFE, is a regular columnist and graybeard in the information security field. As Managing Partner of Pi 'R' Squared Consulting, Ltd., Ralph provides privacy and security consulting services. He is active in national and international standards, is a member of the International Information

Systems Security Certification Consortium, Inc. [(ISC)²] Professional Practices Committee, Chairman of (ISC)² Governance Committee, 2003 recipient of (ISC)² *President's Award*, a member of the Generally Accepted Information Security Principles (GAISP) Steering Committee, a nominee to *Who's Who in Information Security* and an inventor with patents in counter forgery techniques and privacy processes.

Mike Prevost is the DBsign Product Manager at Gradkell Systems, Inc., in Huntsville, Alabama.

Anita Reed, CPA, is currently an accounting doctoral student at the University of South Florida, Tampa, and has 19 years of public accounting experience.

David Rice, CISSP, recognized by the Department of Defense and industry as an information security expert, has spent seven years working on highly sensitive national information security issues and projects. He has held numerous professional certifications; developed and authored several configuration guides, including "Guide to Securing Microsoft Windows 2000 Active Directory," "Guide to Securing Microsoft Windows 2000 Schema," and "Microsoft Windows 2000 Group Policy Reference;" and won Government Executive Magazine's Technical Leadership Award. David is the founder and senior partner of TantricSecurity, LLC, an elite information security consultancy for government and private industry. In addition to his consultancy, research, and publications, David is an adjunct professor for the Information Security Graduate Curriculum at James Madison University, Harrisonburg, Virginia. David Rice is a graduate of the United States Naval Academy and earned his Masters of Science in Systems Engineering and Information Warfare from the Naval Postgraduate School, Monterey, California.

Donald R. Richards, CPP, is former Director of Program Development for IriScan, in Fairfax, Virginia.

Steve A. Rodgers, CISSP, has been assisting clients in securing their information assets for more than six years. Rodgers specializes in attack and penetration testing, security policy and standards development, and security architecture design. He is the co-founder of Security Professional Services (www.securityps.com) and can be reached at srodgers@securityps.com.

Marcus Rogers, Ph.D., CISSP is an assistant research scientist at CERIAS at Purdue University. Prior to that, he was a director with Deloitte & Touche LLP, in Winnipeg, Ontario, Canada

Ben Rothke, CISSP, COO, is a New York City-based senior security consultant with ThruPoint, Inc. and has over 15 years of industry experience in the area of information systems security. His areas of expertise are in PKI, HIPAA, 21 CFR Part 11, design and implementation of systems security, encryption, firewall configuration and review, cryptography and security policy development. Prior to joining ThruPoint, Ben was with Baltimore Technologies, Ernst & Young, and Citicorp, and has provided security solutions to many Fortune 500 companies. Ben is the author of *Computer Security — 20 Things Every Employee Should Know,* a contributing author to *The Handbook of Information Security Management* (Auerbach), and is a former columnist for *Information Security* and *Solutions Integrator* magazine. Ben is also a frequent speaker at industry conferences, such as CSI, RSA, NetSec, and ISACA, and a member of HTCIA, ISSA, ICSA, IEEE, ASIS, CSI and the New Jersey InfraGard chapter.

Ty R. Sagalow is executive vice president and chief operating officer of American International Group eBusiness Risk Solutions, the largest of Internet risk insurance organization. Over the past 18 years, he has held several executive and legal positions within AIG. He graduated summa cum laude from Long Island University, cum laude from Georgetown University Law Center, and holds a Master of Law from New York University. He can be reached at ty.sagalow@aig.com.

Craig Schiller, CISSP, an information security consultant for Hawkeye Security, is the principal author of the first published edition of *Generally Accepted System Security Principles*.

Thomas J. Schleppenbach is a senior information security advisor and security solutions and product manager for Inacom Information Systems in Madison, Wisconsin. With over 16 years of IT experience, Tom provides information security and secure infrastructure design and acts in a strategic role helping organizations plan and build Information Security Programs. Tom also sits on the Western Wisconsin Chapter of InfraGard planning committee and is the co-chair for the Wisconsin Kids Improving Security (KIS) poster contest, working with schools and school districts to educate kids on how to stay safe online. For questions or comments, contact Tom at Tom.Schleppenbach@inacom-msn.com.

E. Eugene Schultz, Ph.D., CISSP, is a principal engineer with Lawrence Berkeley National Laboratory and also teaches computer science courses at the University of California at Berkeley. He previously founded and managed the CIAC (Computer Incident Advisory Capability) for the U.S. Department of Energy and was the Program Manager for the International Information Integrity Institute (I-4). He is co-founder of FIRST (Forum of Incident Response and Security Teams) and an advisor to corporate executives around the world on computer security policy and practice. An expert in a variety of areas within information security, he is the author of four books and over 90 papers. He is a frequent instructor for SANS, ISACA and CSI. Dr. Schultz is also is a member of the ArcSight Security Advisory Board. He has received numerous professional awards, including the NASA Technical Innovation Award, Best Paper Award for the National Information Systems Security Conference, and Information Systems Security Association (ISSA) Professional Contribution Award. Dr. Schultz has also provided expert testimony for the U.S. Senate.

Paul Serritella is a security architect at American International Group. He has worked extensively in the areas of secure application design, encryption, and network security. He received a BA from Princeton University in 1998.

Duane E. Sharp is president of SharpTech Associates, a Canadian company based in Mississauga, Ontario, that specializes in the communication of technology. An electronics engineer with more than 25 years of experience in the technology sector, he has authored numerous articles for clients in information technology and for Auerbach publications, as well as a handbook on interactive computer terminals, and most recently, an Auerbach handbook on CRM entitled *Customer Relationship Management Systems Handbook*.

Ken M. Shaurette, CISSP, CISA, CISM, IAM, is an Information Security Solutions Manager for Omni Tech Corporation in Pewaukee, Wisconsin. With over 25 total years of IT experience, Ken has provided information security and audit advice and vision for companies building information security programs for over 18 of those years. Ken is the President of the Western Wisconsin Chapter of InfraGard, President of ISSA–Milwaukee Chapter (International Systems Security Association), a member of the Wisconsin Association of Computer Crime Investigators (WACCI), a participant in the Cyber Security Alliance (www.staysafeonline.info), co-chair or the HIPAA–COW (Collaborative of Wisconsin) Security Workgroup, and co-chair of the annual Wisconsin InfraGard KIS (Kids Improving Security) Poster Contest.

Sanford Sherizen, Ph.D., CISSP, is President of Data Security Systems, Inc. in Natick, Massachusetts. He can be reach at sherizen@ziplink.net.

Brian Shorten, CISSP, CISA, has been involved in information security since 1986, working in financial institutions and telecommunications companies. He has held positions as data protection officer and business continuity manager. A member of the ISACA, the British Computer Society, and the Business Continuity Institute, he writes and presents on various aspect of information security and business continuity.

Carol A. Siegel is the chief security officer of American International Group. Siegel is a well-known expert in the field of information security and has been in the field for more than ten years. She holds a BS in systems engineering from Boston University, an MBA in computer applications from New York University, and is a CISA. She can be reached at carol.siegel@aig.com.

Valene Skerpac, CISSP, is past chairman of the IEEE Communications Society. Over the past 20 years, she has held positions at IBM and entrepreneurial security companies. Valene is currently president of iBiometrics, Inc.

Ed Skoudis, CISSP, is a consultant at International Network Systems (INS). His expertise includes hacker attacks and defenses, the information security industry, and computer privacy issues. He has performed numerous security assessments, designed secure network architectures, and responded to computer attacks for clients in the financial, high-technology, healthcare, and other industries. A frequent speaker on issues associated with hacker tools and defenses, he has published several articles on these topics, as well as the books, *Malware* (2003) and *Counter Hack* (2001). He is the author of the popular *Crack the Hacker Challenge* series, which challenges InfoSec Professionals to learn from others' mistakes. Additionally, he conducted a demonstration of hacker techniques against financial institutions for the United States Senate. His prior work experience includes Bell Communications Research (Bellcore) and SAIC. Ed received his Master's Degree in Information Networking at Carnegie Mellon University. Ed Skoudis is the vice president of security strategy for Predictive Systems' Global Integrity consulting practice. His expertise includes hacker attacks and defenses, the information security industry, and computer privacy issues. Skoudis is a frequent speaker on issues associated with hacker tools and defenses. He has published the book Counter Hack (Prentice Hall) and the interactive CD-ROM, Hack–Counter Hack.

Robert M. Slade, CISSP, is a data communications and security specialist from North Vancouver, British Columbia, Canada. He has both formal training in data communications and exploration with the BBS and network community, and has done communications training for a number of the international commercial seminar firms. He is the author of "Robert Slade's Guide to Computer Viruses. He has a B.Sc. from the University of BC, and a MS from the University of Oregon. He is the founder of the DECUS Canada Education and Training SIG.

William Stackpole, CISSP, is a senior consultant, Trustworthy Computing Services, for Microsoft Corporation. He was a senior security consultant with Olympic Resource Management in Poulsbo, Washington.

Steve Stanek is a Chicago-based writer specializing in technology issues.

Christopher Steinke, CISSP, Information Security Consulting Staff Member, Lucent World Wide Services, Dallas, Texas

Alan B. Sterneckert, CISA, CISSP, CFE, CCCI, is the owner and general manager of Risk Management Associates located in Salt Lake City, Utah. A retired Special Agent, Federal Bureau of Investigation, Mr. Sterneckert is a professional specializing in risk management, IT system security, and systems auditing. In 2003, Mr. Sterneckert will complete a book about critical incident management, published by Auerbach.

Per Thorsheim is a Senior Consultant with PricewaterhouseCoopers in Bergen, Norway

James S. Tiller, CISSP, Chief Security Officer for International Network Services, manages the development, delivery, and sales of security services worldwide. Jim has spent much of his 15 year career providing secure solutions for organizations throughout North America and Europe. He is author of A Technical Guide to IPSec Virtual Private Networks (Auerbach Publications, 2000) and The Ethical Hack: A Business Value Framework for Penetration Testing (Auerbach Publications, 2004), and holds four patents detailing successful security models and architecture.

Harold F. Tipton, CISSP, currently an independent consultant and Past-President of the International Information System Security Certification Consortium, was Director of Computer Security for Rockwell International Corporation for 15 years. He initiated the Rockwell computer and data security program in 1977 and then continued to administer, develop, enhance and expand the program to accommodate the control needs produced by technological advances until his retirement from Rockwell in 1994.

He has been a member of the Information Systems Security Association (ISSA) since 1982, was president of the Los Angeles Chapter in 1984, and president of the national organization of ISSA (1987–1989). He was added to the ISSA Hall of Fame and the ISSA Honor Role in 2000. He received the Computer Security Institute "Lifetime Achievement Award" in 1994 and the (ISC)² "Hal Tipton Award" in 2001. He was a member of the National Institute for Standards and Technology (NIST) Computer and Telecommunications Security Council and the National Research Council Secure Systems Study Committee (for the National Academy of Science).

He has a BS in engineering from the U.S. Naval Academy, an MA in Personnel Administration from George Washington University, and a Certificate in Computer Science from the University of California at Irvine. He has published several papers on information security issues in the *Information Security Management Handbook*, *Data Security Management, Information Systems Security,* and the *National Academy of Sciences* report, *Computers at Risk*.

He has been a speaker at all of the major information security conferences including: Computer Security Institute, the ISSA Annual Working Conference, the Computer Security Workshop, MIS Conferences, AIS Security for Space Operations, DOE Computer Security Conference, National Computer Security Conference, IIA Security Conference, EDPAA, UCCEL Security & Audit Users Conference, and Industrial Security Awareness Conference. He has conducted and participated in information security seminars for (ISC)², Frost & Sullivan, UCI, CSULB, System Exchange Seminars and the Institute for International Research. He is currently serving as editor of *Data Security Management* and the *Information Security Management Handbook*.

William Tompkins, CISSP, CBCP, is a System Analyst with the Texas Parks and Wildlife Department in Austin, Texas.

James Trulove has more than 25 years of experience in data networking with companies such as Lucent, Ascend, AT&T, Motorola, and Intel. He has a background in designing, configuring, and implementing multimedia communications systems for local and wide area networks, using a variety of technologies. He writes on networking topics and is the author of *LAN Wiring, An Illustrated Guide to Network Cabling* and *A Guide to Fractional T1*, the editor of *Broadband Networking*, as well the author of numerous articles on networking.

Michael Vangelos, CISSP, has over 23 years of IT experience, including 12 specializing in information security. He has managed the information security function at the Federal Reserve Bank of Cleveland for nine years and is currently the bank's information security officer. He is responsible for security policy development, security administration, security awareness, vulnerability assessment, intrusion detection, and information security risk assessment, as well as incident response. He holds a degree in computer engineering from Case Western Reserve University.

Adriaan Veldhuisen is a senior data warehouse/privacy architect with Teradata, San Diego, California.

George Wade is a senior manager with Lucent Technologies in Murray Hill, New Jersey.

Thomas Welch, CISSP, CPP, has over seventeen years in the information systems business, ten of which he designed and developed public safety-related applications. He served as a private investigator and information security consultant since 1988. He was actively engaged in consulting projects, which included security assessments, secure architecture design, security training, high-tech crime investigations and computer forensics. Mr. Welch is an author and frequent lecturer on computer security topics, including computer crime investigation/computer forensics.

Jaymes Williams, CISSP, is a security analyst for the PG&E National Energy Group and is currently the chapter secretary of the Portland, Oregon Chapter of ISSA. He has held security positions at other companies and served eight years in information security-related positions in the U.S. Air Force. The author's proceeds from this chapter will be donated to the Twin Towers fund to benefit those affected by the disaster of September 11, 2001.

Anna Wilson, CISSP, CISA, is a principal consultant with Arqana Technologies, Inc., in Toronto, Ontario.

James M. Wolfe, MSM, is the senior virus researcher and primary technical contact for the Lockheed Enterprise Virus Management Group at Lockheed Martin Corporation. He is a member of the European Institute of Computer Antivirus Researchers (EICAR), the EICAR Antivirus Enhancement Program, the Antivirus Information Exchange Network, Infragard, and is a reporter for the WildList Organization. He has a BS in management information systems and an MS in change management from the University of Central Florida.

John O. Wylder, CISSP, has an extensive background in information technology and the financial services industry. Most recently, he has worked in the field of information security as a consultant. John writes on various topics for a wide variety of publications. John is very active in the business community working, with organizations such as Infragard, and is part of the advisory board of the Georgia Tech School of Economics. John is a graduate of Georgia Tech and has an MBA in finance from Mercer University. He is the author of *Strategic Information Security* (Auerbach Publications, 2003).

Brett Regan Young, CISSP, CBCP, MCSE, and CNE, is Director, Security and Business Continuity Services for Detek Computer Services, Inc., in Houston, Texas. Brett's background includes several years as an independent consultant in the information security and business continuity arenas, primarily for Houston-area companies. Prior to his work as a consultant, he managed the international network of a major oil and gas firm. Brett has also held various positions in the natural gas production, control, and processing environment. Brett has project management experience in the petroleum, banking and insurance industries. He is a frequent contributor to several trade magazines as well as Texas newspapers on the subjects of risk management, security architecture, and business continuity planning and recovery.

Introduction

The research on risks, threats and exposures continues to demonstrate the need for taking an assertive approach to information risk management. According to published sources:

- From 1989 to early 2003, the number of security incidents increased from 130 to over 42,000
- From 2000 to early 2003, the number of security vulnerabilities reported total over 900, which is over twice that of the sum of vulnerabilities reported for the five previous years
- Since 1995, the annual increase in risk from internet hacking is up 60% per year (U.S.)
- Since 1995, the annual increase in risk from viruses and worms is up over 100% per year (U.S.)

Of course, precursors for taking an assertive approach to information risk management are possession of the requisite knowledge and skills as well as the ability to practically apply that knowledge. The mission of the *Information Security Management Handbook* (*ISMH*) is to arm the reader, so that you are prepared to do battle in this challenging environment. The ISMH is designed to cover in detail the ten domains of the Information Security Common Body of Knowledge and offer pragmatic counsel on implementation of technologies, processes and procedures. It is designed to empower the security professional, the information technology professional and the chief information officer with information such that they can do their duty, protect the information assets of their organizations.

This Volume 5 is a blend of some of the most current articles from the previous edition along with new articles that may not have been covered previously. It also includes articles on tried and true topics such as policies, firewalls and Internet security, but with a differing focus or distinction based on the various authors' experiences.

As always, this edition is a comprehensive tome that offers vast amounts of information protection and security advice, from policy development to cryptographic fundamentals and everything between. Whether the reader is an experienced and certified professional (CISSP), an IT executive, or a novice firewall administrator, there is something worthwhile for all.

Hal Tipton

Micki Krause

December, 2003

Domain 1
Access Control Systems and Methodology

The Access Control Systems and Methodology domain addresses the collection of mechanisms that permits system managers to exercise a directing or restraining influence over the behavior, use, and content of a system. Access control permits management to specify what users can do, what resources they can access, and what operations they can perform on a system.

Given the realization that information is valuable and must be secured against misuse, disclosure, and destruction, organizations implement access controls to ensure the integrity and security of the information they use to make critical business decisions. Controlling access to computing resources and information can take on many forms. However, regardless of the method utilized, whether technical or administrative, access controls are fundamental to a well-developed and well-managed information security program.

This domain addresses user identification and authentication, access control techniques and the administration of those techniques, and the evolving and innovative methods of attack against implemented controls.

Biometrics are used to identify and authenticate individuals and are rapidly becoming a popular approach for imposing control over access to information, because they provide the ability to positively identify someone by their personal attributes, typically a person's voice, handprint, fingerprint, or retinal pattern. Although biometric devices have been around for years, innovations continue to emerge. Understanding the potential as well as the limitations of these important tools is necessary so that the technology can be applied appropriately and most effectively. We will lay the foundations here and follow up with more detail in Domain 10: Physical Security.

Nowhere is the use of access controls more apparently important than in protecting the privacy, confidentiality, and security of patient healthcare information. Outside North America, especially in European countries, privacy has been a visible priority for many years. More recently, American consumers have come to demand an assurance that their personal privacy is protected, a demand that demonstrates awareness that their medical information is becoming increasingly widespread and potentially subject to exposure. The Health Insurance Portability and Accountability Act (HIPAA) of 1996 for medical information and the Gramm–Leach–Bliley Act of 1999 for financial information, just to name two regulations, are definitive evidence that the U.S. Government has heeded the mandate of American citizens.

Malicious hacking has been a successful means of undermining information controls and an increasing challenge to the security of information. Hackers tend to chip away at an organization's defenses and have been successful on far too many occasions. In this domain, readers learn about the advancing, state-of-the-art attack tools that have led to highly publicized scenarios; for example, the recent defacement of the U.S. Department of Justice Web site and denial-of-service attacks on many commercial sites.

Social engineering techniques are another of many ways to undercut the installed controls while taking advantage of human nature. In social engineering, unscrupulous persons use devious means to obtain information that can be applied to defeat implemented controls. For example, envision a call to an unsuspecting user by someone masquerading as a desktop technician, in which the caller says he needs the user's network password to diagnose a technical problem and then uses that password to compromise the system.

Contents

1

Enhancing Security through Biometric Technology

Stephen D. Fried, CISSP

Introduction

> The U.S. Immigration and Naturalization Service has begun a program that will allow frequent travelers to the United States to bypass the personal interview and inspection process at selected major airports, by taking electronic readings of the visitor's hand to positively identify the traveler. A similar system is in use at the U.S./Canada border that uses fingerprints and voice recognition to identify people crossing the border.
>
> In 1991, Los Angeles County installed a system that uses fingerprint identification to reduce fraudulent and duplicate claims in the county's welfare system. The county saved more than $5 million in the first six months of use.
>
> Casinos from Las Vegas to Atlantic City use face recognition systems to spot gambling cheats, card counters, and criminals in an attempt to reduce losses and protect their licenses.

All these systems have one thing in common: they all use *biometrics* to provide for enhanced security of people, locations, or financial interests. Biometrics is becoming one of the fastest growing segments of the security field and has gained a great deal of popularity both in the popular press and within the security profession. The use of biometrics — how it works, how it is used, and how effective it can be — is the subject of this chapter.

Biometrics Basics

From its Greek origins, the term "biometrics" literally means "the measurement of life." In more practical usage, biometrics is the science of measuring and analyzing biological information. The use of biometrics involves taking the measurements of various aspects of living (typically human) beings, making analytical judgments on those measurements, and taking appropriate action based on those judgments. Most typically, those judgments help to accurately identify the subject of the measurement. For example, law enforcement officials use the biometric of fingerprints to identify criminals. If the fingerprints of a suspect correspond to the collected at a crime scene, the suspect may be held for further questioning. If the fingerprints do not, the suspect may be set free. In another example, security cameras can scan the faces in the crowd at a football stadium, then match the scanned images against a database of individuals known to be associated with terrorism. If one of the faces in the crowd matches a face in the database, police can take action to take that person into custody. Such a system was used at the 2001 Super Bowl in Tampa Bay, Florida. The system identified 19 individuals in the crowd with criminal records.

Security professionals already have a wide variety of identification and authentication options available to them, including ID badges, passwords, PINs, and smart cards. So why is biometrics different, and why is it considered by many to be the "best" method for accurate identification and authentication? The answer comes from the nature of identification and authentication. Both these processes are based on the concept of *uniqueness*. They assume that there is some unique aspect to an individual that can be isolated and used to positively identify that individual. However, current forms of identification and authentication all suffer from the same fallacy: the "unique" property they measure is artificially attached to the individual. User IDs and passwords are assigned to users and must be remembered by the user. ID badges or tokens are given to users who must then carry them in their possession. Certificate forms of authentication, such as driver's licenses, passports, or X.509 public key certificates are assigned to a person by some authority that attests to the matching between the name on the certificate and the picture or public key the certificate contains. None of these infallibly identify or authenticate the named individual. They can all be fooled or "spoofed" in some form or another.

Biometrics approaches the uniqueness problem in a different way. Instead of artificially attaching some type of uniqueness to the subject, the uniqueness is determined through an intrinsic quality that the subject already possesses. Characteristics such as fingerprints, retina patterns, hand geometry, and DNA are something almost all people already possess and are all naturally unique. It is also something that is with the person at all times and thus available whenever needed. A user cannot forget his finger or leave his voice at home. Biometric traits also have an intrinsic strength in their uniqueness. A person cannot choose a weak biometric in the same way he can choose a weak password or PIN. For very high-security applications, or situations where an extremely high assurance level for identification or authentication is required, this built-in uniqueness gives biometrics the edge it needs over its traditional identification and authentication counterparts.

How Does Biometrics Work?

Although the physiology behind biometrics is quite complex, the process of using biometric measurements in an application is relatively simple. The first step is to determine the specific biometric *characteristic* that must be measured. This is more a function of practicality, personal preference, and user attitude than a strict technology question. The different factors that go into selecting an appropriate biometric measurement are discussed later in this chapter.

Once the specific characteristic to be measured has been determined, a reading of that biometric is taken through some mechanical or technical means. The specific means will be based on the biometric characteristic selected, but biometric readings are generally taken by either (1) photographing or scanning an image of the characteristic, or (2) measuring the characteristic's life signs within the subject. Once the reading is taken, it needs to be modified into a form that makes further comparison easier. Storing the entire scanned or read image for thousands of people would take up large amounts of storage space, and using the whole image for comparison is inefficient. In reality, only a small portion of the entire image contains significant information that is needed for accurate comparison. These significant bits are called *match points*. By identifying and gathering only the match points, biometric measurements can be made accurately and data storage requirements can be significantly reduced.

The match points are collected into a standard format called a *template*. The template is used for further comparison with other templates stored in the system or collected from users. Templates are stored for later retrieval and comparison in whatever data storage system the biometric application is using. Later, when a user needs to be identified or authenticated, another biometric reading is taken of the subject. The template is extracted from this new scan and compared with one or more templates stored in the database. The existence or absence of a matching template will trigger an appropriate response by the system.

Biometric Traits

All biometric systems are based on one of three different types of human traits. *Genotypic* traits are those that are defined by the genetic makeup of the individual. Examples of genotypic traits are facial geometry, hand geometry, and DNA patterns. It is interesting to note that genotypic traits found between identical twins or clones are very similar and often difficult to use as a distinguishing characteristic to tell the two apart.

Randotypic traits are those traits that are formed early in the development of the embryo. Many of the body features that humans possess take on certain patterns during this stage of development, and those patterns are distributed randomly throughout the entire population. This makes duplication highly improbable and, in some cases, impossible. Examples of randotypic traits are fingerprints, iris patterns, and hand-vein patterns.

Behavioral traits are those aspects of a person that are developed through training or repeated learning. As humans develop, they learn certain modes of behavior that they carry throughout their lives. Interestingly, behavioral traits are the one type of biometric trait that can be altered by a person through re-training or behavior modification. Examples of behavioral traits include signature dynamics and keyboard typing patterns.

Common Uses for Biometrics

The science and application of biometrics has found a variety of uses for both security and non-security purposes. *Authentication* of individuals is one of the most popular uses. For example, hand scanners can be used to authenticate people who try to access a high-security building. The biometric reading taken of the subject is then compared against the single record belonging to that individual in the database. When used in this form, biometric authentication is often referred to as *positive matching* or *one-to-one matching*.

Very often, all that is needed is basic *identification* of a particular subject out of a large number of possible subjects. Police in the London borough of Newham use a system of 140 cameras mounted throughout the borough to scan the faces of people passing through the district. Those faces are compared against a database of known criminals to see if any of them are wandering around Newham's streets. In this particular use, the biometric system is performing *negative matching* or *one-to-many matching*. Unlike the single-record lookup used in positive matching, each sample face scanned by the Newham cameras is compared against all the records in the police database looking for a possible match. In effect, the system is trying to show that a particular face is *not* in the database (and, presumably, not an identified criminal).

Fraud prevention is another common use for biometrics. When a user goes through biometric authentication to access a system, that user's identity is then associated with every event, activity, and transaction that the user performs. If a fraudulent transaction is discovered or the system becomes the subject of an investigation or audit, an audit trail of that user's actions can be produced, confirming or refuting their involvement in the illicit activity. If the personnel using the system are made aware of the ID tagging and audit trails, the use of biometrics can actually serve as a deterrent to prevent fraud and abuse.

Biometrics can also be used as a basic *access control* mechanism to restrict access to a high-security area by forcing the identification of individuals before they are allowed to pass. Biometrics are generally used for identification only in a physical security access control role. In other access control applications, biometrics is used as an authentication mechanism. For example, users might be required to biometrically authenticate themselves before they are allowed to view or modify classified or proprietary information. Normally, even in physical access control, it is not efficient to search the database for a match when the person can identify himself (by stating his name or presenting some physical credential) and have the system quickly perform positive matching.

A less security-oriented use of biometrics is to improve an organization's *customer service*. A supermarket can use facial recognition to identify customers at the checkout line. Once customers are identified, they can be given the appropriate "frequent-shopper" discounts, have their credit cards automatically charged, and have their shopping patterns analyzed to offer them more personally targeted sales and specials in the future — all without the customer needing to show a Shopper's Club card or swipe a credit card. Setting aside the privacy aspect of this type of use (for now), this personalized customer service application can be very desirable for consumer-oriented companies in highly competitive markets.

Biometric Measurement Factors

As with any process involving measurement, mechanical reproduction, and analysis, here there are many factors that contribute to the success or failure of the process. All of these factors fall into two general categories: *properties of the characteristics measured* and *properties of the measurement process*.

Characteristic Properties

The most important requirement for determining if a particular characteristic is suitable for biometric measurement is *uniqueness*. The specific characteristic must be measurably unique for each individual in the subject population. As a corollary, the characteristic must be able to produce comparison points that are unique to the particular individual being measured. This uniqueness property is essential, as two people possessing identical characteristics may be able to fool the measurement system into believing one is the other.

The characteristic must also be *universal*, existing in all individuals in the population being measured. This may sound easy at first, because everyone has fingerprints, everyone has DNA, and everyone has a voice. Or do they? When establishing a biometric measurement system, security practitioners need to account for the fact that there will be some part of the measured population that does not have a particular characteristic. For example, people lose fingers to accidents and illness and some people cannot speak. For these people, fingerprint analysis or voice recognition will not work as a valid biometric mechanism. If the number of people in a particular population lacking these qualities is very small, alternate procedures can be set up to handle these cases. If the number is relatively large, an alternative biometric method, or even an altogether different security mechanism, should be considered.

When considering a particular biometric with respect to universality, the security practitioner must also take cultural considerations into account. A measurement system tuned to a specific target population may not perform well with other racial, ethnic, or gender groups. For example, suppose a company uses a voice recognition system that requires users to speak several standard words in order to get an accurate voiceprint. If the system is tuned to clearly understand words spoken by New Yorkers (where the system is used), an employee with a deep southern U.S. accent transferring into the area might have difficulty being recognized when speaking the standard words. Likewise, some cultures have customs regarding the touching of objects and health concerns regarding the shared use of the same device (like a hand scanner or a fingerprint reader). When setting up a biometric system that requires the user to touch or physically interact with the reading device, these types of considerations need to be addressed.

Another important property for a biometric characteristic is *permanence*. The characteristic must be a permanent part of the individual and the individual must not be able to remove or alter the characteristic without causing grave personal harm or danger. This permanence property also applies over time. The characteristic must not change significantly over time or it will make any pattern matching inaccurate. This aspect has several interesting ramifications. For example, the physiology of young children changes quite rapidly during their growing years, so voice or facial characteristics measured when they are young may be invalid just a few years later. Likewise, elderly people who have their physical characteristics damaged through surgery or accidental injury may take an unusually long time to heal, again rendering any physical measurements inaccurate, at least for a time. Pregnancy causes a woman's blood vessels in the back of the eye to change, thereby requiring re-enrollment if retinal scanning is being used. Finally, handwritten signature patterns change over time as people age, or in relation to the number of documents they need to sign on a regular basis. These situations will lead to a higher number of false rejections on the part of the biometric system. To avoid these types of problems it may be advantageous to periodically reestablish a baseline measurement for each individual in the system.

In addition to permanence, the characteristic must be *unalterable*. It should be impossible for a person to change the characteristic without causing an error condition in the biometric system or presenting harm or risk to the subject. For example, it is impossible to change a person's DNA. And while it is theoretically possible to give someone new fingerprints (through skin grafts or digit transplant), most people would consider that too extreme and dangerous to be considered a strong threat for most applications.

It is important that the characteristic has the *ability to be captured or otherwise recognized* by some type of recording device. The characteristic must be measurable by a standard (perhaps specialized) input device that can convert that characteristic (and its match points) to a form that is readable and understandable by human or technical means.

The final important property of any biometric characteristic is that it *can be authenticated*. The characteristic for an individual must be able to be matched against similar characteristics found in other subjects and a definitive positive or negative match must be able to be made based on the measurement and match points presented.

Measurement Properties

The previous section dealt with properties of the various biological characteristics used in biometrics. However, a large part of the success or failure of a biometric system lies in the measurement and analysis process. One of the most important aspects of the process is *accuracy*. As with any monitoring or surveillance system, it is critically important that the biometric system takes accurate measurements and creates an accurate representation of the characteristic in question. Likewise, the template that the system produces from the measurement must accurately depict the characteristic in question and allow the system to perform accurate comparisons with other templates.

The system's ability to produce templates and use these templates in a later evaluation must be *consistent over time*. The measurement process must be able to accurately measure and evaluate the characteristic over an indefinite (although not necessarily infinite) period of time. For example, if an employee enrolls in a face-scanning system on the first day of work, that scanning system should be able to accurately verify that employee throughout the entire length of employment (even accounting for aging, growth or removal of facial hair, and the occasional broken nose).

Because biometric systems are based on examinations of human characteristics, it is important that the system *verify the source of the characteristic,* as opposed to simply checking the characteristic's features or match points. For example, if the system is measuring facial geometry, can holding a picture of the subject's face up to the camera fool it into believing the image is from a real person? If a fingerprint system is used, does the system check to see if the finger is attached to a living person? (This is not as far-fetched as one may think!) Checking for traits like body heat, blood flow, movement, and vocal intonation can help the system distinguish between the real article and a mechanical reproduction.

Finally, the measurement system should work to reduce the influence of *environmental factors* that may play into the accuracy of the biometric readings. An example of this would be the accurate placement of face scanners so that sunlight or glare does not affect the cameras. Fingerprint systems should employ mechanisms to ensure the print reader does not become smudged or laden with dirt, thus affecting its ability to take accurate measurements. The accuracy of a voice matching system might be compromised if it is operated in a crowded or noisy public environment. All these factors work against a successful biometric operation, and all should be considered and dealt with early in the planning phases.

Biometric Measurement

Although the science and technology behind biometrics has improved greatly in recent years, it is not foolproof. Absolute, 100-percent error-free accuracy of the measurements taken by biometric devices, and of the comparisons made between biometric characteristics, is neither realistic nor to be expected. Therefore, implementers of a biometric system need to understand the limitations of the technology and take the appropriate steps to mitigate any possible error-causing conditions. Biometric systems, like all security systems, must be "tuned" based on the particular needs of the installation and must account for real-world variations in use and operating environment.

Measurement Characteristics

The process of comparing biometric templates to determine if they are similar (and how far that similarity extends) is called *matching*. The matching process results in a *score* that indicates how well (or how poorly) the presented template compares against a template found in the database. For every biometric system there is a particular *threshold* that must be met for the system to issue a "pass" result. If the score produced for that match falls above the threshold, the template is accepted. If the score falls below the threshold, the template is rejected. The threshold value is typically set by the system's administrators or operators and is tunable, depending on the degree of sensitivity the operator desires.

Ironically, the template produced by a user during normal system use and the template stored in the system for that user should rarely result in a completely identical match. There is always some degree of change (however small) between user "sessions" in biometric systems, and that degree of change should be accounted for in the system's overall threshold tuning. The detection of a completely identical match between a presented

template and a stored template (e.g., if an intruder obtains a digitized copy of the reader output and subsequently bypasses the reader by feeding the copy into the matching process) may be an indication of tampering or the use of mechanically reproduced biometric characteristics.

Error-Producing Factors

The process of initially measuring a person's characteristics, creating a template, and storing that template in a system is called *enrollment*. During the enrollment process, the system "learns" the biometric characteristic of the subject. This learning process may involve taking several readings of the characteristic under different conditions. As the system gets more experience with the subject, it learns the various ways that the characteristic can be presented and refines the template stored for that user. It then uses that information during actual operation to account for variations in the way the characteristic is presented.

The performance of the enrollment process can have a large impact on the overall accuracy of the system. It is vitally important that enrollment take place not only under ideal conditions (e.g., in a quiet room with good lighting), but also perhaps under less than optimal conditions (e.g., with added background noise or subdued lighting). A well-performed enrollment increases the accuracy of the comparisons made by the system during normal use and will greatly reduce the likelihood of inaccurate readings. If errors are introduced into the enrollment process, they can lead to errors in verifying the user during later system operation or, in extreme conditions, allow for an imposter to be accepted by the system.

Not all the errors introduced into a biometric system are due to mechanical failures or technical glitches. The users of the systems themselves cause many of the problems encountered by biometric systems. Humans are able to easily adapt to new and different situations and learn new modes of behavior much more easily than machines. How a biometric system handles that change will play an important part in its overall effectiveness.

For example, when a biometric system is first put into operation, users might be unsure of how to accurately present their characteristic to the system. How should they hold their head in order to get an accurate eye scan? How do they place their fingers on the reader so an accurate fingerprint reading can be taken? This initial inexperience (and possible discomfort) with the system can lead to a large number of inaccurate readings, along with frustration among the user population. The natural reaction on the part of users will be to blame the system for the inaccuracies when, in fact, it is the user who is making the process more difficult.

As time passes and users become more familiar with the system, they will become conditioned to presenting their information in a way that leads to more accurate measurements. This conditioning will occur naturally and subconsciously as they learn how to "present" themselves for measurement. In effect, the users learn how to be read by the system. This has the effect of speeding up the throughput rate of the system and causing fewer false readings.

User behavior and physiology play a part in the process as well. As humans move through their days, weeks, and months, they experience regular cycles in their physiology and psychology. Some people are more alert and attentive early in the day and show visible signs of fatigue as the day progresses. Others do not reach their physical peak until midday or even the evening. Seasonal changes cause associated physiological changes in some people, and studies have shown that many people grow depressed during the winter months due to the shorter days. Fatigue or stress can also alter a person's physiological makeup. These cyclical changes can potentially affect any biometric reading that may take place.

The *importance of a transaction* also affects user behavior and attitude toward having biometric readings taken. People are much more willing to submit to biometric sampling for more important, critical, sensitive, or valuable transactions. Even nontechnical examples show this to be true. The average person will take more time and care signing a $100,000 check than a $10 check.

Error Rates

With any biometric system there are statistical error rates that affect the overall accuracy of the system. The *False Rejection Rate (FRR)* is the rate at which legitimate system users are rejected and categorized as invalid users. False rejection is also known as a *Type I Error* or a *False Negative*. The general formula for calculating the False Rejection Rate is:

False Rejection Rate = NFR/NEIA (for identification systems)

or

False Acceptance Rate = NFR/NEVA (for authentication systems)

where:

NFR = Number of false rejections

NEIA = Number of enrollee identification attempts

NEVA = Number of enrollee verification attempts

The *False Acceptance Rate (FAR)* is the rate at which nonlegitimate users are accepted by the system as legitimate and categorized as valid users. False acceptance is also known as a *Type II Error* or a *False Positive*. The general formula for calculating the False Acceptance Rate is:

False Acceptance Rate = NFR/NEVA (for authentication systems)

or

False Rejection Rate = NFA/NIVA (for authentication systems)

where:

NFA = Number of false acceptances

NEIA = Number of imposter identification attempts

NEVA = Number of imposter verification attempts

The final statistic that should be known about any biometric system is the *Crossover Error Rate (CER)*, also known as the *Equal Error Rate (EER)*. This is the point where the False Rejection Rate and the False Acceptance Rate are equal over the size of the population. That is, the system is tuned such that the rate of false negatives and the rate of false positives produced by the system are approximately equal. Ideally, the goal is to tune the system to get the Crossover Error Rate as low as possible so as to produce both the fewest false negatives and false positives. However, there are no absolute rules on how to do this, and changes made to the sensitivity of the system affect both factors. Tuning the system for stricter identification in an attempt to reduce false positives will lead to more false negatives, as questionable measurements taken by the system will lean toward rejection rather than acceptance. Likewise, if you tune the system to be more accepting of questionable readings (e.g., in an effort to improve customer service), you increase the likelihood of more false positive readings.

Finally, for every biometric system there is a *Failure To Enroll* rate, or *FTE*. The FTE is the probability that a given user will be unable to enroll in the system. This can be due to errors in the system or because the user's biometric characteristic is not unique enough or is difficult to measure. Users who are unable to provide biometric data (e.g., amputees or those unable to speak) are generally not counted in a system's FTE rate.

Implementation Issues

Like any other automated system that employs highly technological methods, the technology used in biometric systems only plays one part in the overall effectiveness of that system. The other equally important piece is how that technology is implemented in the system and how the users interact with the technology. State-of-the-art technology is of little use if it is implemented poorly or if the users of the system are resistant (or even hostile) to its use.

One important factor is the relative *autonomy of the users* of a biometric system. This refers to the ability of the users to resist or refuse to participate in a system that uses biometric identification. Generally, company employees (or those bound by contractual obligation) can be persuaded or coerced into using the system as a condition of their employment or contract. Although they may resist or protest, they have little recourse or alternative. On the other hand, members of the general public have the ability to opt out of participation in a biometric system that they feel is intrusive or infringes too much on their personal privacy. Each of these users has the power make a "risk-versus-gain" decision and decide whether or not to participate in the system.

Some users will resist using a biometric system that they feel is too *physically intrusive on their person*. Some biometric technologies (e.g., retina scans or fingerprint readings) are more physically imposing on users. Other

technologies, such as voice recognition or facial recognition, are more socially acceptable because they impose less of a personal proximity risk and do not require the user to physically touch anything. As previously stated, cultural aspects pertaining to personal touch or capturing of personal images also play an important part in the issue of intrusiveness. In general, the more physically intrusive a particular biometric technology is, the more users will resist its use and it may also produce higher error rates because uncomfortable users will not become as conditioned to properly presenting themselves for measurement.

The *perception of the user* as to *how the system is being used* also plays an important part in the system's effectiveness. Users want to understand the motivation behind its use. Is the system owner looking to catch "bad guys"? If this is the case, users may feel like they are all potential suspects in the owner's eyes and will not look kindly upon this attempt to "catch" one of them. On the other hand, if the system is being used (and advertised) as a way to protect the people using the system and to prevent unauthorized personnel from entering the premises and harming innocent people, that use may be more readily acceptable to the user population and alter their attitudes toward its use.

Particular technologies themselves might be at issue with users. The use of fingerprints has most often been associated with criminal behavior. Even if a system owner implements a fingerprint scanning system for completely benign purposes, the users of that system may feel as if they are being treated like criminals and resist its use. *Ease of use* is always a factor in the proper operation of a biometric system. Is enrollment performed quickly and does it require minimal effort? Are special procedures needed to perform the biometric measurement, or can the measurements be taken while the user is performing some other activity? How long do users have to wait after taking the measurements to learn if they have passed or failed the process? Proper end-user operational and ergonomic planning can go a long way toward ensuring lower error rates and higher user satisfaction.

In these days of heightened awareness concerning privacy and the security of personal information, it is no wonder that many potential system implementers and users alike have *concerns over the privacy aspects* of the use of biometrics. With most other identification methods, the system gathers information *about* the person in question, such as name, identification number, height, weight, age, etc. With biometric applications, however, the system maintains information *of* the person in question, such as fingerprint patterns or voice patterns. This type of information is truly "personal" in the most literal sense, and many users are uncomfortable sharing that level of personal detail. More than any other technology, biometrics has the ability to capture and record some of the most essentially private information a person possesses.

Many are also concerned with the storage of their personal information. Where will it be stored, how will it be used, and (most importantly) who will have access to it? In effect, the biometric system is storing the very essence of the individual, a characteristic that can uniquely identify that person. If unauthorized individuals were to get hold of that information, they could use it to their advantage or to the victim's detriment. The loss or compromise of stored biometric information presents an opportunity for the truest form of identity theft.

For example, suppose "Joe Badguy" was able to get hold of a user's template used for fingerprint identification. He may be able to use that template to masquerade as that user to the system, or perhaps feed that template into another system to gain access elsewhere. He may even alter the template for a legitimate user and substitute his own template data. At that point, Joe Badguy can present his fingerprints to the system and be correctly identified as "Jane Innocent, authorized user."

Biometrics also *reduces the possibility of anonymity* in the personal lives of its users. Despite the universal use of credit cards in the global economy, many people still prefer to use cash for many transactions because it allows them to retain their anonymity. It is much more difficult to track the flow of cash than it is to trace credit card records. Taking the earlier example of the store using face recognition to help customers speed through the checkout line, suppose the system also stores the items a customer purchases in its database along with the biometric data for that customer. An intruder to that system (or even a trusted insider) will be able to discover potentially embarrassing or compromising information that the subject would rather not make public (e.g., the purchase of certain medications that might be indicative of an embarrassing health condition). By using biometrics to associate people with purchases, you reduce the ability for people to act anonymously — one of the basic tenets of a free society.

A large privacy problem with information systems in general is the issue of *secondary use*. This is the situation where information gathered for one purpose is used (or sold to a third party) for an entirely different purpose. Secondary use is not peculiar to biometric systems per se, but because of the very personal nature of the information stored in a biometric database, the potential for identity fraud is even greater. While a user might

EXHIBIT 1.1 Biometric Technologies by Characteristic Type

Trait Type	Biometric
Rantotypic	Fingerprints
	Eye scanning
	Vein patterns
Genotypic	Facial recognition
	DNA matching
	Hand geometry
	Voice and speech recognition
Behavioral	Signature analysis
	Keystroke dynamics

give grudging approval to have his face used as part of a system for authenticating ATM transactions (after all, that is the trade-off for convenient access to money), that user might not consent to sharing that same biometric characteristic information with a local retailer.

Finally, there is the issue of *characteristic replacement*. When a person has his credit card stolen, the bank issues that person a new card and cancels the old one. When a computer user forgets his password, a system administrator will cancel the old password and assign a new one to the user. In these two processes, when credentials become compromised (through loss or theft), some authority will invalidate the old credential and issue a new (and different) one to the user. Unfortunately, it is not that easy with biometric systems. If a person has their fingerprints stolen they can't call the doctor and get new fingers! And despite advances in cosmetic surgery, getting a new face because the old image has been compromised is beyond the reach of most normal (or sane) people. The use of biometric systems presents unique challenges to security, because compromise of the data in the system can be both unrecoverable and potentially catastrophic to the victim.

When designing the security for a biometrics-based system, the security professional should use all the tools available in the practitioner's toolbox. This includes such time-honored strategies as defense-in-depth, strong access control, separation and rotation of duties, and applying the principle of least privilege to restrict who has access to what parts of the system. Remember that biometric systems store the most personal information about their users, and thus require that extra attention be paid to their security.

Biometric Technologies

The different types of biometric technologies available today can be divided among the three types of biometric traits found in humans. Exhibit 1.1 lists the most common biometric technologies and the trait types with which each is associated.

Fingerprints

Fingerprints are the most popular and most widely used biometric characteristic for identification and authentication. Fingerprints are formed in the fetal stage (at approximately five months) and remain constant throughout a person's lifetime. The human finger contains a large number of ridges and furrows on the surface of the fingertips. Deposits of skin oil or amino acids on the fingers leave the prints on a particular surface. Those prints can be extracted from the surface and analyzed.

- *How it works.* In fingerprint scanning systems, the user places a finger on a small optical or silicon surface the size of a postage stamp for two or three seconds. There are two different types of finger-scanning technology. The first is an *optical scan*, which uses a visual image of a finger. The second uses a *generated electrical field* to electronically capture an image of a finger.
- *Match points used.* The patterns of ridges and furrows in each print are extracted for analysis. Ridge and furrow patterns are classified in four groups: *arch* (which are very rare), *tented arch*, *whorl*, and *loop* (which is the most common). When a line stops or splits, it is called a "minutia." It is the precise pattern and location of the ridges, furrows, and minutiae that give a fingerprint its uniqueness. Most European courts require 16 minutiae for a positive match and a few countries require more. In the United States, the testimony of a fingerprint expert is sufficient to legally establish a match, regardless

of the number of matching minutiae, although a match based on fewer than ten matching points will face a strong objection from the defense.

- *Storage requirements.* Fingerprint systems store either the entire image of the finger or a representation of the match points for comparison. The U.S. Federal Bureau of Investigation stores digitized images at a resolution of 500 pixels per inch with 256 gray levels. With this standard, a single 1.5-square-inch fingerprint image uses approximately 10 megabytes of data per fingerprint card. To save space, many fingerprint storage systems store only information about the ridges, furrows, and minutiae rather than the entire image. The storage requirement for these systems is typically 250 to 1000 bytes per image.
- *Accuracy.* Fingerprint scanning systems tend to exhibit more false negatives (i.e., failure to recognize a legitimate user) than false positives. Most fingerprint systems on the market use a variety of methods to try to detect the presentation of false images. For example, someone might attempt to use latent print residue on the sensor just after a legitimate user accesses the system or even try to use a finger that is no longer connected to its original owner. To combat this, many sensors use special measurements to determine whether a finger is live, and not made of man-made materials (like latex or plastic). Measurements for blood flow, blood-oxygen level, humidity, temperature, pulse, or skin conductivity are all methods of combating this threat.

Eye Scanning

The human eye contains some of the most unique and distinguishing characteristics for use in biometric measurement. The two most common forms of eye-based biometrics are *iris recognition* and *retina recognition*.

- *How it works.* The process of scanning a person's iris consists of analyzing the colored tissue that surrounds the pupil. The scans use a standard video camera and will work from a distance of 2 to 18 inches away, even if the subject is wearing glasses. The iris scan typically takes three- to five seconds. In contrast, retinal scanning analyses the blood vessels found at the back of the eye. Retinal scanning involves the use of a low-intensity green light source that bounces off the user's retina and is then read by the scanner to analyze the patterns. It does, however, require the user to remove glasses, place his eye close to the reading device, and focus at length on a small green light. The user must keep his head still and his eye focused on the light for several seconds, during which time the device will verify the user's identity. Retina scans typically take from ten to twelve seconds to complete.
- *Match points used.* There are more than 200 usable match points in the iris, including rings, furrows, and freckles. Retina scans measure between 400 and 700 different points in order to make accurate templates.
- *Storage requirements.* Typical template size for an iris scan is between 256 and 512 bytes. Most retina scans can be stored in a much smaller template, typically 96 bytes.
- *Accuracy.* The uniqueness of eyes among humans makes eye scanning a very strong candidate for biometric use. This uniqueness even exists between the left and right eyes of the same person. There is no known way to replicate a retina, and a retina from a dead person deteriorates extremely rapidly. The likelihood of a false positive using eye scan technology is extremely low, and its relative speed and ease of use make it an effective choice for security and identification applications. The primary drawbacks to eye scanning as a biometric are the social and health concerns among users needing to be scanned. People are generally uncomfortable allowing something to shine directly into their eyes and are concerned about the residual health effects that may result. This problem is more pronounced among users of retina scanning systems, where the exposure to the scanning light is longer.

Vein Patterns

Vein pattern recognition uses the unique pattern of surface and subcutaneous veins on the human body, most notably around the human hand.

- *How it works.* A special camera and infrared sensor take an image of veins in the palm, wrist, or back of the hand. The image is then digitized into a template and used for comparison.
- *Match points used.* The images show the tree patterns in the veins that are unique to each person, and the veins and other subcutaneous features present large, robust, stable, and largely hidden patterns.

- *Storage requirements.* The template produced from a vein scanner is approximately 250 bytes.
- *Accuracy.* The unique pattern of vein distribution is highly stable and stays the same throughout a person's life into old age. In that respect, vein patterns provide a highly stable biometric for identification. With respect to social acceptability, vein recognition does not have many of the criminal implications that fingerprinting has. Finally, vein patterns are not subject to temporary damage that fingerprints often suffer from through normal use, such as weekend gardening or masonry work. Despite this, vein scanning has not seen the widespread deployment that some of the other biometric measurements have seen.

Facial Recognition

Facial recognition technology involves analyzing certain facial characteristics, storing them in a database, and using them to identify users accessing systems. Humans have a natural ability to recognize a single face with uncanny accuracy, but until relatively recently it has proven extremely difficult to develop a system to handle this task automatically. Recent advances in scientific research and computing power have made facial recognition a powerful and accurate choice for biometric security.

- *How it works.* Facial recognition is based on the principle that there are features of the human face that change very little over a person's lifetime, including the upper sections of eye sockets, the area around cheek bones, and the sides of the mouth. In a typical facial recognition system, the user faces a camera at a distance of one to two feet for three to four seconds. There are several different types of facial recognition. *Eigenface*, developed at MIT, utilizes two-dimensional gray-scale images representing the distinct facial characteristics. Most faces can be reconstructed using 100 to 125 eigenfaces that are converted to numerical coefficients. During analysis, the "live" face will be analyzed using the same process and the results matched against the stored coefficients. The *Feature Analysis* method measures dozens of facial features from different parts of the face. Feature analysis is more forgiving of facial movement or varying camera angles than the Eigenface method. Another alternative, *Neural Network Mapping* systems, compares both the live image and the stored image against each other and conducts a "vote" on whether there is a match. The algorithm can modify the weight it gives to various features during the process to account for difficult lighting conditions or movement of facial features. Finally, *Automatic Face Processing* uses the distances between easily acquired features such as the eyes, the end of nose, and the corners of the mouth.
- *Match points used.* The specific match points used depend on the type of scanning methodology employed. Almost all methods take measurements of facial features as a function of the distance between them or in comparison with "standardized" faces.
- *Storage requirements.* Template size varies based on the method used. One-to-one matching applications generally use templates in the 1 to 2-Kb range. One-to-many applications can use templates as small as 100 bytes.
- *Accuracy.* Many companies marketing facial scanning technology claim accuracy rates as high as 98 to 99 percent. However, a recent U.S. Department of Defense study found that most systems have an accuracy rate of only 50 to 60 percent. Despite this, the ease of use and the lack of need for direct user interaction with scanning devices make facial scanning an attractive method for many applications.

DNA Matching

Perhaps no type of biometric has received more press in recent times than DNA matching. Applications as widely diverse as criminal investigation, disaster victim identification, and child safety have all looked to DNA matching for assistance. The basic hereditary substance found in all living cells is called deoxyribonucleic acid, or DNA. This DNA is created during embryonic development of living creatures and is copied to every cell in the body.

- *How it works.* The majority of DNA molecules are identical for all humans. However, about three million pairs of each person's DNA molecules (called *base pairs*) vary from person to person. When performing DNA analysis, scientists first isolate the DNA contained in a given sample. Next, the DNA is cut into

short fragments that contain identical repeat sequences of DNA known as VNTR. The fragments are then sorted by size and compared to determine a DNA match.

- *Match points used.* Once the VNTR fragments are isolated, they are put through statistical analysis. For example, for any VNTR "locus" of a given length, there may be many people in a population who have a matching VNTR of that length. However, when combined with other samples of VNTR loci, the combination of all those samples becomes a statistically unique pattern possessed only by that person. Using more and more loci, it becomes highly unlikely (statistically) that two unrelated people would have a matching DNA profile.
- *Storage requirements.* DNA matching information can be stored in physical form (using special x-ray film) or in electronic form using a specialized database. Many governments around the world are starting to develop large DNA databases with hundreds of thousands of unique DNA profiles. Because each system stores the DNA template information in its own format, exact sizing requirements are difficult to determine. Note, however, that storing DNA templates is different from storing a person's actual DNA, a medical practice that is gaining in popularity.
- *Accuracy.* Using even four VNTR loci, the probability of finding two people with a DNA match is around one in five million. FBI analysis uses 13 loci on average, making the odds of a match less than one in 100 billion. This makes DNA matching one of the most accurate forms of biometric analysis. However, due to its complexity, DNA analysis is strictly a laboratory science. It is not yet a "consumer marketplace" technology.

Hand Geometry

The process of hand geometry analysis uses the geometric shape and configuration of the features of the hand to conduct identification and authentication. With the exception of fingerprints, individual hand features do not have sufficiently unique information to provide positive identification. However, several features, when taken in combination, provide enough match points to make biometric use possible.

- *How it works.* A user places a hand, palm down, on a large metal surface. On that surface are five short metal contacts, called "guidance pegs." The guidance pegs help the user align the hand on the metal surface for improved accuracy. The device "reads" the hand's properties and records the various match points. Depending on the system, the scan can take a two-dimensional or three-dimensional image. Features such as scars, dirt, and fingernails can be disregarded because these "features" change rapidly over a person's lifetime. Typical hand scans take from two to four seconds.
- *Match points used.* Hand scanning systems typically record 90 to 100 individual hand characteristics, including the length, width, thickness, skin transparency, and surface area of the hand, including the fingers. These features, as well as the relationship each has to each other (e.g., distance, relative size, etc.), are recorded and stored.
- *Storage requirements.* Hand geometry templates can be stored in a relatively small amount of storage, as little as nine bytes. This makes it ideal for applications where memory storage is at a premium, such as smart cards.
- *Accuracy.* The accuracy of hand geometry systems is fairly high, making it a historically popular biometric method. It also has a fairly high acceptance value among users, and current implementations are easy to use. However, hand geometry systems are typically used for authentication purposes, as one-to-many identification matching becomes increasingly more difficult as the size of the database becomes larger. In addition, the equipment can be expensive and difficult to integrate into existing environments.

Voice and Speech Recognition

There are several different varieties of voice-based biometrics. These include *speaker verification*, where patterns in a person's speech are analyzed to positively identify the speaker, and *speech recognition*, which identifies words as they are spoken, irrespective of the individual performing the speaking. Because there is no direct correlation between the speaker and the speech in speech recognition systems, they are *not* useful for identification or authentication. Finally, *voiceprint systems* record a human voice and create an analog or digital representation of the acoustic information present in the speaker's voice.

- *How it works.* A user is positioned near a microphone or telephone receiver so that his voice can be captured and analyzed. The user is prompted to recite a phrase according to one of several scenarios:
  - — *Text-dependent systems* require the user to recite a specific set of predefined words or phrases.
  - — *Text-independent systems* request that the user speak any words or phrases of their choice. These systems use voiceprints to measure the user's speech.
  - — *Text-prompted systems* require the user to recite random words that are supplied by the system.
- The user's voice is digitized by the system and a model template is produced and used for later comparisons. Typical recognition time in voice-based systems is four to six seconds.
- *Match points used.* Each word or phrase spoken into the system is divided into small segments consisting of syllables or phonemes (or small phonetic units), each of which contains several dominant frequencies. These dominant frequencies are fairly consistent over the entire length of the segment. In turn, each of these segments has several (three to five) dominant tones that are captured and converted to a digital format. This digital information is then transferred to a master table. The combined table of tones for all the segments creates the user's unique voiceprint.
- *Storage requirements.* Voiceprint templates vary considerably in size, depending on the application and the quality of voice information required by the system. Storage size can range from 300 to 500 bytes, all the way up to 5000 to 10,000 bytes. This is not particularly well-suited for applications where the storage or analysis system has low memory or storage capacity.
- *Accuracy.* Most voice recognition systems have a high degree of accuracy. The better ones not only analyze the user's voiceprint, but also check for liveliness in an attempt to verify if the voice is original or a mechanical reproduction. Because the system requires no special training on the part of the user, acceptance and convenience satisfaction are high among users. However, external factors such as ambient noise and the fidelity of the recording can negatively affect the accuracy of the process.

Signature Analysis

Probably the least controversial of all the biometric processes is the use of signature analysis. This is because the process of producing a signature, as well as the social and legal implications of accepting one, are well-established in almost all modern societies. Unlike eye scans or fingerprinting, there is almost no social stigma attached to the use of signature-based biometric systems. From a security standpoint, the use of signatures constitutes a deliberate act; they are never given out by accident. Other biometric information, such as eye scans, fingerprints, and DNA, can all be obtained without the user's knowledge. In contrast, a person must deliberately provide his or her signature.

- *How it works.* A user "signs" her name on a special tablet. Rather than using ink to record pen strokes, the tablet uses a special sensor to record the movement of a stylus to simulate the creation of a signature. There are two different types of signature analysis. *Signature comparison* examines the physical features found within the signature, including such characteristics as letter size, spacing, angles, strokes, and slant. Unfortunately, signature comparison systems can be easier to fool because they are susceptible to the use of mechanical reproductions or the handiwork of experienced forgers. In contrast, *dynamic signature verification* goes one step further; in addition to checking the physical features within the signature, it also accounts for the process of creating the signature. Dynamic signature verification systems take into account the changes in speed, timing, pressure, and acceleration that occur as a person signs his or her name. Where an experienced forger can faithfully recreate the look of a victim's signature, only the originator of a signature can repeatedly produce similar penstrokes every time. The typical verification time for a signature biometric system is four to six seconds.
- *Match points used.* The specific match points used vary from vendor to vendor. The most common systems store a digitized graphic representation of the signature as well as the variable pen movement and pressure information recorded during the signature process.
- *Storage requirements.* Most signature analysis systems store templates of approximately 1500 bytes. Some vendors claim that through compression and optimization techniques the template can be reduced to approximately 200 bytes.

- *Accuracy.* Overall, signature analysis systems possess only moderate accuracy, particularly when compared with other types of biometric indicators. This is perhaps due to the wide range of variability with which signature systems must deal. Such factors as fatigue, illness, impatience, and weather all affect how a person signs his or her name in any given instance.

Keystroke Dynamics

One of the most desirable aspects for a potential biometric system is to gather user input without requiring the user to alter his work process or (in the best case) even be aware that the biometric is being measured. To that end, the use of *keystroke dynamics analysis* comes closest to being as unobtrusive on the end user as possible. Measuring keystroke dynamics involves monitoring users as they type on a keyboard and measuring the speed, duration, latencies, errors, force, and intervals of the individual keystrokes. Most computer users can repeatedly type certain known patterns (such as their user ID or a standard phrase) with a consistency that can be repeated and measured, thus making it a natural for biometric use.

- *How it works.* A user types a passphrase into the keyboard. The phrase is one that is previously known to the user and is typically standardized for each user. The system scans the keyboard at a rate of 1000 times per second and records a number of different measurements to create a template. Input time varies, depending on the length of the passphrase, and verification time is typically less than five seconds.
- *Match points used.* The system separates the keystrokes into a series of *digraphs* (two adjacent keystrokes) or *trigraphs* (three adjacent keystrokes). The relationship between each key in the digraph/trigraph is captured and analyzed to create the template for that session. Two aspects of key timing are particularly important: the *dwell time* or *duration* (the amount of time a particular key is held down) and the *flight time* or *latency* (the amount of time between key presses).
- *Storage requirements.* The storage requirements for keystroke dynamics systems depend on the size of the passphrase used and the number of measurements taken per digraph.
- *Accuracy.* The overall accuracy of keystroke-based biometric systems can be highly variable, depending on the method of measurement used and the type of input requested from the user. In a system that uses structured text (i.e., passphrases supplied by the system), rather than allowing the user to supply his own passphrase, accuracy rates of 90 percent or more have been achieved. However, several factors can affect the accuracy, including the user's typing proficiency and even the use of a different keyboard.

Combining Technologies

The choice of which biometric system to use is very much based on the particular security need, the cost and feasibility of implementing a particular method, and the ease with which the measure can be installed and used. However, each different biometric technology has its limitations. When looking to create a high-security environment, it may be advantageous to use a time-honored security strategy: *defense-in-depth*. The concept of defense-in-depth is to place many layers or barriers between a potential attacker and a potential target. Each layer complements and enhances the layer before it, requiring an attacker to jump multiple (and difficult) hurdles to get to the target.

Defense-in-depth can also be applied to biometrics. One method of accomplishing this is through the use of *layering*. The concept behind layering is to use biometric technology in conjunction with other traditional forms of identification and authentication. For example, to gain access to a building, a visitor might have to both show a photo ID card and pass a fingerprint scan. Because photo IDs are not foolproof (despite the use of modern anti-counterfeit techniques like holographic seals and watermarks), the confidence in the accuracy of the process is enhanced by the use of fingerprints to verify that the person on the card and the person at the door are the same.

Another way of providing defense-in-depth is through *multimodal* use of biometrics. In a multimodal installation, two (or more) biometric technologies are used in parallel and the user must pass through each to be successfully identified. For example, a user might need to pass both an iris scan and a voice identification test in order to be admitted into a classified area. Multimodal use of biometrics has a couple of advantages. First, it allows the use of biometric technologies that may have higher error rates because the supplemental

biometric in use will pick up any error slack. Put another way, one biometric technology may have a 10-percent error rate and another may have a 12-percent error rate. By themselves, each of these rates may be too high for practical use. But when combined, the two technologies together may have an error rate of only 1.5 percent. This may be much more acceptable for the potential user. In addition, the use of multiple biometrics allows for more variation in any single measurement. For example, voice recognition systems may have difficulty with scratchy voices (due to a cold), and other biometrics may have difficulty due to altered body features (e.g., scars, bruises, etc.). Multimodal use allows for more variation in body characteristics while still retaining a high overall level of assurance in the biometric process.

Biometric Standards

There are more than 200 vendors developing or marketing biometric equipment and systems. As in any other industry where so many different products and specifications exist, this has led to a situation where there are numerous "standards" for biometric products and measurement, and there are just as many methods of storing, retrieving, and processing biometric information. To rectify the situation and make products and systems more compatible with each other, there have been several efforts to standardize biometric interfaces and processes.

The largest effort is the *Biometric Application Program Interface*, or *BioAPI*. The BioAPI Consortium, a group of more than 90 organizations developing biometric systems and applications, developed the BioAPI. The BioAPI provides applications with a standardized way of interfacing with a broad range of biometric technologies. By using the BioAPI, developers can integrate their biometric systems in a technology-independent and platform-independent manner. For example, developers of finger scanning hardware will be able to integrate their systems with any computing platform, as long as both follow the BioAPI specification. The BioAPI specification is currently in version 1.1 and has been released into the public domain. An open source reference implementation is also available for developers to use for modeling and testing their products.

While the BioAPI addresses the standardization of biometric technology interfaces, the *Common Biometric Exchange File Format*, or *CBEFF*, is concerned with defining a common format for the storage and exchange of biometric templates. Very often, biometric applications will use their own proprietary or platform-specific formats for data storage. Unfortunately, this makes the passing of biometric data between applications or platforms difficult. The CBEFF addresses this issue by defining a platform-independent and biometric-independent format for the storage and exchange of biometric templates between systems and applications. The CBEFF is being promoted by the National Institute of Standards and Technology (NIST) and is gaining wide support as a useful standard.

Conclusion

There was a time when the use of biometric technology was restricted to classified military installations and science-fiction movies. The very notion of using biological traits to identify, authenticate, and track a person seemed too far advanced for "normal" people to consider. However, the day is now here where everyday use of biometrics is not only possible, it is happening everywhere: in office buildings and supermarkets, on computer networks and in banks, on street corners, and at football stadiums. The reduction in cost and the large gains in feasibility and reliability have forced system owners and security professionals alike to consider the use of biometrics in addition to, or even as a replacement for, traditional user identification and authentication systems. Even end users have become more and more accepting of biometrics in their everyday lives, and that trend will only continue into the future. The day is not far off when keyboards will have fingerprint readers built in to replace passwords, ATM machines will use iris scans instead of PINs, and hand scanners will replace ID badges in the office. Whatever the future holds, one thing is certain: biometrics is here to stay and getting more popular. Successful (and informed) security professionals must learn how to plan for, implement, and use biometric technology as part of their ever-growing security toolbox.

2

Biometrics: What Is New?

Judith M. Myerson

For years, security to the network world has been based on what one knows — a password, a PIN, or a piece of personal information such as one's mother's maiden name. This is being supplemented with what one is (a biometric) that one can use with what one has (a card key, smart card, or token). Biometrics measure a person with respect to fingertip, eye, and facial characteristics. One is also measured on how one speaks and strokes keys and the way one walks. At a future date, one may be measured on the way one's ear is formed and how one hears things.

Take a look at traditional biometric systems and then newer technologies and systems. They are followed by short discussions on standardization issues and selection criteria.

Fingerprints

In a few years, the messy days of using black ink pads to get hard copies of fingerprint templates will be a thing of the past. Enter the age of fingerprint sensors that allow one to do things beyond one's wildest dreams. Slide a fingertip on a sensor chip — swiftly and cleanly — to gain access to a remote network system. One will have peace of mind that one's fingerprints can be difficult to duplicate because no two fingerprints are identical.

A fingerprint consists of patterns found on a fingertip. A good pattern consists of the breaks and forks — known as minutiae in fingerprint indexes. An average fingerprint has 40 to 60 minutiae. Even when the patterns are within an acceptable range of minutia, the sensors may not be able to capture all the details of a fingertip. For some individuals, the patterns may become very thin as a result of daily typing on a keyboard or playing difficult classical music pieces on the piano. Additionally, if an individual is born with a genetic defect or has a big scar on the fingertip, the patterns will be difficult to read.

There are four ways of matching the patterns of a fingertip against those of an enrolled fingerprint template: electrical, thermal, optical, and hybrid sensors. An electrical sensor measures the varying electrical field strength between the ridges and valleys of a fingerprint. A thermal sensor measures a temperature difference in a finger swipe, the friction of the ridges generating more heat than the non-touching valleys as they slide along the chip surface. Optical sensors measure differences in wavelengths of the fingerprint. Hybrid sensors are a mixture of optical and electrical capture devices.

Eye Scanning

Unlike a fingertip, an eye can provide thousands of minutiae on its structure. Fingertip minutiae provide information on the pattern of an *external* structure, while eye minutiae look at the pattern of the eye's *internal* structure. One can obtain this information from two sources: retina and iris scanning systems. The former concerns the pattern of veins in the retina, while the latter uses the pattern of fibers, tissues, and rings in the iris.

To scan the unique patterns of the retina, a retina scanner uses a low-intensity light source through an optical coupler. Such a scanner requires one to look into a receptacle and focus on a given point. This raises

concerns about individuals who wear corrective lenses or who do not feel comfortable about close contact with the reading device.

Iris scanning, on the other hand, uses a fairly conventional TV camera element and requires no close contact. Iris biometrics work well with corrective glasses and contacts in place while a lighting source is good. Some airlines have installed iris scanners to expedite the process of admitting travelers onto planes.

Keep in mind that eye patterns may change over time because of illness or injury. Eye scanners are useless to blind people. This is also true for visually impaired individuals, particularly those with retinal damage.

Facial Recognition

Facial recognition systems can automatically scan people's faces as they appear on television or a closed-circuit camera monitoring a building or street. One new system sees the infrared heat pattern of the face as its biometric, implying that the system works in the dark. The casino industry has capitalized on networked-face scanning to create a facial database of scam artists for quick detection by security officers.

The system can become confused when an individual has changed markedly his appearance (e.g., by growing a beard or making an unusual facial expression). Another way of confusing the system is to considerably change the orientation of a person's face toward the cameras. A 15-degree difference in position between the query image and the database image will adversely impact performance. Obviously, at a difference of 45 degrees, recognition becomes ineffective.

Hand and Voice

Hand geometry has been used for prisons. It uses the hand's three-dimensional characteristics, including the length, width, thickness, and contour of the fingers; veins; and other features. A hand must not show swollen parts or genetic defects.

Voice prints are used extensively in Europe for telephone call access. They are more convenient than hand prints particularly in winter when the callers need to wear gloves to warm their hands. A noisy environment, as well injury, age, and illness, can adversely impact voice verification.

What Is New?

To date, biometric applications have been used in prison visitor systems to ensure that identities will not be swapped, and in benefit payment systems to eliminate fraudulent claims. Biometric systems have been set up to check multiple licenses the truck drivers can carry and change to when they cross state lines or national borders. New border control systems monitor travelers entering and leaving the country at selected biometric terminals. Biometric-based voting systems are used to verify the identity of eligible voters, thus eliminating the abuse of proxy voting, although such systems are not yet available on a mass scale.

So, what is new? Especially after arriving at the third millenium that began on January 1, 2001. To provide a glimpse of what is happening, here is a partial list.

- Integration of face, voice, and lip movement
- Wearable biometric systems
- Fingerprint chips on ATM cards
- Personal authentication
- Other stuff

Some of these biometric efforts have already reached the market, while others are still in the research stage. Serving as an impetus to biometric integration is Microsoft through its biometric initiatives.

Integration of Face, Voice, and Lip Movement

The first item, of course, is an interesting one — particularly the biometrics of lip reading movement. More interesting is the integration of this modality with the other two — face and voice. The advantage of this system

is that if one modality is not working properly, the other two modalities will compensate for the errors of the first. What this means is if one modularity is disturbed (e.g., a noisy environment drowning out the voice), the other two modalities still lead to an accurate identification.

One such instance is the BioID, a Multimodal Biometric Identification System as developed by Dialog Communication Systems AG (Erlangen, Germany). This system combines face, voice, and lip movement recognition. The system begins by acquiring the records and processing each biometric feature separately. During the training (enrollment) of the system, biometric templates are generated for each feature. The system then compares these templates with the newly recorded ones and combines the results into one used to recognize people.

BioID collects lip movements by means of an optical-flow technique that calculates a vector field representing the local movement of each image part to the next part in the video sequence. For this process, the preprocessing module cuts the mouth area out of the first 17 images of the video sequence. It gathers the lip movements in 16 vector fields, which represent the movement of the lips from frame to frame. One drawback with reading the lips without hearing the voice is that the lips may appear to move the same way for two or three different words.

The company claims that BioID is suitable for any application in which people require access to a technical system, for example, computer networks, Internet commerce and banking systems, and ATMs. Depending on the application, BioID authorizes people either through identification or verification. In identification mode, the system must search the entire database to identify a person. In verification mode, a person gives his name or a number, which the system then goes directly to a small portion of the database to verify by means of biometric traits.

Wearable Biometrics System

Cameras and microphones today are very small and lightweight and have been successfully integrated with wearable systems used to assist in recognizing faces, for example. Far better than facial recognition software is to have an audio-based camera built into one's eyeglasses. This device can help one remember the name of the person one is looking at by whispering it in one's ear. The U.S. Army has tested such devices for use by border guards in Bosnia. Researchers at the University of Rochester's Center for Future Health are looking at these devices for patients with Alzheimer's disease.

It is expected that the next-generation recognition systems will recognize people in real-time and in much less constrained situations. Systems running in real-time are much more dynamic than those systems restricted to three modalities. When the time comes, the system would have the capability of recognizing a person as one biometric entity — not just one or two biometric pieces of this individual.

Fingerprint Chip on ATM Cards

Most leading banks have been experimenting with biometrics for the ATM machine to combat identity fraud that happens when cards are stolen. One example is placing a fingerprint sensor chip on an ATM. Some companies are looking at PKI with biometrics on an ATM card. PKI uses public-key cryptography for user identification and authentication; the private key would be stored on the ATM card and protected with a biometric. While PKI is mathematically more secure, its main drawback is maintaining secrecy of the user's private key. To be secure, the private key must be protected from compromise. A solution is to store the private key on a smart card and protect it with a biometric.

On January 18, 2001, Keyware (a provider of biometric and centralized authentication solutions) entered into a partnership with Context Systems. The latter is a provider of network security solutions and PKI-enabled applications for a biometric interface as an overlay to the ATM operating system. This interface would replace the standard PIN as the authorization or authentication application. A bank debit card would contain a fingerprint plus a unique identifier number (UIN) such as access card number, bank account number, and other meaningful information the banking institutions can use.

Personal Authentication

Applications in portable authentication include personal computing, cryptography, and automotive. The first is gaining widespread use, while the second associates itself with the first where applicable. The third will be

available once the manufacturers come up with better ways of controlling unfavorable environmental impacts on the chip.

Portable computing is one of the first widespread applications of personal authentication. It involves a fingerprint sensor chip on a laptop, providing access to a corporate network. With appropriate software, the chip authenticates the five entries to laptop contents: login, screen saver, boot-up, file encryption, and then to network access.

Veridicom offers laptop and other portable computing users a smart card reader combined with a fingerprint sensor. It aims to replace passwords for access to data, computer systems, and digital certificates. A smaller more efficient model of the company's sensor chip is available for built-in authentication in keyboards, notebook computers, wireless phones, and Internet appliances.

Cryptography for laptop users can come as a private-key lockbox to provide access to a private key via the owner's fingerprint. The owner can use this lockbox to encrypt information over the private networks and Internet. This lockbox should also contain digital certificates or more secure passwords.

Manufacturers are currently working on automotive sensor chips that one would find on the car door handle, in a key fob to unlock the car, or on the dashboard to turn on the ignition. They are trying to overcome reliability issues, such as the ability of a chip to function under extreme weather conditions and a high temperature in the passenger compartment. Another issue being researched is the ability to withstand an electrostatic discharge at higher levels.

Other New Stuff

Other new stuff includes multi-travel fingerprint applications, public ID cards, and surveillance systems. Multi-travel applications would allow travelers to participate in frequent flyer and border control systems. Travelers could use one convenient fingerprint template to pay for their travel expenses, such as airplane tickets and hotel rooms. A pubic ID card for multipurpose use could incorporate biometrics. For example, a closed-circuit surveillance video camera system can be automatically monitored with facial software.

Researchers are working on relaxing some constraints of existing face recognition algorithms to better adjust to changes due to lighting, aging, rotation in depth, and common expressions. They are also studying how to deal with variations in appearance due to such things as facial hair, glasses, and makeup — problems that already have partial solutions.

The Microsoft Factor

On May 5, 2000, Microsoft entered into a partnership with I/O Software to integrate biometric authentication technology into the Windows operating systems. Microsoft acquired I/O Software's Biometric API (BAPI) technology and SecureSuite core authentication technology to provide users with a higher level of network security based on a personal authorization method.

This integration will enable users to log on to their computers and conduct secure E-commerce transactions using a combination of fingerprint, iris pattern, or voice recognition and a cryptographic private key, instead of a password. A biometric template is much more difficult to duplicate because no two individuals have the same set of characteristics. Biometrics are well-suited to replace passwords and smart card PINs because biometric data cannot be forgotten, lost, stolen, or shared with others.

Standardization Issues

The biometrics industry includes more than 150 separate hardware and software vendors, each with their own proprietary interfaces, algorithms, and data structures. Standards are emerging to provide a common software interface, to allow sharing of biometric templates, and to permit good comparison and evaluation of different biometric technologies.

One such instance is the BioAPI standard that defines a common method for interfacing with a given biometric application. BioAPI is an open-systems standard developed by a consortium of more than 60 vendors and government agencies. Written in C, it consists of a set of function calls to perform basic actions common to all biometric technologies, such as enroll user, verify asserted identity (authentication), and discover identity.

Microsoft, the original founder of the BioAPI Consortium, dropped out and developed its own BAPI biometric interface standard. This standard is based on BAPI technologies that Microsoft acquired from I/O Software. Another draft standard is the Common Biometric Exchange File Format, which defines a common means of exchanging and storing templates collected from a variety of biometric devices. The Biometric Consortium has also presented a proposal for the Common Fingerprint Minutiae Exchange format, which attempts to provide a level of interoperability for fingerprint technology vendors.

In addition to interoperability issues, biometrics standards are seen as a way of building a foundation for biometrics assurance and testing methodologies. Biometric assurance refers to confidence that a biometric device can achieve the intended level of security. Current metrics for comparing biometric technologies are limited.

As a partial solution, the U.S. Department of Defense's Biometrics Management Office and other groups are developing standard testing methodologies. Much of this work is occurring within the contextual framework of the Common Criteria. It is a model that the international security community developed to standardize evaluation and comparison of all security products.

Selection Criteria

The selection of a static, integrated, or dynamic biometrics system depends on perceived user profiles, the need to interface with other systems or databases, environmental conditions, and other parameters for each characteristic, including:

- Ease of use
- Error incidence
- Accuracy
- Cost
- User acceptance
- Required security level
- Long-term suitability

The rating for each parameter, except for the error incidence, varies from medium to very high. The error incidence parameter refers to a short description on what causes the error (e.g., head injury, age, and glasses). This is also a possibility that an imposter could be correctly authenticated (false acceptance as opposed to false rejection where an authorized person is denied access).

Conclusion

We are entering an age of biometrics. Many technologies, once labeled as research projects, are now marketable. Their popularity is attributed to the fact that biometrics are more difficult to steal, forget, or lose than passwords. Each biometric type, however, has it own limitations. It will not work for all individuals because some may have a disability that a biometric system is unable to enroll as a template. They also do not work with individuals who markedly change their appearances.

While integration of facial, voice, and lip movement recognition is an interesting one, higher granularity of lip movements is needed. Many individuals are not aware that lip reading without voice can be somewhat confusing. This is true when lip movements appear to be the same for two or three different words. Wearable biometrics — once science fiction — is now a reality. Seen in comic books decades ago, now one hears about them with regard to military and health use.

Also, today personal computing for laptops along with a fingerprint secure lockbox containing a private key, digital certificates, and secure passwords. Tomorrow, one may be able to swipe one's fingertip on a car door handle to gain access to one's car. This, however, will not happen until the automobile manufacturers succeed in making a chip that can adapt to a variety of weather conditions — ranging from mild to severe.

All of these have raised standardization issues. Standards on interoperability have been recommended, and a few have been implemented. Trailing them are standards on testing methodologies that are still in the developmental stage. Once the standardization efforts become more mature, new biometric technologies we have not yet seen will make their grand entrance to the market. More of these technologies will be more dynamic, in real-time, and in less constrained environments.

Despite the progress that biometrics technologies will make, passwords are here to stay for some individuals who have problems with enrolling a biometric template — due to a genetic defect, illness, age, or injury. Of course, this is an assumption today. It may not be so tomorrow — particularly with breakthrough technologies not yet on the blueprints.

It Is All about Control

Chris Hare, CISSP, CISA

The security professional and the auditor come together around one topic: control. The two professionals may not agree with the methods used to establish control, but their concerns are related. The security professional is there to evaluate the situation, identify the risks and exposures, recommend solutions, and implement corrective actions to reduce the risk. The auditor also evaluates risk, but the primary role is to evaluate the controls implemented by the security professional. This role often puts the security professional and the auditor at odds, but this does not need to be the case.

This chapter discusses controls in the context of the Common Body of Knowledge of the Certified Information Systems Security Professional (CISSP), but it also introduces the language and definitions used by the audit profession. This approach will ease some of the concept misconceptions and terminology differences between the security and audit professions. Because both professions are concerned with control, albeit from different perspectives, the security and audit communities should have close interaction and cooperate extensively.

Before discussing controls, it is necessary to define some parameters. Audit does not mean security. Think of it this way: the security professional does not often think in control terms. Rather, the security professional is focused on what measures or controls should be put into operation to protect the organization from a variety of threats. The goal of the auditor is not to secure the organization but to evaluate the controls to ensure risk is managed to the satisfaction of management. Two perspectives of the same thing — control.

WHAT IS CONTROL?

According to *Webster's Dictionary*, control is a method "to exercise restraining or directing influence over." An organization uses controls to regulate or define the limits of behavior for its employees or its operations for processes and systems. For example, an organization may have a process for defining widgets and uses controls within the process to maintain quality or production standards. Many manufacturing facilities use controls

to limit or regulate production of their finished goods. Professions such as medicine use controls to establish limits on acceptable conduct for their members. For example, the actions of a medical student or intern are monitored, reviewed, and evaluated — hence controlled — until the applicable authority licenses the medical student.

Regardless of the application, controls establish the boundaries and limits of operation.

The security professional establishes controls to limit access to a facility or system or privileges granted to a user. Auditors evaluate the effectiveness of the controls. There are five principle objectives for controls:

1. Propriety of information
2. Compliance with established rules
3. Safeguarding of assets
4. Efficient use of resources
5. Accomplishment of established objectives and goals

Propriety of information is concerned with the appropriateness and accuracy of information. The security profession uses *integrity* or *data integrity* in this context, as the primary focus is to ensure the information is accurate and has not been inappropriately modified.

Compliance with established rules defines the limits or boundaries within which people or systems must work. For example, one method of compliance is to evaluate a process against a defined standard to verify correct implementation of that process.

Safeguarding the organization's assets is of concern for management, the security professional, and the auditor alike. The term *asset* is used to describe any object, tangible or intangible, that has value to the organization.

The *efficient use of resources* is of critical concern in the current market. Organizations and management must concern themselves with the appropriate and controlled use of all resources, including but not limited to cash, people, and time.

Most importantly, however, organizations are assembled to *achieve a series of goals and objectives.* Without goals to establish the course and desired outcomes, there is little reason for an organization to exist.

To complete our definition of controls, Sawyer's *Internal Auditing, 4th Edition,* provides an excellent definition:

> Control is the employment of all the means and devices in an enterprise to promote, direct, restrain, govern, and check upon its various activities for the purpose of seeing that enterprise objectives are met. These means of control include, but are not limited to, form of organization,

> policies, systems, procedures, instructions, standards, committees, charts of account, forecasts, budgets, schedules, reports, checklists, records, methods, devices, and internal auditing.
>
> — Lawrence Sawyer
> *Internal Auditing, 4th Edition*
> The Institute of Internal Auditors

Careful examination of this definition demonstrates that security professionals use many of these same methods to establish control within the organization.

COMPONENTS USED TO ESTABLISH CONTROL

A series of components are used to establish controls, specifically:

- The control environment
- Risk assessment
- Control activities
- Information and communication
- Monitoring

The *control environment* is a term more often used in the audit profession, but it refers to all levels of the organization. It includes the integrity, ethical values, and competency of the people and management. The organizational structure, including decision making, philosophy, and authority assignments are critical to the control environment. Decisions such as the type of organizational structure, where decision-making authority is located, and how responsibilities are assigned all contribute to the control environment. Indeed, these areas can also be used as the basis for directive or administrative controls as discussed later in the chapter.

Consider an organization where all decision-making authority is at the top of the organization. Decisions and progress are slower because all information must be focused upward. The resulting pace at which the organization changes is lower, and customers may become frustrated due to the lack of employee empowerment.

However, if management abdicates its responsibility and allows anyone to make any decision they wish, anarchy results, along with differing decisions made by various employees. Additionally, the external audit organization responsible for reviewing the financial statements may have less confidence due to the increased likelihood that poor decisions are being made.

Risk assessments are used in many situations to assess the potential problems that may arise from poor decisions. Project managers use risk assessments to determine the activities potentially impacting the schedule or budget associated with the project. Security professionals use risk

assessments to define the threats and exposures and to establish appropriate controls to reduce the risk of their occurrence and impact. Auditors also use risk assessments to make similar decisions, but more commonly use risk assessment to determine the areas requiring analysis in their review.

Control activities revolve around authorizations and approvals for specific responsibilities and tasks, verification and review of those activities, and promoting job separation and segregation of duties within activities. The control activities are used by the security professional to assist in the design of security controls within a process or system. For example, SAP associates a transaction — an activity — with a specific role. The security professional assists in the review of the role to ensure no unauthorized activity can occur and to establish proper segregation of duties.

The *information and communication* conveyed within an organization provide people with the data they need to fulfill their job responsibilities. Changes to organizational policies or management direction must be effectively communicated to allow people to know about the changes and adjust their behavior accordingly. However, communications with customers, vendors, government, and stockholders are also of importance. The security professional must approach communications with care. Most commonly, the issue is with the security of the communication itself. Was the communication authorized? Can the source be trusted, and has the information been modified inappropriately since its transmission to the intended recipients? Is the communication considered sensitive by the organization, and was the confidentiality of the communication maintained?

Monitoring of the internal controls systems, including security, is of major importance. For example, there is little value gained from the installation of intrusion detection systems if there is no one to monitor the systems and react to possible intrusions. Monitoring also provides a sense of learning or continuous improvement. There is a need to monitor performance, challenge assumptions, and reassess information needs and information systems in order to take corrective action or even take advantage of opportunities for enhanced operations. Without monitoring or action resulting from the monitoring, there is no evolution in an organization. Organizations are not closed static systems and, hence, must adapt their processes to changes, including controls. Monitoring is a key control process to aid the evolution of the organization.

CONTROL CHARACTERISTICS

Several characteristics available to assess the effectiveness of the implemented controls are commonly used in the audit profession. Security professionals should consider these characteristics when selecting or designing the control structure. The characteristics are:

- Timeliness
- Economy
- Accountability
- Placement
- Flexibility
- Cause identification
- Appropriateness
- Completeness

Ideally, controls should prevent and detect potential deviations or undesirable behavior early enough to take appropriate action. The *timeliness* of the identification and response can reduce or even eliminate any serious cost impact to the organization. Consider anti-virus software: organizations deploying this control must also concern themselves with the delivery method and timeliness of updates from the anti-virus vendor. However, having updated virus definitions available is only part of the control because the new definitions must be installed in the systems as quickly as possible.

Security professionals regularly see solutions provided by vendors that are not *economical* due to the cost or lack of scalability in large environments. Consequently, the control should be economical and cost effective for the benefit it brings. There is little economic benefit for a control costing $100,000 per year to manage a risk with an annual impact of $1000.

The control should be designed to hold people *accountable* for their actions. The user who regularly attempts to download restricted material and is blocked by the implemented controls must be held accountable for such attempts. Similarly, financial users who attempt to circumvent the controls in financial processes or systems must also be held accountable. In some situations, users may not be aware of the limits of their responsibilities and thus may require training. Other users knowingly attempt to circumvent the controls. Only an investigation into the situation can tell the difference.

The effectiveness of the control is often determined by its *placement*. Accepted placement of controls are considered:

- *Before an expensive part of a process.* For example, before entering the manufacturing phase of a project, the controls must be in place to prevent building the incorrect components.
- *Before points of difficulty or no return.* Some processes or systems have a point where starting over introduces new problems. Consequently, these systems must include controls to ensure all the information is accurate before proceeding to the next phase.
- *Between discrete operations.* As one operation is completed, a control must be in place to separate and validate the previous operation. For

example, authentication and authorization are linked but discrete operations.

- *Where measurement is most convenient.* The control must provide the desired measurement in the most appropriate place. For example, to measure the amount and type of traffic running through a firewall, the measurement control would not be placed at the core of the network.
- *Corrective action response time.* The control must alert appropriate individuals and initiate corrective action either automatically or through human intervention within a defined time period.
- *After the completion of an error-prone activity.* Activities such as data entry are prone to errors due to keying the data incorrectly.
- *Where accountability changes.* Moving employee data from a human resources system to a finance system may involve different accountabilities. Consequently, controls should be established to provide both accountable parties confidence in the data export and import processes.

As circumstances or situations change, so too must the controls. *Flexibility* of controls is partially a function of the overall security architecture. The firewall with a set of hard-coded and inflexible rules is of little value as organizational needs change. Consequently, controls should ideally be modular in a systems environment and easily replaced when new methods or systems are developed.

The ability to respond and correct a problem when it occurs is made easier when the control can *establish the cause* of the problem. Knowing the cause of the problem makes it easier for the appropriate corrective action to be taken.

Controls must provide management with the *appropriate* responses and actions. If the control impedes the organization's operations or does not address management's concerns, it is not appropriate. As is always evident to the security professional, a delicate balance exists between the two; and often the objectives of business operations are at odds with other management concerns such as security. For example, the security professional recommending system configuration changes may affect the operation of a critical business system. Without careful planning and analysis of the controls, the change may be implemented and a critical business function paralyzed.

Finally, the control must be complete. Implementing controls in only one part of the system or process is no better than ignoring controls altogether. This is often very important in information systems. We can control the access of users and limit their ability to perform specific activities within an application. However, if we allow the administrator or programmer a backdoor into the system, we have defeated the controls already established.

There are many factors affecting the design, selection, and implementation of controls. This theme runs throughout this chapter and is one the security professional and auditor must each handle on a daily basis.

TYPES OF CONTROLS

There are many types of controls found within an organization to achieve its objectives. Some are specific to particular areas within the organization but are nonetheless worthy of mention. The security professional should be aware of the various controls because he will often be called upon to assist in their design or implementation.

Internal

Internal controls are those used to primarily manage and coordinate the methods used to safeguard an organization's assets. This process includes verifying the accuracy and reliability of accounting data, promoting operational efficiency, and adhering to managerial polices.

We can expand upon this statement by saying internal controls provide the ability to:

- Promote an effective and efficient operation of the organization, including quality products and services
- Reduce the possibility of loss or destruction of assets through waste, abuse, mismanagement, or fraud
- Adhere to laws and external regulations
- Develop and maintain accurate financial and managerial data and report the same information to the appropriate parties on a timely basis

The term *internal control* is primarily used within the audit profession and is meant to extend beyond the limits of the organization's accounting and financial departments.

Directive/Administrative

Directive and administrative controls are often used interchangeably to identify the collection of organizational plans, policies, and records. These are commonly used to establish the limits of behavior for employees and processes. Consider the organizational conflict of interest policy.

Such a policy establishes the limits of what the organization's employees can do without violating their responsibilities to the organization. For example, if the organization states employees cannot operate a business on their own time and an employee does so, the organization may implement the appropriate repercussions for violating the administrative control.

Using this example, we can more clearly see why these mechanisms are called *administrative* or *directive* controls — they are not easily enforced in

automated systems. Consequently, the employee or user must be made aware of limits and stay within the boundaries imposed by the control.

One directive control is legislation. Organizations and employees are bound to specific conduct based upon the general legislation of the country where they work, in addition to any specific legislation regarding the organization's industry or reporting requirements. Every organization must adhere to revenue, tax collection, and reporting legislation. Additionally, a publicly traded company must adhere to legislation defining reporting requirements, senior management, and the responsibilities and liabilities of the board of directors. Organizations that operate in the healthcare sector must adhere to legislation specific to the protection of medical information, confidentiality, patient care, and drug handling. Adherence to this legislation is a requirement for the ongoing existence of the organization and avoidance of criminal or civil liabilities.

The organizational structure is an important element in establishing decision-making and functional responsibilities. The division of functional responsibilities provides the framework for segregation of duties controls. Through segregation of duties, no single person or department is responsible for an entire process. This control is often implemented within the systems used by organizations.

Aside from the division of functional responsibilities, organizations with a centralized decision-making authority have all decisions made by a centralized group or person. This places a high degree of control over the organization's decisions, albeit potentially reducing the organization's effectiveness and responsiveness to change and customer requirements.

Decentralized organizations place decision making and authority at various levels in the company with a decreasing range of approval. For example, the president of the company can approve a $1 million expenditure, but a first-level manager cannot. Limiting the range and authority of decision making and approvals gives the company control while allowing the decisions to be made at the correct level. However, there are also many examples in the news of how managers abuse or overstep their authority levels. The intent in this chapter is not to present one as better than the other but rather to illustrate the potential repercussions of choosing either. The organization must make the decision regarding which model is appropriate at which time.

The organization also establishes internal policies to control the behavior of its employees. These policies typically are implemented by procedures, standards, and guidelines. Policies describe senior management's decisions. They limit employee behavior by typically adding sanctions for noncompliance, often affecting an employee's position within the organization. Policies may also include codes of conduct and ethics in addition to

the normal finance, audit, HR, and systems policies normally seen in an organization.

The collective body of documentation described here instructs employees on what the organization considers acceptable behavior, where and how decisions are made, how specific tasks are completed, and what standards are used in measuring organizational or personal performance.

Accounting

Accounting controls are an area of great concern for the accounting and audit departments of an organization. These controls are concerned with safeguarding the organization's financial assets and accounting records. Specifically, these controls are designed to ensure that:

- Only authorized transactions are performed, recorded correctly, and executed according to management's directions.
- Transactions are recorded to allow for preparation of financial statements using generally accepted accounting principles.
- Access to assets, including systems, processes, and information, is obtained and permitted according to management's direction.
- Assets are periodically verified against transactions to verify accuracy and resolve inconsistencies.

While these are obviously accounting functions, they establish many controls implemented within automated systems. For example, an organization that allows any employee to make entries into the general ledger or accounting system will quickly find itself financially insolvent and questioning its operational decisions.

Financial decision making is based upon the data collected and reported from the organization's financial systems. Management wants to know and demonstrate that only authorized transactions have been entered into the system. Failing to demonstrate this or establish the correct controls within the accounting functions impacts the financial resources of the organization. Additionally, internal or external auditors cannot validate the authenticity of the transactions; they will not only indicate this in their reports but may refuse to sign the organization's financial reports. For publicly traded companies, failing to demonstrate appropriate controls can be disastrous.

The recent events regarding mishandling of information and audit documentation in the Enron case (United States, 2001–2002) demonstrate poor compliance with legislation, accepted standards, accounting, and auditing principles.

Preventive

As presented thus far, controls may exist for the entire organization or for subsets of specific groups or departments. However, some controls are implemented to prevent undesirable behavior before it occurs. Other controls are designed to detect the behaviors when they occur, to correct them, and improve the process so that a similar behavior will not recur.

This suite of controls is analogous to the prevent–detect–correct cycle used within the information security community.

Preventive controls establish mechanisms to prevent the undesirable activity from occurring. Preventive controls are considered the most cost-effective approach of the preventive–detective–corrective cycle. When a preventive control is embedded into a system, the control prevents errors and minimizes the use of detective and corrective techniques. Preventive controls include trustworthy, trained people, segregation of duties, proper authorization, adequate documents, proper record keeping, and physical controls.

For example, an application developer who includes an edit check in the zip or postal code field of an online system has implemented a preventive control. The edit check validates the data entered as conforming to the zip or postal code standards for the applicable country. If the data entered does not conform to the expected standards, the check generates an error for the user to correct.

Detective

Detective controls find errors when the preventive system does not catch them. Consequently, detective controls are more expensive to design and implement because they not only evaluate the effectiveness of the preventive control but must also be used to identify potentially erroneous data that cannot be effectively controlled through prevention. Detective controls include reviews and comparisons, audits, bank and other account reconciliation, inventory counts, passwords, biometrics, input edit checks, checksums, and message digests.

A situation in which data is transferred from one system to another is a good example of detective controls. While the target system may have very strong preventive controls when data is entered directly, it must accept data from other systems. When the data is transferred, it must be processed by the receiving system to detect errors. The detection is necessary to ensure that valid, accurate data is received and to identify potential control failures in the source system.

Corrective

The corrective control is the most expensive of the three to implement and establishes what must be done when undesirable events occur. No

matter how much effort or resources are placed into the detective controls, they provide little value to the organization if the problem is not corrected and is allowed to recur.

Once the event occurs and is detected, appropriate management and other resources must respond to review the situation and determine why the event occurred, what could have been done to prevent it, and implement the appropriate controls. The corrective controls terminate the loop and feed back the new requirements to the beginning of the cycle for implementation.

From a systems security perspective, we can demonstrate these three controls.

- An organization is concerned with connecting the organization to the Internet. Consequently, it implements firewalls to limit (prevent) unauthorized connections to its network. The firewall rules are designed according to the requirements established by senior management in consultation with technical and security teams.
- Recognizing the need to ensure the firewall is working as expected and to capture events not prevented by the firewall, the security teams establish an intrusion detection system (IDS) and a log analysis system for the firewall logs. The IDS is configured to detect network behaviors and anomalies the firewall is expected to prevent. Additionally, the log analysis system accepts the firewall logs and performs additional analysis for undesirable behavior. These are the detective controls.
- Finally, the security team advises management that the ability to review and respond to issues found by the detective controls requires a computer incident response team (CIRT). The role of the CIRT is to accept the anomalies from the detective systems, review them, and determine what action is required to correct the problem. The CIRT also recommends changes to the existing controls or the addition of new ones to close the loop and prevent the same behavior from recurring.

Deterrent

The deterrent control is used to discourage violations. As a control itself, it cannot prevent them. Examples of deterrent controls are sanctions built into organizational policies or punishments imposed by legislation.

Recovery

Recovery controls include all practices, procedures, and methods to restore the operations of the business in the event of a disaster, attack, or system failure. These include business continuity planning, disaster recovery plans, and backups.

All of these mechanisms enable the enterprise to recover information, systems, and business processes, thereby restoring normal operations.

Compensating

If the control objectives are not wholly or partially achieved, an increased risk of irregularities in the business operation exists. Additionally, in some situations, a desired control may be missing or cannot be implemented. Consequently, management must evaluate the cost–benefit of implementing additional controls, called compensating controls, to reduce the risk. Compensating controls may include other technologies, procedures, or manual activities to further reduce risk.

For example, it is accepted practice to prevent application developers from accessing a production environment, thereby limiting the risk associated with insertion of improperly tested or unauthorized program code changes. However, in many enterprises, the application developer may be part of the application support team. In this situation, a compensating control could be used to *allow* the developer *restricted* (monitored and/or limited) access to the production system, *only when access is required*.

CONTROL STANDARDS

With this understanding of controls, we must examine the control standards and objectives of security professionals, application developers, and system managers. Control standards provide developers and administrators with the knowledge to make appropriate decisions regarding key elements within the security and control framework. The standards are closely related to the elements discussed thus far.

Standards are used to implement the control objectives, namely:

- Data validation
- Data completeness
- Error handling
- Data management
- Data distribution
- System documentation

Application developers who understand these objectives can build applications capable of meeting or exceeding the security requirements of many organizations. Additionally, the applications will be more likely to satisfy the requirements established by the audit profession.

Data accuracy standards ensure the correctness of the information as entered, processed, and reported. Security professionals consider this an element of data integrity. Associated with data accuracy is data completeness. Similar to ensuring the accuracy of the data, the security professional

must also be concerned with ensuring that all information is recorded. Data completeness includes ensuring that only authorized transactions are recorded and none are omitted.

Timeliness relates to processing and recording the transactions in a timely fashion. This includes service levels for addressing and resolving error conditions. Critical errors may require that processing halts until the error is identified and corrected.

Audit trails and logs are useful in determining what took place after the fact. There is a fundamental difference between audit trails and logs. The audit trail is used to record the status and processing of individual transactions. Recording the state of the transaction throughout the processing cycle allows for the identification of errors and corrective actions. Log files are primarily used to record access to information by individuals and what actions they performed with the information.

Aligned with audit trails and logs is system monitoring. System administrators implement controls to warn of excessive processor utilization, low disk space, and other conditions. Developers should insert controls in their applications to advise of potential or real error conditions. Management is interested in information such as the error condition, when it was recorded, the resolution, and the elapsed time to determine and implement the correction.

Through techniques including edit controls, control totals, log files, checksums, and automated comparisons, developers can address traditional security concerns.

CONTROL IMPLEMENTATION

The practical implementations of many of the control elements discussed in this chapter are visible in today's computing environments. Both operating system and application-level implementations are found, often working together to protect access and integrity of the enterprise information.

The following examples illustrate and explain various control techniques available to the security professional and application developer.

Transmission Controls

The movement of data from the origin to the final processing point is of importance to security professionals, auditors, management, and the actual information user. Implementation of transmission controls can be established through the communications protocol itself, hardware, or within an application.

For example, TCP/IP implementations handle transmission control through the retransmission of information errors when received. The ability of TCP/IP to perform this service is based upon error controls built into the protocol or service. When a TCP packet is received and the checksum calculated for the packet is incorrect, TCP requests retransmission of the packet. However, UDP packets must have their error controls implemented at the application layer, such as with NFS.

Sequence

Sequence controls are used to evaluate the accuracy and completeness of the transmission. These controls rely upon the source system generating a sequence number, which is tested by the receiving system. If the data is received out of sequence or a transmission is missing, the receiving system can request retransmission of the missing data or refuse to accept or process any of it.

Regardless of the receiving system's response, the sequence controls ensure data is received and processed in order.

Hash

Hash controls are stored in the record before it is transmitted. These controls identify errors or omissions in the data. Both the transmitting and receiving systems must use the same algorithm to compute and verify the computed hash. The source system generates a hash value and transmits both the data and the hash value.

The receiving system accepts both values, computes the hash, and verifies it against the value sent by the source system. If the values do not match, the data is rejected. The strength of the hash control can be improved through strong algorithms that are difficult to fake and by using different algorithms for various data types.

Batch Totals

Batch totals are the precursors to hashes and are still used in many financial systems. Batch controls are sums of information in the transmitted data. For example, in a financial system, batch totals are used to record the number of records and the total amounts in the transmitted transactions. If the totals are incorrect on the receiving system, the data is not processed.

Logging

A transaction is often logged on both the sending and receiving systems to ensure continuity. The logs are used to record information about the

transmission or received data, including date, time, type, origin, and other information.

The log records provide a history of the transactions, useful for resolving problems or verifying that transmissions were received. If both ends of the transaction keep log records, their system clocks must be synchronized with an external time source to maintain traceability and consistency in the log records.

Edit

Edit controls provide data accuracy and consistency for the application. With edit activities such as inserting or modifying a record, the application performs a series of checks to validate the consistency of the information provided.

For example, if the field is for a zip code, the data entered by the user can be verified to conform to the data standards for a zip code. Likewise, the same can be done for telephone numbers, etc.

Edit controls must be defined and inserted into the application code as it is developed. This is the most cost-efficient implementation of the control; however, it is possible to add the appropriate code later. The lack of edit controls affects the integrity and quality of the data, with possible repercussions later.

PHYSICAL

The implementation of physical controls in the enterprise reduces the risk of theft and destruction of assets. The application of physical controls can decrease the risk of an attacker bypassing the logical controls built into the systems. Physical controls include alarms, window and door construction, and environmental protection systems. The proper application of fire, water, electrical, temperature, and air controls reduces the risk of asset loss or damage.

DATA ACCESS

Data access controls determine who can access data, when, and under what circumstances. Common forms of data access control implemented in computer systems are file permissions. There are two primary control methods — discretionary access control and mandatory access control.

Discretionary access control, or DAC, is typically implemented through system services such as file permissions. In the DAC implementation, the user chooses who can access a file or program based upon the file permissions established by the owner. The key element here is that the ability to access the data is decided by the owner and is, in turn, enforced by the system.

Mandatory access control, also known as MAC, removes the ability of the data owner alone to decide who can access the data. In the MAC model, both the data and the user are assigned a classification and clearance. If the clearance assigned to the user meets or exceeds the classification of the data and the owner permits the access, the system grants access to the data. With MAC, the owner and the system determine access based upon owner authorization, clearance, and classification.

Both DAC and MAC models are available in many operating system and application implementations.

WHY CONTROLS DO NOT WORK

While everything present in this chapter makes good sense, implementing controls can be problematic. Overcontrolling an environment or implementing confusing and redundant controls results in excessive human/monetary expense. Unclear controls might bring confusion to the work environment and leave people wondering what they are supposed to do, delaying and impacting the ability of the organization to achieve its goals. Similarly, controls might decrease effectiveness or entail an implementation that is costlier than the risk (potential loss) they are designed to mitigate.

In some situations, the control may become obsolete and effectively useless. This is often evident in organizations whose polices have not been updated to reflect changes in legislation, economic conditions, and systems.

Remember: people will resist attempts to control their behaviors. This is human nature and very common in situations in which the affected individuals were not consulted or involved in the development of the control. Resistance is highly evident in organizations in which the controls are so rigid or overemphasized as to cause mental or organizational rigidity. The rigidity causes a loss of flexibility to accommodate certain situations and can lead to strict adherence to procedures when common sense and rationality should be employed.

Personnel can and will accept controls. Most people are more willing to accept them if they understand what the control is intended to do and why. This means the control must be a means to an end and not the end itself. Alternatively, the control may simply not achieve the desired goal. There are four primary reactions to controls the security professional should consider when evaluating and selecting the control infrastructure:

1. *The control is a game.* Employees consider the control as a challenge, and they spend their efforts in finding unique methods to circumvent the control.
2. *Sabotage.* Employees attempt to damage, defeat, or ignore the control system and demonstrate, as a result, that the control is worthless.

3. *Inaccurate information.* Information may be deliberately managed to demonstrate the control as ineffective or to promote a department as more efficient than it really is.
4. *Control illusion.* While the control system is in force and working, employees ignore or misinterpret results. The system is credited when the results are positive and blamed when results are less favorable.

The previous four reactions are fairly complex reactions. Far more simplistic reactions leading to the failure of control systems have been identified:

- *Apathy.* Employees have no interest in the success of the system, leading to mistakes and carelessness.
- *Fatigue.* Highly complex operations result in fatigue of systems and people. Simplification may be required to address the problem.
- *Executive override.* The executives in the organization provide a "get out of jail free" card for ignoring the control system. Unfortunately, the executives involved may give permission to employees to ignore all the established control systems.
- *Complexity.* The system is so complex that people cannot cope with it.
- *Communication.* The control operation has not been well communicated to the affected employees, resulting in confusion and differing interpretations.
- *Efficiency.* People often see the control as impeding their abilities to achieve goals.

Despite the reasons why controls fail, many organizations operate in very controlled environments due to business competitiveness, handling of national interest or secure information, privacy, legislation, and other reasons. People can accept controls and assist in their design, development, and implementation. Involving the correct people at the correct time results in a better control system.

SUMMARY

This chapter has examined the language of controls, including definitions and composition. It has looked at the different types of controls, some examples, and why controls fail. The objective for the auditor and the security professional alike is to understand the risk the control is designed to address and implement or evaluate as their role may be. Good controls do depend on good people to design, implement, and use the control.

However, the balance between the good and the bad control can be as simple as the cost to implement or the negative impact to business operations. For a control to be effective, it must achieve management's objectives, be relevant to the situation, be cost effective to implement, and easy for the affected employees to use.

Acknowledgments

Many thanks to my colleague and good friend, Mignona Cote. She continues to share her vast audit experience daily, having a positive effect on information systems security and audit. Her mentorship and leadership have contributed greatly to my continued success.

References

Gallegos, Frederick. *Information Technology Control and Audit*. Auerbach Publications, Boca Raton, FL, 1999.

Sawyer, Lawrence. *Internal Auditing*. The Institute of Internal Auditors, 1996.

ABOUT THE AUTHOR

Chris Hare, CISSP, CISA, is an information security and control consultant with Nortel Networks in Dallas, Texas. A frequent speaker and author, his experience includes application design, quality assurance, systems administration and engineering, network analysis, and security consulting, operations, and architecture.

3

Controlling FTP: Providing Secured Data Transfers

Chris Hare, CISSP, CISA

Several scenarios exist that must be considered when looking for a solution:

- The user with a log-in account who requires FTP access to upload or download reports generated by an application. The user does not have access to a shell; rather, his default connection to the box will connect him directly to an application. He requires access to only his home directory to retrieve and delete files.
- The user who uses an application as his shell but does not require FTP access to the system.
- An application that automatically transfers data to a remote system for processing by a second application.

It is necessary to find an elegant solution to each of these problems before that solution can be considered viable by an organization.

Scenario A

A user named Bob accesses a UNIX system through an application that is a replacement for his normal UNIX log-in shell. Bob has no need for, and does not have, direct UNIX command-line access. While using the application, Bob creates reports or other output that he must upload or download for analysis or processing. The application saves this data in either Bob's home directory or a common directory for all application users.

Bob may or may not require the ability to put files onto the application server. The requirements break down as follows:

- Bob requires FTP access to the target server.
- Bob requires access to a restricted number of directories, possibly one or two.
- Bob may or may not require the ability to upload files to the server.

Scenario B

Other application users in the environment illustrated in Scenario A require no FTP access whatsoever. Therefore, it is necessary to prevent them from connecting to the application server using FTP.

Scenario C

The same application used by the users in Scenarios A and B regularly dumps data to move to another system. The use of hard-coded passwords in scripts is not advisable because the scripts must be readable for them to

be executed properly. This may expose the passwords to unauthorized users and allow them to access the target system. Additionally, the use of hard-coded passwords makes it difficult to change the password on a regular basis because all scripts using this password must be changed.

A further requirement is to protect the data once stored on the remote system to limit the possibility of unauthorized access, retrieval, and modification of the data.

While there are a large number of options and directives for the `/etc/ ftpaccess` file, the focus here is on those that provide secured access to meet the requirements in the scenarios described.

Controlling FTP Access

Advanced FTP servers such as wu-ftpd provide extensive controls for controlling FTP access to the target system. This access does not extend to the IP layer, as the typical FTP client does not offer encryption of the data stream. Rather, FTP relies on the properties inherent in the IP (Internet Protocol) to recover from malformed or lost packets in the data stream. This means one still has no control over the network component of the data transfer. This may allow for the exposure of the data if the network is compromised. However, that is outside the scope of the immediate discussion.

wu-ftpd uses two control files: `/etc/ftpusers and /etc/ftpaccess`. The `/etc/ftpusers` file is used to list the users who do **not** have FTP access rights on the remote system. For example, if the `/etc/ftpusers` file is empty, then all users, including root, have FTP rights on the system. This is not the desired operation typically, because access to system accounts such as root are to be controlled. Typically, the `/etc/ftpusers` file contains the following entries:

- `root`
- `bin`
- `daemon`
- `adm`
- `lp`
- `sync`
- `shutdown`
- `halt`
- `mail`
- `news`
- `uucp`
- `operator`
- `games`
- `nobody`

When users in this list, root for example, attempt to access the remote system using FTP, they are denied access because their account is listed in the `/etc/ftpusers` file. This is illustrated in Exhibit 3.1.

By adding additional users to this list, one can control who has FTP access to this server. This does, however, create an additional step in the creation of a user account, but it is a related process and could be added as a step in the script used to create a user. Should a user with FTP privileges no longer require this access, the user's name can be added to the `/etc/ ftpusers` list at any time. Similarly, if a denied user requires this access in the future, that user can be removed from the list and FTP access restored.

Recall the requirements of Scenario B: the user has a log-in on the system to access his application but does not have FTP privileges. This scenario has been addressed through the use of `/etc/ftpusers`. The user can still have UNIX shell access or access to a UNIX-based application through the normal UNIX log-in process. However, using `/etc/ftpusers` prevents access to the FTP server and eliminates the problem of unauthorized data movement to or from the FTP server. Most current FTP server implementations offer the `/etc/ftpusers` feature.

EXHIBIT 3.1 Denying FTP Access

```
C:\WINDOWS>ftp 192.168.0.2
Connected to 192.168.0.2.
220 poweredge.home.com FTP server (Version wu-
 2.6.1(1) Wed Aug 9 05:54:50 EDT 20
00) ready.
User (192.168.0.2:(none)): root
331 Password required for root.
Password:
530 Login incorrect.
Login failed.
ftp>
```

Extending Control

Scenarios A and C require additional configuration because reliance on the extended features of the wu-ftpd server is required. These control extensions are provided in the file `/etc/ftpaccess`. A sample `/etc/ftpaccess` file is shown in Exhibit 3.2. This is the default `/etc/ftpaccess` file distributed with wu-ftpd. Before one can proceed to the problem at hand, one must examine the statements in the `/etc/ftpaccess` file. Additional explanation for other statements not found in this example, but required for the completion of our scenarios, are also presented later in the article.

The `class` statement in `/etc/ftpaccess` defines a class of users, in the sample file a user class named all, with members of the class being `real`, `guest`, and `anonymous`. The syntax for the class definition is:

```
class <class> <typelist> <addrglob> [<addrglob> ...]
```

`Typelist` is one of `real`, `guest`, or `anonymous`. The real keyword matches users to their real user accounts. `Anonymous` matches users who are using anonymous FTP access, while `guest` matches guest account access. Each of these classes can be further defined using other options in this file. Finally, the `class` statement can also identify the list of allowable addresses, hosts, or domains that connections will be accepted from. There can be multiple `class` statements in the file; the first one matching the connection will be used.

Defining the hosts requires additional explanation. The host definition is a domain name, a numeric address, or the name of a file, beginning with a slash ('/') that specifies additional address definitions. Additionally, the address specification may also contain IP `address:netmask` or IP address/CIDR definition. (CIDR, or Classless Internet Domain Routing, uses a value after the IP address to indicate the number of bits used for the network. A Class C address would be written as 192.168.0/24, indicating 24 bits are used for the network.)

It is also possible to exclude users from a particular class using a '!' to negate the test. Care should be taken in using this feature. The results of each of the class statements are OR'd together with the others, so it is possible to exclude an allowed user in this manner. However, there are other mechanisms available to deny connections from specific hosts or domains. The primary purpose of the class statement is to assign connections from specific domains or types of users to a class. With this in mind, one can interpret the class statement in Exhibit 3.2, shown here as:

```
class  all  real,guest,anonymous  *
```

This statement defines a `class` named `all`, which includes user types real, `anonymous`, and `guest`. Connections from any host are applicable to this class.

The `email` clause specifies the e-mail address of the FTP archive maintainer. It is printed at various times by the FTP server.

EXHIBIT 3.2 Sample `/etc/ftpaccess` File

```
class all real,guest,anonymous *

email root@localhost

loginfails 5

readme     README*     login
readme     README*     cwd=*

message /var/ftp/welcome.msg   login
message .message             cwd=*

compressyesall
taryesall
chmodnoguest,anonymous
deletenoguest,anonymous
overwritenoguest,anonymous
renamenoguest,anonymous

log transfers anonymous,real inbound,outbound

shutdown /etc/shutmsg

passwd-check rfc822 warn
```

The message clause defines a file to be displayed when the user logs in or when they change to a directory. The statement

```
message  /var/ftp/welcome.msg  login
```

causes wu-ftpd to display the contents of the file `/var/ftp/welcome.msg` when a user logs in to the FTP server. It is important for this file be somewhere accessible to the FTP server so that anonymous users will also be greeted by the message.

NOTE: Some FTP clients have problems with multiline responses, which is how the file is displayed.

When accessing the test FTP server constructed for this article, the message file contains:

```
***** WARNING *****
This is a private FTP server. If you do not have an account,
you are not welcome here.
******************
It is currently %T local time in Ottawa, Canada.
You are %U@%R accessing %L.
for help, contact %E.
```

The `%<char>` strings are converted to the actual text when the message is displayed by the server. The result is:

```
331 Password required for chare.
Password:
230-***** WARNING *****
230-This is a private FTP server. If you do not have an account,
230-you are not welcome here.
230-******************
230-It is currently Sun Jan 28 18:28:01 2001 local time in Ottawa,
 Canada.
```

EXHIBIT 3.3 `%char` Definitions

Tag	Description
`%T`	Local time (form Thu Nov 15 17:12:42 1990)
`%F`	Free space in partition of CWD (kbytes)
`%C`	Current working directory
`%E`	The maintainer's e-mail address as defined in ftpaccess
`%R`	Remote host name
`%L`	Local host name
`%u`	Username as determined via RFC931 authentication
`%U`	Username given at log-in time
`%M`	Maximum allowed number of users in this class
`%N`	Current number of users in this class
`%B`	Absolute limit on disk blocks allocated
`%b`	Preferred limit on disk blocks
`%Q`	Current block count
`%I`	Maximum number of allocated inodes (+1)
`%i`	Preferred inode limit
`%q`	Current number of allocated inodes
`%H`	Time limit for excessive disk use
`%h`	Time limit for excessive files
`%xu`	Uploaded bytes
`%xd`	Downloaded bytes
`%xR`	Upload/download ratio (1:n)
`%xc`	Credit bytes
`%xT`	Time limit (minutes)
`%xE`	Elapsed time since log-in (minutes)
`%xL`	Time left
`%xU`	Upload limit
`%xD`	Download limit

```
230-You are chare@chris accessing poweredge.home.com.
230-for help, contact root@localhost.
230-
230-
230 User chare logged in.
ftp>
```

The `%<char>` tags available for inclusion in the message file are listed in Exhibit 3.3.

It is allowable to define a class and attach a specific message to that class of users. For example:

```
classrealreal*
classanonanonymous*
message/var/ftp/welcome.msgloginreal
```

Now, the message is only displayed when a real user logs in. It is not displayed for either anonymous or guest users. Through this definition, one can provide additional information using other tags listed in Exhibit 3.3. The ability to display class-specific message files can be extended on a user-by-user basis by creating a class for each user. This is important because individual limits can be defined for each user.

The message command can also be used to display information when a user enters a directory. For example, using the statement

```
message /var/ftp/etc/.message CWD=*
```

EXHIBIT 3.4 Directory-Specific Messages

```
User (192.168.0.2:(none)): anonymous
331 Guest login ok, send your complete e-mail address
 as password.
Password:
230 Guest login ok, access restrictions apply.
ftp> cd etc
250-***** WARNING *****
250-There is no data of any interest in the /etc
 directory.
250-
250 CWD command successful.
ftp>
```

causes the FTP server to display the specified file when the user enters the directory. This is illustrated in Exhibit 3.4 for the anonymous user. The message itself is displayed only once to prevent annoying the user.

The `noretrieve` directive establishes specific files no user is permitted to retrieve through the FTP server. If the path specification for the file begins with a '/', then only those files are marked as nonretrievable. If the file specification does not include the leading '/', then any file with that name cannot be retrieved.

For example, there is a great deal of sensitivity with the password file on most UNIX systems, particularly if that system does not make use of a shadow file. Aside from the password file, there is a long list of other files that should not be retrievable from the system, even if their use is discouraged. The files that should be marked for nonretrieval are files containing the names:

- `passwd`
- `shadow`
- `.profile`
- `.netrc`
- `.rhosts`
- `.cshrc`
- `profile`
- `core`
- `.htaccess`
- `/etc`
- `/bin`
- `/sbin`

This is not a complete list, as the applications running on the system will likely contain other files that should be specifically identified.

Using the `noretrieve` directive follows the syntax:

```
noretrieve [absolute|relative] [class=<classname>] ...
[-] <file— name> <filename> ...
```

For example,

```
noretrieve passwd
```

prevents any user from downloading any file on the system named passwd.

When specifying files, it is also possible to name a directory. In this situation, all files in that directory are marked as nonretrievable. The option `absolute` or `relative` keywords identify if the file or directory is an absolute or relative path from the current environment. The default operation is to consider any file starting with a '/' as an absolute path. Using the optional `class` keyword on the `noretrieve` directive allows this

restriction to apply to only certain users. If the `class` keyword is not used, the restriction is placed against all users on the FTP server.

Denying Connections

Connections can be denied based on the IP address or domain of the remote system. Connections can also be denied based on how the user enters his password at log-in.

NOTE: This password check applies only to anonymous FTP users. It has no effect on real users because they authenticate with their standard UNIX password.

The password-check directive informs the FTP server to conduct checks against the password entered. The syntax for the password-check directive is

```
passwd-check <none|trivial|rfc822> (<enforce|warn>)
```

It is not recommended to use `password-check` with the none argument because this disables analysis of the entered password and allows meaningless information to be entered. The `trivial` argument performs only checking to see if there is an '@' in the password. Using the argument is the recommended action and ensures the password is compliant with the RFC822 e-mail address standard.

If the password is not compliant with the `trivial` or `rfc822` options, the FTP server can take two actions. The `warn` argument instructs the server to warn the user that his password is not compliant but still allows access. If the `enforce` argument is used, the user is warned and the connection terminated if a noncomplaint password is entered.

Use of the deny clause is an effective method of preventing access from specific systems or domains. When a user attempts to connect from the specified system or domain, the message contained in the specified file is displayed. The syntax for the `deny` clause is:

```
deny <addrglob> <message_file>
```

The file location must begin with a slash ('/'). The same rules described in the class section apply to the `addrglob` definition for the `deny` command. In addition, the use of the keyword `!nameservd` is allowed to deny connections from sites without a working nameserver.

Consider adding a deny clause to this file; for example, adding `deny!nameservd /var/ftp/.deny` to `/etc/ftpaccess`. When testing the deny clause, the denied connection receives the message contained in the file. Using the `!nameservd` definition means that any host not found in a reverse DNS query to get a host name from an IP address is denied access.

```
Connected to 192.168.0.2.
220 poweredge.home.com FTP server (Version wu-2.6.1(1)
 Wed Aug 9 05:54:50 EDT 20
00) ready.
User (192.168.0.2:(none)): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
530-**** ACCESS DENIED ****
530-
530-Access to this FTP server from your domain has been denied by the
 administrator.
530-
530 Login incorrect.
Login failed.
ftp>
```

The denial of the connection is based on where the connection is coming from, not the user who authenticated to the server.

EXHIBIT 3.5 Timeout Directives

Timeout Value	Default	Recommended
Timeout accept <seconds>	120	120
Timeout connect <seconds>	120	120
Timeout data <seconds>	1200	1200
Timeout idle <seconds>	900	900
Timeout maxidle <seconds>	7200	1200
Timeout RFC931 <seconds>	10	10

Connection Management

With specific connections denied, this discussion must focus on how to control the connection when it is permitted. A number of options for the server allow this and establish restrictions from throughput to access to specific files or directories.

Preventing anonymous access to the FTP server is best accomplished by removing the ftp user from the `/etc/passwd` file. This instructs the FTP server to deny all anonymous connection requests.

The `guestgroup` and guestuser commands work in a similar fashion. In both cases, the session is set up exactly as with anonymous FTP. In other words, a `chroot()` is done and the user is no longer permitted to issue the USER and PASS commands. If using `guestgroup`, the `groupname` must be defined in the `/etc/group` file; or in the case of `guestuser`, a valid entry in `/etc/passwd`.

```
guestgroup <groupname> [<groupname> ...]
guestuser <username> [<username> ...]
realgroup <groupname> [<groupname> ...]
realuser <username> [<username> ...]
```

In both cases, the user's home directory must be correctly set up. This is accomplished by splitting the home directory entry into two components separated by the characters '/./'. The first component is the base directory for the FTP server and the second component is the directory the user is to be placed in. The user can enter the base FTP directory but cannot see any files above this in the file system because the FTP server establishes a restricted environment.

Consider the `/etc/passwd` entry:

```
systemx:<passwd>:503:503:FTP Only Access from
systemx:/var/ftp/./systemx:/etc/ftponly
```

When systemx successfully logs in, the FTP server will `chroot("/var/ftp")` and then `chdir("/systemx")`. The guest user will only be able to access the directory structure under `/var/ftp` (which will look and act as `/` to systemx), just as an anonymous FTP user would.

Either an actual name or numeric ID specifies the group name. To use a numeric group ID, place a '%' before the number. Ranges may be given and the use of an asterisk means all groups. `guestuser` works like `guestgroup` except uses the username (or numeric ID).

`realuser` and `realgroup` have the same syntax but reverse the effect of `guestuser` and `guestgroup`. They allow real user access when the remote user would otherwise be determined a guest. For example:

```
guestuser *
realuser chare
```

causes all nonanonymous users to be treated as guest, with the sole exception of user chare, who is permitted real user access. Bear in mind, however, that the use of `/etc/ftpusers` overrides this directive. If the user is listed in `/etc/ftpusers`, he is denied access to the FTP server.

It is also advisable to set timeouts for the FTP server to control the connection and terminate it appropriately. The timeout directives are listed in Exhibit 3.5. The `accept` timeout establishes how long the FTP server will

wait for an incoming connection. The default is 120 seconds. The `connect` value establishes how long the FTP server will wait to establish an outgoing connection. The FTP server generally makes several attempts and will give up after the defined period if a successful connection cannot be established.

The data timeout determines how long the FTP server will wait for some activity on the data connection. This should be kept relatively long because the remote client may have a low-speed link and there may be a lot of data queued for transmission. The idle timer establishes how long the server will wait for the next command from the client. This can be overridden with the `–a` option to the server. Using the `access` clause overrides both the command-line parameter if used and the default.

The user can also use the `SITE IDLE` command to establish a higher value for the idle timeout. The `maxidle` value establishes the maximum value that can be established by the FTP client. The default is 7200 seconds. Like the idle timeout, the default can be overridden using the `–A` command-line option to the FTP server. Defining this parameter overrides the default and the command line. The last timeout value allows the maximum time for the RFC931 `ident/AUTH` conversation to occur. The information recorded from the RFC931 conversation is recorded in the system logs and used for any authentication requests.

Controlling File Permissions

File permissions in the UNIX environment are generally the only method available to control who has access to a specific file and what they are permitted to do with that file. It may be a requirement of a specific implementation to restrict the file permissions on the system to match the requirements for a specific class of users.

The `defumask` directive allows the administrator to define the umask, or default permissions, on a per-class or systemwide basis. Using the `defumask` command as

```
defumask 077
```

causes the server to remove all permissions except for the owner of the file. If running a general access FTP server, the use of a 077 umask may be extreme. However, umask should be at least 022 to prevent modification of the files by other than the owner.

By specifying a class of user following the umask, as in

```
defumask 077 real
```

all permissions are removed. Using these parameters prevents world writable files from being transferred to your FTP server. If required, it is possible to set additional controls to allow or disallow the use of other commands on the FTP server to change file permissions or affect the files. By default, users are allowed to change file permissions and delete, rename, and overwrite files. They are also allowed to change the umask applied to files they upload. These commands allow or restrict users from performing these activities.

```
chmod <yes|no> <typelist>
delete <yes|no> <typelist>
overwrite <yes|no> <typelist>
rename <yes|no> <typelist>
umask <yes|no> <typelist>
```

To restrict all users from using these commands, apply the directives as:

```
chmod no all
delete no all
overwrite no all
rename no all
umask no all
```

Setting these directives means no one can execute commands on the FTP server that require these privileges. This means the FTP server and the files therein are under the full control of the administrator.

Additional Security Features

There are a wealth of additional security features that should be considered when configuring the server. These control how much information users are shown when they log in about the server, and print banner messages among other capabilities.

The `greeting` directive informs the FTP server to change the level of information printed when the user logs in. The default is `full`, which prints all information about the server. A `full` message is:

```
220 poweredge.home.com FTP server (Version wu-2.6.1(1)
Wed Aug 9 05:54:50 EDT 2000) ready.
```

A `brief` message on connection prints the server name as:

```
220 poweredge.home.com FTP server ready.
```

Finally, the `terse` message, which is the preferred choice, prints only:

```
220 FTP server ready.
```

The `full` greeting is the default unless the greeting directive is defined. This provides the most information about the FTP server. The `terse` greeting is the preferred choice because it provides no information about the server to allow an attacker to use that information for identifying potential attacks against the server.

The greeting is controlled with the directive:

```
greeting <full|brief|terse>
```

An additional safeguard is the banner directive using the format:

```
banner <path>
```

This causes the text contained in the named file to be presented when the users connect to the server prior to entering their username and password. The path of the file is relative from the real root directory, not from the anonymous FTP directory. If one has a corporate log-in banner that is displayed when connecting to a system using Telnet, it would also be available to use here to indicate that the FTP server is for authorized users only.

NOTE: Use of this command can completely prevent noncompliant FTP clients from establishing a connection. This is because not all clients can correctly handle multiline responses, which is how the banner is displayed.

```
Connected to 192.168.0.2.
220-
220-* *
220-*                    * W A R N I N G **
220-* *
220-*ACCESS TO THIS FTP SERVER IS FOR AUTHORIZED USERS ONLY.*
220-*ALL ACCESS IS LOGGED AND MONITORED. IF YOU ARE NOT AN*
220-*AUTHORIZED USER, OR DO NOT AGREE TO OUR MONITORING POLICY,*
220-*DISCONNECT NOW.*
220-* *
```

```
220-*NO ABUSE OR UNAUTHORIZED ACCESS IS TOLERATED.*
220-*  *
220-
220-
220 FTP server ready.
User (192.168.0.2:(none)):
```

At this point, one has controlled how the remote user gains access to the FTP server, and restricted the commands they can execute and the permissions assigned to their files. Additionally, certain steps have been taken to ensure they are aware that access to this FTP server is for authorized use only. However, one must also take steps to record the connections and transfers made by users to fully establish what is being done on the FTP server.

Logging Capabilities

Recording information in the system logs is a requirement for proper monitoring of transfers and activities conducted on the FTP server. There are a number of commands that affect logging, and each is presented in this section. Normally, only connections to the FTP server are logged. However, using the `log commands` directive, each command executed by the user can be captured. This may create a high level of output on a busy FTP server and may not be required. However, it may be advisable to capture traffic for anonymous and guest users specifically. The directive syntax is:

```
log commands <typelist>
```

As with other directives, it is known that `typelist` is a combination of `real`, `anonymous`, and `guest`. If the `real` keyword is used, logging is done for users accessing FTP using their real accounts. `Anonymous` logs all commands performed by anonymous users, while `guest` matches users identified using the `guest-group` or `guestuser` directives.

Consider the line

```
log commands guest, anonymous
```

which results in all commands performed by anonymous and guest users being logged. This can be useful for later analysis to see if automated jobs are being properly performed and what files are uploaded or downloaded.

Like the `log commands` directive, `log transfers` performs a similar function, except that it records all file transfers for a given class of users. The directive is stated as:

```
log transfers <typelist> <directions>
```

The directions argument is `inbound` or `outbound`. Both arguments can be used to specify logging of transfers in both directions. For clarity, inbound are files transferred to the server, or uploads, and `outbound` are transfers from the server, or downloads. The `typelist` argument again consists of `real`, `anonymous`, and `guest`.

It is not only essential to log all of the authorized functions, but also to record the various command and requests made by the user that are denied due to security requirements. For example, if there are restrictions placed on retrieving the `password` file, it is desirable to record the security events. This is accomplished for `real`, `anonymous`, and `guest` users using the `log security` directive, as in:

```
log security <typelist>
```

If `rename` is a restricted command on the FTP server, the log security directive results in the following entries

```
Feb 11 20:44:02 poweredge ftpd[23516]: RNFR dayo.wav
Feb 11 20:44:02 poweredge ftpd[23516]: RNTO day-o.wav
```

```
Feb 11 20:44:02 poweredge ftpd[23516]: systemx of localhost.home.com
 [127.0.0.1]
tried to rename /var/ftp/systemx/dayo.wav to /var/ftp/systemx/day-o.wav
```

This identifies the user who tried to rename the file, the host that the user connected from, and the original and desired filenames. With this information, the system administrator or systems security personnel can investigate the situation.

Downloading information from the FTP server is controlled with the `noretrieve` clause in the `/etc/ftpaccess` file. It is also possible to limit uploads to specific directories. This may not be required, depending on the system configuration. A separate entry for each directory one wishes to allow uploads to is highly recommended. The syntax is:

```
upload [absolute|relative] [class=<classname>]... [-] <root-dir>
 <dirglob> <yes|no> <owner> <group> <mode> ["dirs"|"nodirs"] [<d_mode>]
```

This looks overly complicated, but it is in fact relatively simple. Define a directory called `<dirglob>` that permits or denies uploads. Consider the following entry:

```
upload /var/ftp /incoming yes ftpadmin ftpadmin 0440 nodirs
```

This means that for a user with the home directory of `/var/ftp`, allow uploads to the incoming directory. Change the owner and group to be ftpadmin and change the permissions to `readonly`. Finally, do not allow the creation of directories. In this manner, users can be restricted to the directories to which they can upload files. Directory creation is allowed by default, so one must disable it if required.

For example, if one has a user on the system with the following password file entry:

```
chare:x:500:500:Chris Hare:/home/chare:/bin/bash
```

and one wants to prevent the person with this `userid` from being able to upload files to his home directory, simply add the line:

```
upload /home/chare no
```

to the `/etc/ftpaccess` file. This prevents the user `chare` from being able to upload files to his home directory. However, bear in mind that this has little effect if this is a real user, because real users will be able to upload files to any directory they have write permission to. The `upload` clause is best used with anonymous and guest users.

NOTE: The wu-ftpd server denies anonymous uploads by default.

To see the full effect of the upload clause, one must combine its use with a guest account, as illustrated with the systemx account shown here:

```
systemx:x:503:503:FTP access from System X:/home/
systemx/./:/bin/false
```

Note in this password file entry the home directory path. This entry cannot be made when the user account is created. The `'/./'` is used by wu-ftpd to establish the `chroot` environment. In this case, the user is placed into his home directory, `/home/systemx`, which is then used as the base for his chroot file system. At this point, the guest user can see nothing on the system other than what is in his home directory.

Using the upload clause of

```
upload /home/chare yes
```

means the user can upload files to this home directory. When coupled with the `noretrieve` clause discussed earlier, it is possible to put a high degree of control around the user.

The Complete /etc/ftpaccess File

The discussion thus far has focused on a number of control directives available in the wu-ftpd FTP server. It is not necessary that these directives appear in any particular order. However, to further demonstrate the directives and relationships between those directives, the `/etc/ftpaccess` file is illustrated in Exhibit 3.6.

Revisiting the Scenarios

Recall the scenarios from the beginning of this article. This section reviews each scenario and defines an example configuration to achieve it.

Scenario A

A user named Bob accesses a UNIX system through an application that is a replacement for his normal UNIX log-in shell. Bob has no need for, and does not have, direct UNIX command-line access. While using the application, Bob creates reports or other output that he must retrieve for analysis. The application saves this data in either Bob's home directory or a common directory for all application users.

Bob may or may not require the ability to put files onto the application server. The requirements break down as follows:

- Bob requires FTP access to the target server.
- Bob requires access to a restricted number of directories, possibly one or two.
- Bob may or may not require the ability to upload files to the server.

Bob requires the ability to log into the FTP and access several directories to retrieve files. The easiest way to do this is to deny retrieval for the entire system by adding a line to `/etc/ftpaccess` as

```
noretrieve /
```

This marks every file and directory as nonretrievable. To allow Bob to get the files he needs, one must set those files or directories as such. This is done using the `allow-retrieve` directive. It has exactly the same syntax as the `noretrieve` directive, except that the file or directory is now retrievable. Assume that Bob needs to retrieve files from the `/tmp` directory. Allow this using the directive

```
allow-retrieve /tmp
```

When Bob connects to the FTP server and authenticates himself, he cannot get files from his home directory.

```
ftp> pwd
257 "/home/bob" is current directory.
ftp> get .xauth xauth
200 PORT command successful.
550 /home/chare/.xauth is marked unretrievable
```

However, Bob can retrieve files from the /tmp directory.

```
ftp> cd /tmp
250 CWD command successful.
ftp> pwd
257 "/tmp" is current directory.
ftp> get .X0-lock X0lock
200 PORT command successful.
150 Opening ASCII mode data connection for .X0-lock (11 bytes).
226 Transfer complete.
ftp: 12 bytes received in 0.00Seconds 12000.00Kbytes/sec.
ftp>
```

EXHIBIT 3.6 The /etc/ftpaccess File

```
#
# Define the user classes
#
class    all        real,guest *
class    anonymous  anonymous  *
class    real       real       *
#
# Deny connections from systems with no reverse DNS
# deny !nameservd /var/ftp/.deny
#
# What is the email address of the server
 administrator. Make sure
# someone reads this from time to time.
email root@localhost
#
# How many login attempts can be made before logging
 an error message and
# terminating the connection?
#
loginfails 5
greeting terse

readme   README*    login
readme   README*    cwd=*
#
# Display the following message at login
#
message /var/ftp/welcome.msg  login
banner /var/ftp/warning.msg
#
# display the following message when entering the
 directory
#
message .message              cwd=*

#
# ACCESS CONTROLS
#
# What is the default umask to apply if no other
 matching directive exists
#
defumask 022
chmod    no         guest,anonymous
delete   no         guest,anonymous
overwriteno         guest,anonymous
rename   no         guest,anonymous
# remove all permissions except for the owner if
 the user is a member of the
# real class
#
defumask 077real
guestuser   systemx
realuser    chare
#
```

EXHIBIT 3.6 The `/etc/ftpaccess` File *(continued)*

```
#establish timeouts
#
timeout  accept 120
timeout  connect 120
timeout  data 1200
timeout  idle 900
timeout  maxidel 1200

#
# establish non-retrieval
#
# noretrieve passwd
# noretrieve shadow
# noretrieve .profile
# noretrieve .netrc
# noretrieve .rhosts
# noretrieve .cshrc
# noretrieve profile
# noretrieve core
# noretrieve .htaccess
# noretrieve /etc
# noretrieve /bin
# noretrieve /sbin
noretrieve /
allow-retrieve /tmp

upload /home/systemx / no

#
# Logging
#
log commands anonymous,guest,real
log transfers anonymous,guest,real inbound,outbound
log security anonymous,real,guest

compress yes        all
tar      yes        all

shutdown /etc/shutmsg

passwd-check rfc822 warn
```

If Bob must be able to retrieve files from his home directory, an additional `allow-retrieve` directive is required:

```
class real real *
allow-retrieve /home/bob class=real
```

When Bob tries to retrieve a file from anywhere other than /tmp or his home directory, access is denied.

Additionally, it may be necessary to limit Bob's ability to upload files. If a user requires the ability to upload files, no additional configuration is required, as the default action for the FTP server is to allow uploads for real users. If one wants to prohibit uploads to Bob's home directory, use the upload directive:

```
upload /home/bob / no
```

This command allows uploads to the FTP server.

The objective of Scenario A has been achieved.

Scenario B

Other application users in the environment illustrated in Scenario A require no FTP access whatsoever. Therefore, it is necessary to prevent them from connecting to the application server using FTP.

This is done by adding those users to the `/etc/ftpaccess` file. Recall that this file lists a single user per line, which is checked. Additionally, it may be advisable to deny anonymous FTP access.

Scenario C

The same application used by the users in Scenarios A and B regularly dumps data to move to another system. The use of hard-coded passwords in scripts is not advisable because the scripts must be readable for them to be executed properly. This may expose the passwords to unauthorized users and allow them to access the target system. Additionally, the use of hard-coded passwords makes it difficult to change the password on a regular basis because all scripts using this password must be changed.

A further requirement is to protect the data once stored on the remote system to limit the possibility of unauthorized access, retrieval, and modification of the data.

Accomplishing this requires the creation of a guest user account on the system. This account will not support a log-in and will be restricted in its FTP abilities. For example, create a UNIX account on the FTP server using the source hostname, such as systemx. The password is established as a complex string but with the other compensating controls, the protection on the password itself does not need to be as stringent. Recall from an earlier discussion that the account resembles

```
systemx:x:503:503:FTP access from System X:/home/
 systemx/./:/bin/false
```

Also recall that the home directory establishes the real user home directory, and the ftp chroot directory. Using the upload command

```
upload /home/systemx / no
```

means that the systemx user cannot upload files to the home directory. However, this is not the desired function in this case. In this scenario, one wants to allow the remote system to transfer files to the FTP server. However, one does not want to allow for downloads from the FTP server. To do this, the command

```
noretrieve /
upload /home/systemx / yes
```

prevents downloads and allows uploads to the FTP server.

One can further restrict access by controlling the ability to rename, overwite, change permissions, and delete a file using the appropriate directives in the `/etc/ftpaccess` file:

```
chmodnoguest,anonymous
deletenoguest,anonymous
overwritenoguest,anonymous
renamenoguest,anonymous
```

Because the user account has no interactive privileges on the system and has restricted privileges on the FTP server, there is little risk involved with using a hard-coded password. While using a hard-coded password is

not considered advisable, there are sufficient controls in place to compensate for this. Consider the following controls protecting the access:

The user cannot retrieve files from the system.
The user can upload files.
The user cannot see what files are on the system and thus cannot determine the names of the files to block the system from putting the correct data on the server.
The user cannot change file permissions.
The user cannot delete files.
The user cannot overwrite existing files.
The user cannot rename files.
The user cannot establish an interactive session.
FTP access is logged.

With these compensating controls to address the final possibility of access to the system and the data using a password attack or by guessing the password, it will be sufficiently difficult to compromise the integrity of the data.

The requirements defined in the scenario have been fulfilled.

Summary

This discussion has shown how one can control access to an FTP server and allow controlled access for downloads or uploads to permit the safe exchange of information for interactive and automated FTP sessions. The extended functionality offered by the wu-ftpd FTP server provides extensive access, and preventative and detective controls to limit who can access the FTP server, what they can do when they can connect, and the recording of their actions.

4

Privacy in the Healthcare Industry

Kate Borten, CISSP

> *All that may come to my knowledge in the exercise of my profession or outside of my profession or in daily commerce with men, which ought not to be spread abroad, I will keep secret and will never reveal.*
>
> **— from the Hippocratic Oath**
> Hippocrates, "Father of Medicine," approximately 400 B.C.

Years ago, doctors worked alone, or with minimal support, and personally hand-wrote their patients' medical records. Sometimes the most intimate information was not even recorded. Doctors knew their patients as friends and neighbors and simply remembered many details. Patients paid doctors directly, sometimes in cash and sometimes in goods or services. There were no "middle men" involved. And the Hippocratic Oath served patients well.

But along the way to today's world, in which the healthcare delivery and payment systems are one of the nation's biggest industries, many intermediaries have arisen, and mass processing and computers have replaced pen, paper, and the locked desk drawer.

After all, there are so many players involved, private and public, delivering services and paying for them, all under complex conditions and formulas, that it is almost impossible for all but the smallest organizations to do business without some degree of automation. Think about the data trail in the following scenario.

Imagine that a person is covered by a health insurance plan and that person develops a respiratory problem. The person sees his primary care doctor who recommends a chest x-ray. The person visits his local radiology practice, perhaps at his nearby hospital, and has the x-ray. If all goes smoothly, the x-ray results are communicated back to the doctor who calls in a prescription to the pharmacy. Along the way, one may pay a co-payment or partial payment, but one expects that the bulk of the charges will be paid automatically by one's insurance plan. Sometime later, one may receive an "explanation of benefits" describing some of these services, how much was charged for them, and how much was paid by the insurance plan. But because one is not expected to respond, one files it without much thought or one might even throw it away.

Instead of limited and independent interactions between a patient and each provider (primary doctor, radiologist, pharmacist) in which the patient is provided with some healthcare service and pays for it directly, nowadays there is a complex intertwining of businesses behind the scenes, resulting in information about the patient being spread far and wide.

Consider who has acquired information about the patient, simply because of these few interactions with the healthcare system:

- The primary physician
- The primary physician's staff:
  - — The secretary or receptionist who checks in the patient and books a follow-up appointment when the patient leaves; may also book an appointment for the x-ray
  - — The nurse who takes blood pressure and other measurements and notes them in the patient's record

— The medical records personnel who pull the medical record before the appointment, make sure it is updated by the physician, and then re-file it
— The biller who compiles the demographic, insurance, and clinical information about the patient, which the insurance plan requires in order to pay the bill for this visit

- The radiologist
- The radiologist's staff:
  — The secretary/receptionist who checks the patient in
  — The technician who takes the x-rays
  — The medical/film records personnel who file the patient's record
  — The biller who compiles the demographic, insurance, and clinical information about this x-ray visit so that the radiologist gets paid by the insurer
- The hospital where the radiologist is based:
  — Business staff, including billers who compile the same information in order to bill the insurer for the hospital-based components of the radiology visit
  — Possibly additional medical records staff if the primary doctor is also part of the hospital and the hospital keeps a medical record for the patient
- The pharmacy:
  — The clerk who takes the message with the patient's name, doctor's information, and prescription
  — The pharmacist who fills the prescription
  — The clerk who interacts with the patient when picking up the prescription
  — The billing personnel who submit the patient's information to the insurer for payment
- The patient's insurance company:
  — The claims processing staff who receive separate claims from the primary physician, the radiologist, the hospital, and the pharmacy
  — Sometimes another, secondary insurance company or agency if bills are covered by more than one insurer

If the large number of people with the patient's private information is beginning to make one uneasy, consider these *additional* people who may have access to this information — often including the full set of demographic information, insurance information, diagnoses, procedures or tests performed, medications prescribed, etc.:

- Quality assurance personnel, typically hospital-based, who periodically review records
- Surveyors from national accreditation agencies who may read the patient's record as part of a review of the hospital
- Fund-raising personnel
- Marketing personnel or even marketing companies separate from the doctor, hospital, or pharmacy
- Researchers who may use detailed information about the patient for research studies

Now imagine that the patient's condition worsens and he or she is admitted to the hospital. The number of people with access to the information becomes a roaring crowd:

- The admitting department staff
- Dietary department staff
- Housekeeping staff
- All physicians at the hospital
- Medical students, residents, nursing students
- Pharmacy staff and students
- Social services staff and students
- State agencies to which the hospital reports all patient admissions

Finally, peel back another layer and note further access:

- Many information systems staff, including those supporting the healthcare applications, the databases, the servers, the network
- Many computer system vendors that provide customer support

- Numerous third-party businesses, such as:
  - — Transcriptionists who "key in" doctors' notes on patients
  - — Clearinghouses that transform the hospital's electronic data into acceptable formats for the insurance companies
  - — Law firms
  - — Auditors

What if instead of a simple respiratory condition, the patient's ailment results from HIV infection? Consider the case of the Washington (D.C.) Hospital Center. A patient's HIV status was revealed to his co-workers after a hospital employee failed to keep the information confidential. The jury ordered the hospital to pay $250,000 (P. Slevin, "Man Wins Suit over Disclosure of HIV Status," *The Washington Post*, December 30, 1999, p. B4).

Many people may feel that they have nothing sensitive in their records, nothing that would cause them embarrassment or could lead to discrimination. But even so, people should be entitled to basic protections and access controls. These are basic information security tenets, after all.

Rarely are people informed of how their personal health information is used or disseminated, and it is even more unusual that they are given a *choice* about it and an opportunity to restrict some uses.

Much of this information sharing is, in fact, legitimate and necessary. If people are to receive good healthcare, it is important that their caregivers have access to all relevant information. People generally accept that insurance companies will have access to information about them in order to pay their bills. But the industry has left the door wide open by passively permitting access (1) by many more individuals, and (2) to much more information than appropriate or necessary, thus violating the basic information security principle of least necessary privilege. People want their caregivers to have access, but not every caregiver at a given hospital. People understand that insurance companies need some information to ensure that the claims they are paying are legitimate, but it is not clear that they need access to as much personal detail as is common today.

Until recently, the healthcare industry generally lacked formal information security programs. There are several reasons for this. For one, many in the industry believe that there is little commercial value in medical data en masse, and, therefore, such organizations are not likely targets for theft. There are examples of highly visible individuals' records being exposed, but the industry has viewed them as exceptions. Tennis star Arthur Ashe took pains to keep his HIV-positive status secret, but it was leaked to the press by a healthcare worker. In fact, it is highly probably that individual privacy breaches occur regularly, but go undetected and perhaps without visible consequence to the patients. People now recognize that there definitely is commercial value in large databases of medical data from ordinary people, as noted by drug stores sharing their patient prescription records with pharmaceutical companies, for example.

Furthermore, hospitals and other healthcare providers have traditionally based their policies primarily on ethical values and an honor system alone, and have not implemented consistent, specific, written procedures and technical controls. After all, there has been an assumption that all doctors (and, by extension, their support staffs) are ethical, and no one would want to prevent access to a medical record when that patient is in crisis. Unfortunately, that approach does not scale well. In a small office where each person's behavior is under scrutiny, it may suffice with the addition of a few procedures and technical controls. But once an organization becomes large and multifunctional, this approach alone simply cannot provide assurance of the confidentiality, integrity, and availability of patient information.

While the lack of a formal security program protecting health data in the context of treatment and payment is disconcerting, many *secondary* uses of personal information are not even known to us, nor does one have any control over them.

As Simson Garfinkel asserts so chillingly in his book, *Database Nation: The Death of Privacy in the 21st Century*, never before has so much information about each one of us been gathered and used in ways we can barely imagine. Identity theft is a rapidly growing problem. Although not covered in this chapter, many resources are available that focus on the problem. (Government Web sites such as the Department of Justice's www.usdoj.gov, the Social Security Administration's www.ssa.gov, and the joint agency site www.consumer.gov all explore the topic of identity theft.) And although one may not clearly understand what is happening and the potential damage, there definitely is a growing sense in this country that one's privacy is very much at risk.

In September 1999, a *Wall Street Journal*/ABC poll asked Americans to identify their biggest concern about the twenty-first century. While economic, political, and environmental concerns might first come to mind, the most commonly cited response was the loss of personal privacy.

What does this mean in the context of healthcare? Examples abound showing that this concern is valid:

- Following routine tests by her doctor, an Orlando, Florida, woman received a letter from a drug company promoting its treatment for her high cholesterol ("Many Can Hear What You Tell Your Doctors: Records of Patients Are Not Kept Private," *Orlando Sentinel*, November 30, 1997, p. A1).
- A banker who served on his local health board compared patient information to his bank's loan information. He called due the mortgages of patients with cancer (*The National Law Journal*, May 30, 1994).
- In the course of investigating a mental health therapist for fraud, the FBI obtained patients' records. When the FBI discovered one of its own employees among those patients, it targeted the employee as unfit, forcing him into early retirement, although he was later found fit for employment (A. Rubin, "Records No Longer for Doctor's Eyes Only," *Los Angeles Times*, September 1, 1998, p. A1).

This reality has negative implications for healthcare. Dr. Donald Palmisano, a member of the American Medical Association's board of trustees states, "If the patient doesn't believe [his or her] medical information will remain confidential, then we won't get the information we need to make the diagnosis." (*The Boston Globe Magazine*, September 17, 2000, p. 7.)

Indeed, in January 1999, a survey by Princeton Survey Research Associates for the California Health Care Foundation concluded that 15 percent of U.S. adults have "done something out of the ordinary to keep personal medical information confidential. The steps people have taken to protect medical privacy include behaviors that may put their own health at risk ...." Those steps include "going to another doctor; ... not seeking care to avoid disclosure to an employer; giving inaccurate or incomplete information on medical history; and asking a doctor to not write down the health problem or record a less serious or embarrassing condition."

This loss of privacy and trust in the healthcare system is at last being forcefully addressed through federal legislation.

HIPAA

The Health Insurance Portability and Accountability Act (HIPAA) of 1996 has multiple objectives, one of which is cost-savings through standardization of the electronic transactions that flow between business partners in the healthcare system. Hence, at the time that that section of the HIPAA becomes effective, when an individual enrolls in a health insurance plan or seeks care resulting in a claim and payment, the relevant information will be transmitted via electronic records of a standard format, using standard code sets and unique, universal identifiers for employers, providers, and payers.

While standardization will reduce costs, Congress fortunately recognized that it will also increase risks to information security and privacy. As more personal health information than ever is captured in electronic form and, furthermore, in common formats, it becomes vastly easier for someone to inappropriately access and use our information. HIPAA does away with proprietary formats, so one loses some of the safety of "security through obscurity." While there may be direct benefits to letting one's doctor have access to all one's health information — from hospital records to pharmacies and labs all across the country — it could be very damaging or, at least, embarrassing for one's employer or a marketing company to have such easy access.

Therefore, Congress added both security and privacy requirements to this Act. The Act directed the U.S. Department of Health and Human Services (HHS) to develop information security regulations. And it directed Congress to pass health privacy legislation by August 1999, or else HHS would be required to step in and develop privacy regulations. Unfortunately, while a number of health privacy bills were debated in committee, none ever made it to the members of Congress for a vote. Thus, it fell to HHS to develop privacy regulations in addition to those for security. But HHS has limited authority and can regulate only healthcare providers and health insurance companies, essentially omitting many other businesses using health information, such as the transcription agency and law firm mentioned above. So, until a broad-scope health privacy law is passed by Congress, large gaps in our legal protections remain.

The HIPAA privacy rule was finalized in December 2000 and, barring intervention, the deadline for compliance for most covered organizations is February 2003. The HIPAA security rule was finalized on April 21, 2003. Covered entities have until April 21, 2005, to comply; small health plans have until April 21, 2006.

How do the security and privacy regulations relate to each other? Information security professionals generally recognize a common definition of information security as the assurance of confidentiality, integrity, and availability of protected resources. In the healthcare arena, confidentiality receives the most attention because of the perceived sensitivity of patient information. But the creators of the HIPAA security rule recognized the

full scope of security and mandated a comprehensive information security program. After all, the integrity of the results of one's lab tests and the availability of one's record of allergic reactions, for example, can be extremely important to one's health!

Hence, those organizations covered by HIPAA are responsible for implementing a formal information security program. On the other hand, the concept of privacy is centered primarily on the individual. Privacy laws specify what rights a person has regarding access to and control of information about oneself, and they describe the obligations organizations have in assuring those rights. Privacy requires information security, and in many ways they are two sides of the same coin.

Anticipating the challenge of crafting an appropriate and acceptable health privacy law, Congress called on the Secretary of HHS for recommendations. In 1997, then-Secretary Donna Shalala presented a report to Congress that she based on five principles. These principles are drawn from the fair information practices drawn up decades earlier by the U.S. Government.

The fair information practices were used as the foundation for the Fair Credit Reporting Act, which gives people the right to obtain a plain-language copy of their financial credit report (at little or no cost) and to have errors corrected through a straightforward process. They also form the basis for privacy laws in many European Union countries and other modern nations. However, in the United States, moves toward an all-encompassing federal privacy law in the 1970s were derailed due to fears of "Big Brother" or the government having too much control over people's personal information.

Secretary Shalala's five principles — which are also reflected in HHS's privacy rule — are these:

1. *Boundaries*. Information collected for one purpose cannot be used for a different purpose without the express consent of the individual.
2. *Consumer control*. Individuals have the right to a copy of their record, have the right to correct erroneous information in their record, and have a right to know how their information is being used and given to other organizations.
3. *Public responsibility*. There must be a fair balance between the rights of the individual and the public good. (In other words, there is not an absolute right to privacy.)
4. *Accountability*. There will be penalties for those who violate the rules.
5. *Security*. Organizations have an obligation to protect the personally identifiable information under their control.

The last principle is particularly significant in understanding the relationship between the HIPAA security and privacy requirements. This makes it clear that one cannot have privacy without security, particularly in the area of access controls. The HIPAA privacy rule from HHS tells us when access to a person's health information is appropriate, when it is not, when explicit consent is required, etc. It also requires adherence to the "minimum necessary" security principle, the creation of audit trails, and the security training of the workforce. These regulations can be translated directly into conventional security and access control mechanisms that make up an organization's formal security program — policies, procedures, physical and technical controls, and education. In fact, the privacy rule broadly reiterates the need for security safeguards and thus could be interpreted as *encompassing* the separate HIPAA security rule requirements.

Other Patient Privacy Laws

In 1999, President Clinton signed the Gramm-Leach-Bliley Act (GLB) into law with some reluctance. This law breaks down the legal barriers between the insurance, banking, and brokerage businesses, allowing them to merge and share information. It is assumed that this will provide rich marketing opportunities. However, despite privacy protections in GLB, individuals will not have control over much of that sharing of their detailed, personal information, sometimes including health information. Clinton pledged to give greater control to individuals and, with the HIPAA privacy rule, appears to have done so with health data, at least to some degree.

Turning to case law and privacy, the outcomes are uneven across the country, as described in *The Right to Privacy* by lawyers Ellen Alderman and Caroline Kennedy in 1995. But a case from 1991 involving the Princeton Medical Center and one of their surgeons who became HIV-positive makes a significant statement. The court found that medical center staff had breached the doctor's privacy when they looked up his medical information, although they were not responsible for his treatment. In other words, they accessed his information for other

than a professional "need to know," and the court agreed that this constituted a breach of privacy. For those in the information security field, this case confirms a basic tenet of information security.

With the advent of HIPAA and the growing sophistication of lawyers and judges in the realms of security and technology, one should expect more such lawsuits.

Technical Challenges in Complying with New Privacy Laws and Regulations

As healthcare organizations collectively review the HIPAA security and privacy requirements, several areas present technical challenges.

Lack of Granular Access Controls

One of the current technical issues is the lack of sufficiently granular controls in the applications to limit the access of authorized users. This issue has several facets.

First, while systems have long been capable of limiting access by function or by types of data through role-based access control, it is difficult to develop algorithms to limit access to only certain patients. For example, it is typical for patient registration clerks to have access to demographic and insurance data in order to record or update a patient's address or insurance plan. But they do not have access to a patient's lab tests or a doctor's notes about the patient's condition. On the other hand, they have access to the demographic and insurance data of *every patient* in that healthcare organization. Because that information is kept historically, that often means the registration clerk has access to thousands, if not millions, of personal records. That type of information is usually not considered particularly sensitive. People's names, addresses, and telephone numbers are commonly published in telephone books, and most people do not keep the name of their health insurance plan a secret. But, in fact, this information falls under the full protection of HIPAA and can put people at risk if left unprotected. Imagine a battered woman who is seeking treatment while she is in hiding. She willingly gives her temporary address to her doctor so that the doctor can contact her, and she has a reasonable expectation that this information will be kept private and not divulged to her former partner.

An even more disconcerting example of the lack of granular access control is the wide access to a person's actual medical information: diagnoses, test results, doctors notes, surgery or procedures performed, medications prescribed, etc. It is not unusual for all physicians at a hospital and their support staff to have access to the full historic database of patients — thousands or even millions of patients' records. The same is true of medical and other students, as well as numerous individuals in business functions such as billing and medical records.

If organizations recognize the risks in these instances, they most often react by indicating they are at the mercy of their application vendors and the products simply do not provide tighter controls. So organizations use compensating controls such as policies, procedures, and education to counteract system deficiencies. It is very common for healthcare organizations to require workforce members to sign a confidentiality agreement stating that they will not access information other than for a business need to know. That done, many organizations have been lulled into believing they have met their obligation to protect the confidentiality of health information.

Indeed, this is not a trivial problem to solve. In a small medical practice, it may be clear-cut; but in an academic medical center — arguably the most complex healthcare organization — it becomes very difficult to anticipate the circle of workforce professionals, support staff, and business and administrative personnel who should have access to any given patient's record.

This presents an exciting opportunity for system designers to develop creative solutions. For example, in Britain, a new system developed by Dr. Ross J. Anderson, University of Cambridge, and implemented in several hospitals uses a distinct access control list (ACL) for each patient. This ACL is maintained by the patient's primary doctor who can, for example, temporarily add names of consulting specialists as needed. Support staff are linked to their physicians and thereby gain access as appropriate.

An analogous context-based access control solution could be developed in the United States based on relationships with a given patient. For example, many health plans require a designated primary care physician as a gatekeeper for healthcare services. A growing number of healthcare applications allow for such a designation, as well as for consulting physicians. And hospital admitting systems have long allowed for designation of a referring physician, an admitting physician, and an attending physician. Thus, in addition to standard

role-based access control, an individual's access can be further limited to those patients with whom there is a relationship. But the solution must be easy to administer and must extend to the non-professionals who have broad access.

Even if only a rough algorithm were developed to define some subset of the total patient population, a "break the glass" technique could readily be applied. This would work as follows. If a physician needed to access the record of a patient beyond the usual circle of patients, a warning screen would appear with a message such as, "Are you sure you need to access this patient? This action will be recorded and reviewed." If the doctor proceeded to access the patient's record, an immediate alarm would sound; for example, a message would be sent to the security officer's pager, or that audit log record would be flagged for explicit follow-up on the next business day. This mechanism would serve as a powerful deterrent to inappropriate accesses.

The second facet of the granular access control problem has to do with the requirements of the HIPAA privacy rule. That rule states that organizations must ask patients for permission to use their data for each specific purpose, such as marketing. Patients may agree and later revoke that authorization. This suggests that applications, or even database access tools, may no longer freely access every patient's record in the database if the reason for the access is related to marketing. Before retrieving a record, the software must somehow determine this patient's explicit wishes. That is not a technically challenging problem, but identifying the *reason* for the access is. One can make assumptions based on the user's role, which is often defined in the security system. For example, if a user works in the marketing department and has authorizations based on that role, one might assume the purpose for the access is related to marketing. But this approach does not apply neatly across the spectrum of users. The most obvious example is the physician whose primary role is patient care, but who may also serve in an administrative or research function. Some vendors have attempted to solve this problem by asking the user, as he accesses a record, to pick the reason for the access from a list of choices. However, a self-selected reason would not be likely to qualify as a security control in the eyes of information security professionals or auditors.

Patient-Level Auditing

The lack of sufficiently granular access control as described above, combined with the human tendency toward curiosity, lead to a common problem of inappropriate "browsing" or looking up patient records for other than authorized business reasons. In the best light, this may be done because of sympathy and concern for a family member, friend, or colleague. At its worst, it may be done for malicious intent or for monetary gain. A group of Medicaid clerks were prosecuted for selling copies of recipients' financial resources to sales representatives of managed care companies (*Forbes*, May 20, 1996, p. 252).

This behavior obviously threatens the confidentiality of the data entrusted to healthcare organizations and the privacy of the particular patients. It is a problem of particular significance in the healthcare industry where simply reading a record can be extremely damaging to the patient.

When concerns about inappropriate browsing are so great that the hospital's own employees are reluctant to seek care there, stronger measures are called for. Years ago, one hospital with an in-house-developed online system added a patient-level audit capability to counteract this threat. Since then, other hospitals and some healthcare system vendors have incorporated this valuable security feature into their systems. It is conceptually different from a standard database audit trail or record of changes to information in that it records *all* access, regardless of whether the information was altered or not. Second, unlike a database audit trail, it is less important to record exactly what data was accessed beyond which patient was accessed. If a user looked up a neighbor's record although there was no business reason, the security rules were broken, regardless of how much information about the neighbor the user actually saw.

Inappropriate browsing is a fundamental privacy issue that organizations are required by HIPAA to address through information security techniques such as the patient-level audit trail. This audit trail is also used to inform patients, upon their request, of disclosures of their information for a variety of reasons, whether appropriate and authorized or not.

The technical challenges with this type of audit trail are the potential performance and storage impacts and the retrospective review of large volumes of audit trail records.

This type of audit trail must be in effect for every patient, not just selected individuals. Thus, it is easy to imagine that system performance could be degraded to an unacceptable level if this feature is not carefully designed. Similarly, the size of each audit record must be considered in terms of online storage space. As

computing power and storage become less expensive, these should not be major barriers. But the remaining technical challenge is for designers to provide tools for analyzing the masses of audit data to identify potential abuses. Under the HIPAA, it will no longer be sufficient to have these audit trails on hand when a problem arises; organizations will be expected to proactively monitor these files. Yet picking out the inappropriate access from the vast majority of appropriate record accesses is not yet simple or routine. Clever filters are needed to help us discern appropriate from inappropriate accesses.

Internet Use

The healthcare industry is rapidly embracing the Internet, somewhat surprisingly because it is not known for being an early adopter of new technologies. However, the Internet is enticing as a communications vehicle between providers and payers, among geographically separate parts of the same organization, and, ultimately, between the business and the consumer.

It has long been acknowledged that the Internet can be used with relative safety if transmissions are encrypted and if entities use strong, two-factor authentication. Indeed, those are the Internet-use requirements imposed by the HIPAA on the healthcare world.

The encryption requirement can be met today through numerous products and solutions using proven algorithms such as 3DES, RSA, and ECC — and AES in the near future. But the authentication requirement presents significant implementation challenges.

Many healthcare organizations today use tokens with PINs for reliable, two-factor authentication of remote users. But consider current and arising Internet business activities and it becomes apparent that this solution is not scalable to the healthcare industry's consumers, that is, the public. Yet the HIPAA does not release healthcare organizations from their duty to protect when the communications are with a patient or health insurance plan member.

Already there are examples of healthcare organizations interacting with patients and plan members via the Internet. Some hospitals permit patient access to test results and other medical record information. Some pharmacies permit patients to order prescription refills. Some insurance plans permit subscribers to update address and primary care physician designation. And e-mail communications between physicians or insurance plans and patients are becoming commonplace.

Ross Perot's Dallas company, Perot Systems, has a multimillion dollar contract with Harvard Pilgrim Health Care, a major Boston-area HMO, to "create an Internet-based 'HMO of the future.'" The first step in November 2000 was the unveiling of a Web site for employers and employees to enroll in the health plan. But in the future, "Perot envisions a system where hospitals, doctors, employers, members, and the HMO will ... be able to log on and update patient accounts..., 'a model for how medicine should be practiced in the 21st century.'" (L. Kowalczyk, "Perot's Model HMO: Billionaire, Harvard Pilgrim Eye Internet-Based System," *The Boston Globe*, March 8, 2000, p. D1).

But while these communications are often encrypted (although not always), they typically authenticate the patient or subscriber using only a static password or PIN. This is occurring even at healthcare facilities using two-factor authentication for their own workforce's dial-up access. How do they reconcile these significantly different levels of security? Today, many healthcare organizations are simply unaware of the HIPAA requirement or are hoping it will somehow not apply to communications with the public — which flies in the face of reason. That avoidance is due to the real or perceived high costs (in dollars and human resources) of implementing a two-factor authentication solution and extending it to all patients or plan members. Yet the volume of health-related Internet transactions and the variety of healthcare business uses are guaranteed to expand in the future. A few organizations, however, are beginning to consider how to achieve this security control within their strategic goals over the next few years.

The most feasible solution appears to be with the implementation of public key infrastructure (PKI) and digital certificates/signatures. Although some PKI supporters mistakenly claimed that the 1998 proposed HIPAA Security and Electronic Signature Standards *requires* the adoption of PKI, PKI as a cluster of interoperating technologies does appear to hold the most promise for strong remote authentication — along with encryption, non-repudiation, and message integrity — comprising a powerful set of security controls.

Consider the financial world and the possibilities for fraud when a credit or bank card is not visible to the merchant. While only a small percentage of all credit card transactions occur over the Internet (and so cards are not viewable), they make up the majority of the fraudulent cases. And according to Visa USA, "fraudulent orders account for 10 to 15 cents of every $100 spent online, compared to just 6 cents for every $100 spent at brick-and-

mortar stores." (*The Boston Globe*, October 9, 2000, p. C1,9.) A consumer's liability is minimal, but not the bank's. In 1999, American Express introduced its American Express Blue card with a chip intended to give greater security and assurance of identity (i.e., authentication of the cardholder), among other features. More recently, VISA has also begun issuing cards carrying chips. In both cases, card readers could be free to the consumer. As businesses with real dollars to lose take steps to prevent fraud, they move the PKI industry forward by forcing standardization, interoperability, and lower costs.

If our bank and credit cards become smart cards carrying our digital certificates, soon it may be standard for home and laptop computers to have smart card readers, and those readers will be able to handle a variety of cards. At first this may be the new-age equivalent of a wallet full of credit cards from each gas station and department store as people had decades ago; many businesses and organizations will issue their own smart cards through which they can be assured that a person is the true cardholder. After all, one must have the card in one's possession and one must know the secret PIN to use it.

And just as today people have a small number of multipurpose credit or debit cards, the electronic smart card will rapidly become multipurpose — recognized across banks and other financial institutions as well as by merchants — thus reducing the number of cards (and digital certificates and private keys) people hold. Because the financial infrastructure is already in place (notice the common network symbols on ATMs: NYCE, Cirrus, and others), this time the migration to a small number of standards and physical cards could happen "overnight."

At the NIST/NCSC 22nd Annual National Information Systems Security Conference in October 1999, information security experts predicted that smart cards carrying digital certificates plus a biometric such as a fingerprint will become the standard in three to five years. With HIPAA security and privacy compliance deadlines coming in early 2003, that should be just in time for adoption by the healthcare industry to help secure remote communications. Today's health plan and hospital identification cards will become tomorrow's smart cards, allowing patients and subscribers to update their own records, make appointments, get prescription refills — all at their own convenience and with the assurance that no one else can easily pose as that person and gain unlawful access to his records. After all, this is about privacy of one's personal information.

Conclusion

The healthcare industry has historically lagged behind many other sectors of the U.S. economy in recognizing the societal and business need for a formal information security program. At a time of increasing exposures — in part due to the rapid embracing of the Internet by the industry — and the public's heightened sensitivity to privacy issues, the advent of federal legislation, HIPAA, mandating security, and privacy controls pushes healthcare to the forefront. This is an exciting opportunity for the information security world to apply its knowledge and skills to an area that affects each one of us: our health.

Types of Information Security Controls

Harold F. Tipton

Security is generally defined as the freedom from danger or as the condition of safety. Computer security, specifically, is the protection of data in a system against unauthorized disclosure, modification, or destruction and protection of the computer system itself against unauthorized use, modification, or denial of service. Because certain computer security controls inhibit productivity, security is typically a compromise toward which security practitioners, system users, and system operations and administrative personnel work to achieve a satisfactory balance between security and productivity.

Controls for providing information security can be physical, technical, or administrative. These three categories of controls can be further classified as either preventive or detective. Preventive controls attempt to avoid the occurrence of unwanted events, whereas detective controls attempt to identify unwanted events after they have occurred. Preventive controls inhibit the free use of computing resources and therefore can be applied only to the degree that the users are willing to accept. Effective security awareness programs can help increase users' level of tolerance for preventive controls by helping them understand how such controls enable them to trust their computing systems. Common detective controls include audit trails, intrusion detection methods, and checksums.

Three other types of controls supplement preventive and detective controls. They are usually described as deterrent, corrective, and recovery. Deterrent controls are intended to discourage individuals from intentionally violating information security policies or procedures. These usually take the form of constraints that make it difficult or undesirable to perform unauthorized activities or threats of consequences that influence a potential intruder to not violate security (e.g., threats ranging from embarrassment to severe punishment).

Corrective controls either remedy the circumstances that allowed the unauthorized activity or return conditions to what they were before the

violation. Execution of corrective controls could result in changes to existing physical, technical, and administrative controls. Recovery controls restore lost computing resources or capabilities and help the organization recover monetary losses caused by a security violation.

Deterrent, corrective, and recovery controls are considered to be special cases within the major categories of physical, technical, and administrative controls; they do not clearly belong in either preventive or detective categories. For example, it could be argued that deterrence is a form of prevention because it can cause an intruder to turn away; however, deterrence also involves detecting violations, which may be what the intruder fears most. Corrective controls, on the other hand, are not preventive or detective, but they are clearly linked with technical controls when antiviral software eradicates a virus or with administrative controls when backup procedures enable restoring a damaged data base. Finally, recovery controls are neither preventive nor detective but are included in administrative controls as disaster recovery or contingency plans.

Because of these overlaps with physical, technical, and administrative controls, the deterrent, corrective, and recovery controls are not discussed further in this chapter. Instead, the preventive and detective controls within the three major categories are examined.

PHYSICAL CONTROLS

Physical security is the use of locks, security guards, badges, alarms, and similar measures to control access to computers, related equipment (including utilities), and the processing facility itself. In addition, measures are required for protecting computers, related equipment, and their contents from espionage, theft, and destruction or damage by accident, fire, or natural disaster (e.g., floods and earthquakes).

Preventive Physical Controls

Preventive physical controls are employed to prevent unauthorized personnel from entering computing facilities (i.e., locations housing computing resources, supporting utilities, computer hard copy, and input data media) and to help protect against natural disasters. Examples of these controls include:

- Backup files and documentation.
- Fences.
- Security guards.
- Badge systems.
- Double door systems.
- Locks and keys.
- Backup power.

- Biometric access controls.
- Site selection.
- Fire extinguishers.

Backup Files and Documentation. Should an accident or intruder destroy active data files or documentation, it is essential that backup copies be readily available. Backup files should be stored far enough away from the active data or documentation to avoid destruction by the same incident that destroyed the original. Backup material should be stored in a secure location constructed of noncombustible materials, including two-hour-rated fire walls. Backups of sensitive information should have the same level of protection as the active files of this information; it is senseless to provide tight security for data on the system but lax security for the same data in a backup location.

Fences. Although fences around the perimeter of the building do not provide much protection against a determined intruder, they do establish a formal no trespassing line and can dissuade the simply curious person. Fences should have alarms or should be under continuous surveillance by guards, dogs, or TV monitors.

Security Guards. Security guards are often stationed at the entrances of facilities to intercept intruders and ensure that only authorized persons are allowed to enter. Guards are effective in inspecting packages or other hand-carried items to ensure that only authorized, properly described articles are taken into or out of the facility. The effectiveness of stationary guards can be greatly enhanced if the building is wired with appropriate electronic detectors with alarms or other warning indicators terminating at the guard station. In addition, guards are often used to patrol unattended spaces inside buildings after normal working hours to deter intruders from obtaining or profiting from unauthorized access.

Badge Systems. Physical access to computing areas can be effectively controlled using a badge system. With this method of control, employees and visitors must wear appropriate badges whenever they are in access-controlled areas. Badge-reading systems programmed to allow entrance only to authorized persons can then easily identify intruders.

Double Door Systems. Double door systems can be used at entrances to restricted areas (e.g., computing facilities) to force people to identify themselves to the guard before they can be released into the secured area. Double doors are an excellent way to prevent intruders from following closely behind authorized persons and slipping into restricted areas.

Locks and Keys. Locks and keys are commonly used for controlling access to restricted areas. Because it is difficult to control copying of keys,

many installations use cipher locks (i.e., combination locks containing buttons that open the lock when pushed in the proper sequence). With cipher locks, care must be taken to conceal which buttons are being pushed to avoid a compromise of the combination.

Backup Power. Backup power is necessary to ensure that computer services are in a constant state of readiness and to help avoid damage to equipment if normal power is lost. For short periods of power loss, backup power is usually provided by batteries. In areas susceptible to outages of more than 15–30 min., diesel generators are usually recommended.

Biometric Access Controls. Biometric identification is a more sophisticated method of controlling access to computing facilities than badge readers, but the two methods operate in much the same way. Biometrics used for identification include fingerprints, handprints, voice patterns, signature samples, and retinal scans. Because biometrics cannot be lost, stolen, or shared, they provide a higher level of security than badges. Biometric identification is recommended for high-security, low-traffic entrance control.

Site Selection. The site for the building that houses the computing facilities should be carefully chosen to avoid obvious risks. For example, wooded areas can pose a fire hazard, areas on or adjacent to an earthquake fault can be dangerous and sites located in a flood plain are susceptible to water damage. In addition, locations under an aircraft approach or departure route are risky, and locations adjacent to railroad tracks can be susceptible to vibrations that can precipitate equipment problems.

Fire Extinguishers. The control of fire is important to prevent an emergency from turning into a disaster that seriously interrupts data processing. Computing facilities should be located far from potential fire sources (e.g., kitchens or cafeterias) and should be constructed of noncombustible materials. Furnishings should also be noncombustible. It is important that appropriate types of fire extinguishers be conveniently located for easy access. Employees must be trained in the proper use of fire extinguishers and in the procedures to follow should a fire break out.

Automatic sprinklers are essential in computer rooms and surrounding spaces and when expensive equipment is located on raised floors. Sprinklers are usually specified by insurance companies for the protection of any computer room that contains combustible materials. However, the risk of water damage to computing equipment is often greater than the risk of fire damage. Therefore, carbon dioxide extinguishing systems were developed; these systems flood an area threatened by fire with carbon dioxide, which suppresses fire by removing oxygen from the air. Although carbon

dioxide does not cause water damage, it is potentially lethal to people in the area and is now used only in unattended areas.

Current extinguishing systems flood the area with Halon, which is usually harmless to equipment and less dangerous to personnel than carbon dioxide. At a concentration of about 10%, Halon extinguishes fire and can be safely breathed by humans. However, higher concentrations can eventually be a health hazard. In addition, the blast from releasing Halon under pressure can blow loose objects around and can be a danger to equipment and personnel. For these reasons and because of the high cost of Halon, it is typically used only under raised floors in computer rooms. Because it contains chlorofluorocarbons, it will soon be phased out in favor of a gas that is less hazardous to the environment.

Detective Physical Controls

Detective physical controls warn protective services personnel that physical security measures are being violated. Examples of these controls include:

- Motion detectors.
- Smoke and fire detectors.
- Closed-circuit television monitors.
- Sensors and alarms.

Motion Detectors. In computing facilities that usually do not have people in them, motion detectors are useful for calling attention to potential intrusions. Motion detectors must be constantly monitored by guards.

Fire and Smoke Detectors. Fire and smoke detectors should be strategically located to provide early warning of a fire. All fire detection equipment should be tested periodically to ensure that it is in working condition.

Closed-Circuit Television Monitors. Closed-circuit televisions can be used to monitor the activities in computing areas where users or operators are frequently absent. This method helps detect individuals behaving suspiciously.

Sensors and Alarms. Sensors and alarms monitor the environment surrounding the equipment to ensure that air and cooling water temperatures remain within the levels specified by equipment design. If proper conditions are not maintained, the alarms summon operations and maintenance personnel to correct the situation before a business interruption occurs.

TECHNICAL CONTROLS

Technical security involves the use of safeguards incorporated in computer hardware, operations or applications software, communications

hardware and software, and related devices. Technical controls are sometimes referred to as logical controls.

Preventive Technical Controls

Preventive technical controls are used to prevent unauthorized personnel or programs from gaining remote access to computing resources. Examples of these controls include:

- Access control software.
- Antivirus software.
- Library control systems.
- Passwords.
- Smart cards.
- Encryption.
- Dial-up access control and callback systems.

Access Control Software. The purpose of access control software is to control sharing of data and programs between users. In many computer systems, access to data and programs is implemented by access control lists that designate which users are allowed access. Access control software provides the ability to control access to the system by establishing that only registered users with an authorized log-on ID and password can gain access to the computer system.

After access to the system has been granted, the next step is to control access to the data and programs residing in the system. The data or program owner can establish rules that designate who is authorized to use the data or program.

Antivirus Software. Viruses have reached epidemic proportions throughout the microcomputing world and can cause processing disruptions and loss of data as well as significant loss of productivity while cleanup is conducted. In addition, new viruses are emerging at an ever-increasing rate — currently about one every 48 hours. It is recommended that antivirus software be installed on all microcomputers to detect, identify, isolate, and eradicate viruses. This software must be updated frequently to help fight new viruses. In addition, to help ensure that viruses are intercepted as early as possible, antivirus software should be kept active on a system, not used intermittently at the discretion of users.

Library Control Systems. These systems require that all changes to production programs be implemented by library control personnel instead of the programmers who created the changes. This practice ensures separation of duties, which helps prevent unauthorized changes to production programs.

Passwords. Passwords are used to verify that the user of an ID is the owner of the ID. The ID-password combination is unique to each user and therefore provides a means of holding users accountable for their activity on the system.

Fixed passwords that are used for a defined period of time are often easy for hackers to compromise; therefore, great care must be exercised to ensure that these passwords do not appear in any dictionary. Fixed passwords are often used to control access to specific data bases. In this use, however, all persons who have authorized access to the data base use the same password; therefore, no accountability can be achieved.

Currently, dynamic or one-time passwords, which are different for each log-on, are preferred over fixed passwords. Dynamic passwords are created by a token that is programmed to generate passwords randomly.

Smart Cards. Smart cards are usually about the size of a credit card and contain a chip with logic functions and information that can be read at a remote terminal to identify a specific user's privileges. Smart cards now carry prerecorded, usually encrypted access control information that is compared with data that the user provides (e.g., a personal ID number or biometric data) to verify authorization to access the computer or network.

Encryption. Encryption is defined as the transformation of plaintext (i.e., readable data) into ciphertext (i.e., unreadable data) by cryptographic techniques. Encryption is currently considered to be the only sure way of protecting data from disclosure during network transmissions.

Encryption can be implemented with either hardware or software. Software-based encryption is the least expensive method and is suitable for applications involving low-volume transmissions; the use of software for large volumes of data results in an unacceptable increase in processing costs. Because there is no overhead associated with hardware encryption, this method is preferred when large volumes of data are involved.

Dial-Up Access Control and Callback Systems. Dial-up access to a computer system increases the risk of intrusion by hackers. In networks that contain personal computers or are connected to other networks, it is difficult to determine whether dial-up access is available or not because of the ease with which a modem can be added to a personal computer to turn it into a dial-up access point. Known dial-up access points should be controlled so that only authorized dial-up users can get through.

Currently, the best dial-up access controls use a microcomputer to intercept calls, verify the identity of the caller (using a dynamic password mechanism), and switch the user to authorized computing resources as requested. Previously, call-back systems intercepted dial-up callers, veri-

fied their authorization and called them back at their registered number, which at first proved effective; however, sophisticated hackers have learned how to defeat this control using call-forwarding techniques.

Detective Technical Controls

Detective technical controls warn personnel of violations or attempted violations of preventive technical controls. Examples of these include audit trails and intrusion detection expert systems, which are discussed in the following sections.

Audit Trails. An audit trail is a record of system activities that enables the reconstruction and examination of the sequence of events of a transaction, from its inception to output of final results. Violation reports present significant, security-oriented events that may indicate either actual or attempted policy transgressions reflected in the audit trail. Violation reports should be frequently and regularly reviewed by security officers and data base owners to identify and investigate successful or unsuccessful unauthorized accesses.

Intrusion Detection Systems. These expert systems track users (on the basis of their personal profiles) while they are using the system to determine whether their current activities are consistent with an established norm. If not, the user's session can be terminated or a security officer can be called to investigate. Intrusion detection can be especially effective in cases in which intruders are pretending to be authorized users or when authorized users are involved in unauthorized activities.

ADMINISTRATIVE CONTROLS

Administrative, or personnel, security consists of management constraints, operational procedures, accountability procedures, and supplemental administrative controls established to provide an acceptable level of protection for computing resources. In addition, administrative controls include procedures established to ensure that all personnel who have access to computing resources have the required authorizations and appropriate security clearances.

Preventive Administrative Controls

Preventive administrative controls are personnel-oriented techniques for controlling people's behavior to ensure the confidentiality, integrity, and availability of computing data and programs. Examples of preventive administrative controls include:

- Security awareness and technical training.
- Separation of duties.
- Procedures for recruiting and terminating employees.

- Security policies and procedures.
- Supervision.
- Disaster recovery, contingency, and emergency plans.
- User registration for computer access.

Security Awareness and Technical Training. Security awareness training is a preventive measure that helps users to understand the benefits of security practices. If employees do not understand the need for the controls being imposed, they may eventually circumvent them and thereby weaken the security program or render it ineffective.

Technical training can help users prevent the most common security problem — errors and omissions — as well as ensure that they understand how to make appropriate backup files and detect and control viruses. Technical training in the form of emergency and fire drills for operations personnel can ensure that proper action will be taken to prevent such events from escalating into disasters.

Separation of Duties. This administrative control separates a process into component parts, with different users responsible for different parts of the process. Judicious separation of duties prevents one individual from obtaining control of an entire process and forces collusion with others in order to manipulate the process for personal gain.

Recruitment and Termination Procedures. Appropriate recruitment procedures can prevent the hiring of people who are likely to violate security policies. A thorough background investigation should be conducted, including checking on the applicant's criminal history and references. Although this does not necessarily screen individuals for honesty and integrity, it can help identify areas that should be investigated further.

Three types of references should be obtained: (1) employment, (2) character, and (3) credit. Employment references can help estimate an individual's competence to perform, or be trained to perform, the tasks required on the job. Character references can help determine such qualities as trustworthiness, reliability, and ability to get along with others. Credit references can indicate a person's financial habits, which in turn can be an indication of maturity and willingness to assume responsibility for one's own actions.

In addition, certain procedures should be followed when any employee leaves the company, regardless of the conditions of termination. Any employee being involuntarily terminated should be asked to leave the premises immediately upon notification, to prevent further access to computing resources. Voluntary terminations may be handled differently, depending on the judgment of the employee's supervisors, to enable the employee to complete work in process or train a replacement.

All authorizations that have been granted to an employee should be revoked upon departure. If the departing employee has the authority to grant authorizations to others, these other authorizations should also be reviewed. All keys, badges, and other devices used to gain access to premises, information, or equipment should be retrieved from the departing employee. The combinations of all locks known to a departing employee should be changed immediately. In addition, the employee's log-on IDs and passwords should be canceled, and the related active and backup files should be either deleted or reassigned to a replacement employee.

Any special conditions to the termination (e.g., denial of the right to use certain information) should be reviewed with the departing employee; in addition, a document stating these conditions should be signed by the employee. All terminations should be routed through the computer security representative for the facility where the terminated employee works to ensure that all information system access authority has been revoked.

Security Policies and Procedures. Appropriate policies and procedures are key to the establishment of an effective information security program. Policies and procedures should reflect the general policies of the organization as regards the protection of information and computing resources. Policies should cover the use of computing resources, marking of sensitive information, movement of computing resources outside the facility, introduction of personal computing equipment and media into the facility, disposal of sensitive waste, and computer and data security incident reporting. Enforcement of these policies is essential to their effectiveness.

Supervision. Often, an alert supervisor is the first person to notice a change in an employee's attitude. Early signs of job dissatisfaction or personal distress should prompt supervisors to consider subtly moving the employee out of a critical or sensitive position.

Supervisors must be thoroughly familiar with the policies and procedures related to the responsibilities of their department. Supervisors should require that their staff members comply with pertinent policies and procedures and should observe the effectiveness of these guidelines. If the objectives of the policies and procedures can be accomplished more effectively, the supervisor should recommend appropriate improvements. Job assignments should be reviewed regularly to ensure that an appropriate separation of duties is maintained, that employees in sensitive positions are occasionally removed from a complete processing cycle without prior announcement, and that critical or sensitive jobs are rotated periodically among qualified personnel.

Disaster Recovery, Contingency, and Emergency Plans. The disaster recovery plan is a document containing procedures for emergency response,

extended backup operations, and recovery should a computer installation experience a partial or total loss of computing resources or physical facilities (or of access to such facilities). The primary objective of this plan, used in conjunction with the contingency plans, is to provide reasonable assurance that a computing installation can recover from disasters, continue to process critical applications in a degraded mode, and return to a normal mode of operation within a reasonable time. A key part of disaster recovery planning is to provide for processing at an alternative site during the time that the original facility is unavailable.

Contingency and emergency plans establish recovery procedures that address specific threats. These plans help prevent minor incidents from escalating into disasters. For example, a contingency plan might provide a set of procedures that defines the condition and response required to return a computing capability to nominal operation; an emergency plan might be a specific procedure for shutting down equipment in the event of a fire or for evacuating a facility in the event of an earthquake.

User Registration for Computer Access. Formal user registration ensures that all users are properly authorized for system and service access. In addition, it provides the opportunity to acquaint users with their responsibilities for the security of computing resources and to obtain their agreement to comply with related policies and procedures.

Detective Administrative Controls

Detective administrative controls are used to determine how well security policies and procedures are complied with, to detect fraud, and to avoid employing persons that represent an unacceptable security risk. This type of control includes:

- Security reviews and audits.
- Performance evaluations.
- Required vacations.
- Background investigations.
- Rotation of duties.

Security Reviews and Audits. Reviews and audits can identify instances in which policies and procedures are not being followed satisfactorily. Management involvement in correcting deficiencies can be a significant factor in obtaining user support for the computer security program.

Performance Evaluations. Regularly conducted performance evaluations are an important element in encouraging quality performance. In addition, they can be an effective forum for reinforcing management's support of information security principles.

Required Vacations. Tense employees are more likely to have accidents or make errors and omissions while performing their duties. Vacations contribute to the health of employees by relieving the tensions and anxieties that typically develop from long periods of work. In addition, if all employees in critical or sensitive positions are forced to take vacations, there will be less opportunity for an employee to set up a fraudulent scheme that depends on the employee's presence (e.g., to maintain the fraud's continuity or secrecy). Even if the employee's presence is not necessary to the scheme, required vacations can be a deterrent to embezzlement because the employee may fear discovery during his or her absence.

Background Investigations. Background investigations may disclose past performances that might indicate the potential risks of future performance. Background investigations should be conducted on all employees being considered for promotion or transfer into a position of trust; such investigations should be completed before the employee is actually placed in a sensitive position. Job applicants being considered for sensitive positions should also be investigated for potential problems. Companies involved in government-classified projects should conduct these investigations while obtaining the required security clearance for the employee.

Rotation of Duties. Like required vacations, rotation of duties (i.e., moving employees from one job to another at random intervals) helps deter fraud. An additional benefit is that as a result of rotating duties, employees are cross-trained to perform each other's functions in case of illness, vacation, or termination.

SUMMARY

Information security controls can be classified as physical, technical, or administrative. These are further divided into preventive and detective controls. Exhibit 1 lists the controls discussed in this chapter.

The organization's security policy should be reviewed to determine the confidentiality, integrity, and availability needs of the organization. The appropriate physical, technical, and administrative controls can then be selected to provide the required level of information protection, as stated in the security policy.

A careful balance between preventive and detective control measures is needed to ensure that users consider the security controls reasonable and to ensure that the controls do not overly inhibit productivity. The combination of physical, technical, and administrative controls best suited for a specific computing environment can be identified by completing a quantitative risk analysis. Because this is usually an expensive, tedious, and subjective process, however, an alternative approach — referred to as meeting the standard of due care — is often used. Controls that meet a standard of

PHYSICAL CONTROLS

Preventive

- Backup files and documentation
- Fences
- Security guards
- Badge systems
- Locks and keys
- Backup power
- Biometric access controls
- Site selection
- Fire extinguishers

Detective

- Motion detectors
- Smoke and fire detectors
- Closed-circuit television monitoring
- Sensors and alarms

TECHNICAL CONTROLS

Preventive

- Access control software
- Antivirus software
- Library control systems
- Passwords
- Smart cards
- Encryption
- Dial-up access control and callback systems

Detective

- Audit trails
- Intrusion-detection expert systems

ADMINISTRATIVE CONTROLS

Preventive

- Security awareness and technical training
- Separation of duties
- Procedures for recruiting and terminating employees
- Security policies and procedures
- Supervision
- Disaster recovery and contingency plans
- User registration for computer access

Detective

- Security reviews and audits
- Performance evaluations
- Required vacations
- Background investigations
- Rotation of duties

Exhibit 1. Information Security Controls

due care are those that would be considered prudent by most organizations in similar circumstances or environments. Controls that meet the standard of due care generally are readily available for a reasonable cost and support the security policy of the organization; they include, at the least, controls that provide individual accountability, auditability, and separation of duties.

When Technology and Privacy Collide

Edward H. Freeman

Payoff

Civil libertarians consider computer and communications technology to be a serious threat to individuals' personal privacy and freedom of speech. Some advocate laws to provide both an effective legal basis for accountability in the handling of personal data and procedures for redressing and compensating individuals. The development of the information superhighway may compromise personal privacy even more.

Problems Addressed

Data encryption refers to the methods used to prepare messages that cannot be understood without additional information. Government agencies, private individuals, civil libertarians, and the computer industry have all worked to develop methods of data encryption that will guarantee individual and societal rights.

The Clinton administration's proposed new standards for encryption technology—the Clipper Chip—was supposed to be the answer to the individual's concern for data security and the government's concern for law enforcement. Law-abiding citizens would have access to the encryption they need and the criminal element would be unable to use encryption to hide their illicit activity.

Cryptography and Secret Messages

Cryptography is the science of secure and secret communications. This security allows the sender to transform information into a coded message by using a secret key, a piece of information known only to the sender and the authorized receiver. The authorized receiver can decode the cipher to recover hidden information. If unauthorized individuals somehow receive the coded message, they should be unable to decode it without knowledge of the key.

The first recorded use of cryptography for correspondence was the Skytale created by the Spartans 2,500 years ago. The Skytale consisted of a staff of wood around which a strip of papyrus was tightly wrapped. The secret message was written on the parchment down the length of the staff. The parchment was then unwound and sent on its way. The disconnected letters made no sense unless the parchment was rewrapped around a staff of wood that was the same size as the first staff.

Methods of encoding and decoding messages have always been a factor in wartime strategies. The American effort that cracked Japanese ciphers during World War II played a major role in Allied strategy. At the end of the war, cryptography and issues of privacy remained largely a matter of government interest that were pursued by organizations such as the National Security Agency, which routinely monitors foreign communications.

Today, data bases contain extensive information about every individual's finances, health history, and purchasing habits. This data is routinely transferred or made accessible by telephone networks, often using an inexpensive personal computer and modem.

The government and private organizations realize—and individuals expect—certain standards to be met to maintain personal privacy. For example:

- Stored data should only be available to those individuals, organizations, and government agencies that have a need to know that information. Such information should not be available to others (e.g., the customer's employer) without the permission of the concerned individual.

- When organizations make decisions based on information received from a data base, the individual who is affected by such decisions should have the right to examine the data base and correct or amend any information that is incorrect or misleading. The misuse of information can threaten an individual's employment, insurance, and credit. If the facts of a previous transaction are in dispute, individuals should be able to explain their side of the dispute.

- Under strict constitutional and judicial guidelines and constraints, government agencies should have the right to collect information secretly as part of criminal investigations.

Existing Legislation

The Privacy Act of 1974

The Privacy Act of 1974 addressed some of these issues, particularly as they relate to government and financial activities. Congress adopted The Privacy Act to provide safeguards for an individual against an invasion of privacy. Under the Privacy Act, individuals decide what records kept by a federal agency or bureau are important to them. They can insist that this data be used only for the purposes for which the information was collected. Individuals have the right to see the information and to get copies of it. They may correct mistakes or add important details when necessary.

Federal agencies must keep the information organized so it is readily available. They must try to keep it accurate and up-to-date, using it only for lawful purposes. If an individual's rights are infringed upon under the Act, that person can bring suit in a federal district court for damages and a court order directing the agency to obey the law.

The Fair Credit Reporting Act of 1970

The Fair Credit Reporting Act of 1970 requires consumer reporting and credit agencies to disclose information in their files to affected consumers. Consumers have the right to challenge any information that may appear in their files. Upon written request from the consumer, the agency must investigate the completeness or accuracy of any item contained in that individual's files. The agency must then either remove the information or allow the consumer to file a brief statement setting forth the nature of the dispute.

Researchers are continuing to develop sophisticated methods to protect personal data and communications from unlawful interception. In particular, the development of Electronic Funds Transfer systems, where billions of dollars are transferred electronically, has emphasized the need to keep computerized communications accurate and confidential.

Privacy Rights

In short, the rapid advances in computer and communications technology have brought a new dimension to the individual's right to privacy. The power of today's computers, especially as it relates to record keeping, has the potential to destroy individual privacy rights.

Whereas most data is originally gathered for legitimate and appropriate reasons, “the mere existence of this vast reservoir of personal information constitutes a covert invitation to misuse.”[43]

---

[43] Sloan, I.J., ed., *Law of Privacy Rights in a Technological Society* (Dobbs Ferry, NY, Oceans Publications, 1986).

Personal liberty includes not only the freedom from physical restraint, but also the right to be left alone and to manage one's own affairs in a manner that may be most agreeable to that person, as long as the rights of others or of the public are respected. The word privacy does not even appear in the Constitution. When the Founders drafted the Bill of Rights, they realized that no document could possibly include all the rights that were granted to the American people.

After listing the specific rights in the first eight Amendments, the Founders drafted the Ninth Amendment, which declares, “The enumeration in this Constitution, of certain rights, shall not be construed to deny or disparage others retained by the people.” These retained rights are not specifically defined in the Constitution. The courts have pointed out that many rights are not specifically mentioned in the Constitution, but are derived from specific provisions. The Supreme Court held that several amendments already extended privacy rights. The Ninth Amendment then could be interpreted to encompass a right to privacy.

Federal Communications Act of 1934.
The federal laws that protect telephone and telegraphs from eavesdroppers are primarily derived from the Federal Communications Act of 1934. The Act prohibits any party involved in sending such communications from divulging or publishing anything having to do with its contents. It makes an exception and permits disclosure if the court has issued a legitimate subpoena. Any materials gathered through an illegal wiretap is inadmissable and may not be introduced as evidence in federal courts.

Data Encryption Standard

The National Bureau of Standards' Data Encryption Standard (DES), which specifies encryption procedures for computer data protection, has been a federal standard since 1977. The use of the DES algorithm was made mandatory for all financial transactions of the US government involving Electronic Funds Transfer, including those conducted by member banks of the Federal Reserve System.

The DES is a complex nonlinear ciphering algorithm that operates at high speeds when implemented in hardware. The DES algorithm converts 64 bits of plain text to 64 bits of cipher text under the action of a 56-bit keying parameter. The key is generated so that each of the 56 bits used directly by the algorithm is random. Each member of a group of authorized users of encrypted data must have the key that was used to encipher the data to use it. This technique strengthens the algorithm and makes it resistant to analysis.

Loopholes in the Traditional Methods of Data Encryption

The DES uses a 64-bit key that controls the transformation and converts information to ciphered code. There are a virtually infinite number of possible keys, so even the fastest computers would need centuries to try all possible keys.

Traditional encryption methods have an obvious loophole: their reliance on a single key to encode and decode messages. The privacy of coded messages is always a function of how carefully the decoder key is kept. When people exchange messages, however, they must find a way to exchange the key. This immediately makes the key vulnerable to interception. The problem is more complex when encryption is used on a large scale.

Diffle's Soution.

This problem was theoretically solved approximately 20 years ago, when an MIT student named Whitfield Diffle set out to plug this loophole. Diffle's solution was to give each user two separate keys, a public key and a private one. The public key could be widely distributed and the private key was known only to the user. A message encoded

with either key could be decoded with the other. If an individual sends a message scrambled with someone's public key, it can be decoded only with that person's private key.

The Clipper Controversy

In April 1993, the Clinton administration proposed a new standard for encryption technology, developed with the National Security Agency. The new standard is a plan called the Escrowed Encryption Standard. Under the standard, computer chips would use a secret algorithm called Skipjack to encrypt information. The Clipper Chip is a semiconductor device designed to be installed on all telephones, computer modems, and fax machines to encrypt voice communications.

The Clipper Chip

The Clipper Chip combines a powerful algorithm that uses an 80-bit encryption scheme and that is considered impossible to crack with today's computers within a normal lifetime. The chip also has secret government master keys built in, which would be available only to government agencies. Proper authorization, in the form of a court order, would be necessary to intercept communications.

The difference between conventional data encryption chips and the Clipper Chip is that the Clipper contains a law enforcement access field (LEAF). The LEAF is transmitted along with the user's data and contains the identity of the user's individual chip and the user's key—encrypted under the government's master key. This could stop eavesdroppers from breaking the code by finding out the user's key. Once an empowered agency knew the identity of the individual chip, it could retrieve the correct master key, use that to decode the user's key, and so decode the original scrambled information.

The Long Key.

Clipper uses a long key, which could have as many as 1,024 values. The only way to break Clipper's code would be to try every possible key. A single supercomputer would take a billion years to run through all of Clipper's possible keys.

Opponents of the the Clipper-Chip plan have criticized its implementation on several counts:

- Terrorists and drug dealers would circumvent telephones if they had the Clipper Chip. Furthermore, they might use their own chip.
- Foreign customers would not buy equipment from American manufacturers if they knew that their communications could be intercepted by US government agents.
- The integrity of the “back door” system could be compromised by unscrupulous federal employees.
- The remote possibility exists that an expert cryptologist could somehow break the code.

Recommended Action

Despite opposition from the computer industry and civil libertarians, government agencies are phasing in the Clipper technology for unclassified communications. Commercial use of Clipper is still entirely voluntary, and there is no guarantee it will be adopted by any organizations other than government ones. Yet several thousand Clipper-equipped telephones are currently on order for government use. The Justice Department is evaluating

proposals that would prevent the police and FBI from listening in on conversations without a warrant.

A possible solution to these concerns about privacy invasion would be to split the decryption key into two or more parts and give single parts to trustees for separate government agencies.

In theory, this would require the cooperation of several individuals and agencies before a message could be intercepted. This solution could compromise the secrecy needed to conduct a clandestine criminal investigation, but the Justice Department is investigating its feasibility.

No method of data encryption will always protect individual privacy and society's desire to stop criminal activities. Electronic Funds Transfer systems and the information superhighway have made the need for private communications more important than ever before. Society's problems with drugs and terrorism complicate the issues, highlighting the sensitive balance among the individual's right to privacy, society's need to protect itself, and everyone's fear of Big Brother government tools.

Author Biographies

Edward H. Freeman

Edward H. Freeman is an attorney, teacher, and lecturer in West Hartford CT, with 15 years' experience in data processing, most recently with a major insurance company. He is a part-time faculty member at Central Connecticut State University.

5

The Case for Privacy

Michael J. Corby, CISSP

> Any revelation of a secret happens by the mistake of [someone] who shared it in confidence.
>
> **— La Bruyere, 1645–1694**

It is probably safe to say that since the beginning of communication, back in prehistoric times, there were things that were to be kept private. From the location of the best fishing to the secret passage into the cave next door, certain facts were reserved only for a few knowledgeable friends. Maybe even these facts were so private that there was only one person in the world who knew them. We have made "societal rules" around a variety of things that we want to keep private or share only among a few, but still the concept of privacy expectations comes with our unwritten social code. And wherever there has been the code of privacy, there has been the concern over its violation. Have computers brought this on? Certainly not! Maintaining privacy has been important and even more important have been the methods used to try to keep that data a secret. Today in our wired society, however, we still face the same primary threat to privacy that has existed for centuries: mistakes and carelessness of the individuals who have been entrusted to preserve privacy — maybe even the "owner" of the data.

In the past few years, and heightened within the past few months, we have become more in tune to the cry — no, the public *outcry* — regarding the "loss of privacy" that has been forced upon us because of the information age. Resolving this thorny problem requires that we re-look at the way we design and operate our networked systems, and most importantly, that we re-think the way we allocate control to the rightful owners of the information which we communicate and store. Finally, we need to be careful about how we view the data that we provide and for which we are custodians.

Privacy and Control

The fact that data is being sent, printed, recorded, and shared is not the real concern of privacy. The real concern is that some data has been implied, by social judgment, to be private, for sharing only by and with the approval of its owner. If a bank balance is U.S.$1240, that is an interesting fact. If it happens to be my account, that is private information. I have, by virtue of my agreement with the bank, given them the right to keep track of my balance and to provide it *to me* for the purpose of keeping me informed and maintaining a control point with which I can judge their accuracy. I did not give them permission to share that balance with other people indiscriminately, nor did I give them permission to use that balance even subtly to communicate my standing in relation to others (i.e., publish a list of account holders sorted by balance).

The focal points of the issue of privacy are twofold:

1. How is the data classified as private?
2. What can be done to preserve the owner's (my) expectations of privacy?

Neither of these are significantly more challenging than, for example, sending digital pictures and sound over a telephone line. Why has this subject caused such a stir in the technology community? This chapter sheds some light on this issue and then comes up with an organized approach to resolve the procedural challenges of maintaining data privacy.

EXHIBIT 5.1 Types of Private Data

1. Static data:
   a. Who we are:
      i. Bio-identity (fingerprints, race, gender, height, weight)
      ii. Financial identity (bank accounts, credit card numbers)
      iii. Legal identity (Social Security number, driver's license, birth certificate, passport)
      iv. Social identity (church, auto clubs, ethnicity)
   b. What we have:
      i. Property (buildings, automobiles, boats, etc.)
      ii. Non-real property (insurance policies, employee agreements)
2. Dynamic data:
   a. Transactions (financial, travel, activities)
   b. How we live (restaurants, sporting events)
   c. Where we are (toll cards, cell phone records)
3. Derived data:
   a. Financial behavior (market analysis):
      i. Trends and changes (month-to-month variance against baseline)
      ii. Perceived response to new offerings (match with experience)
   b. Social behavior (profiling):
      i. Behavior statistics (drug use, violations or law, family traits)

Rudiments of Privacy

One place to start examining this issue is with a key subset of the first point on classifying data as private: what, exactly, is the data we are talking about? Start with the obvious: private data includes those facts that I can recognize as belonging to me, and for which I have decided reveal more about myself or my behavior than I would care to reveal. This includes three types of data loosely included in the privacy concerns of information technology (IT). These three types of data shown in Exhibit 5.1 are: static, dynamic, and derived data.

Static Data

Static data is pretty easy to describe. It kind of sits there in front of us. It does not move. It does not change (very often). Information that describes who we are, significant property identifiers, and other tangible elements is generally static. This information can of course take any form. It can be entered into a computer by a keyboard; it can be handwritten on a piece of paper or on a form; it can be photographed or created as a result of using a biological interface such as a fingerprint pad, retina scanner, voice or facial image recorder, or pretty much any way that information can be retained. It does not need to describe an animate object. It can also identify something we have. Account numbers, birth certificates, passport numbers, and employee numbers are all concepts that can be recorded and would generally be considered static data.

In most instances, we get to control the initial creation of static data. Because we are the one identifying ourselves by name, account number, address, driver's license number, or by speaking into a voice recorder or having our retina or face scanned or photographed, we usually will know when a new record is being made of our static data. As we will see later, we need to be concerned about the privacy of this data under three conditions: when we participate in its creation, when it is copied from its original form to a duplicate form, and when it is covertly created (created without our knowledge) such as in secretly recorded conversations or hidden cameras.

Dynamic Data

Dynamic data is also easy to identify and describe, but somewhat more difficult to control. Records of transactions we initiate constitute the bulk of dynamic data. It is usually being created much more frequently than static data. Every charge card transaction, telephone call, and bank transaction adds to the collection of

dynamic data. Even when we drive on toll roads or watch television programs, information can be recorded without our doing anything special. These types of transactions are more difficult for us to control. We may know that a computerized recording of the event is being made, but we often do not know what that information contains, nor if it contains more information than we suspect. Take, for example, purchasing a pair of shoes. You walk into a shoe store, try on various styles and sizes, make your selection, pay for the shoes, and walk out with your purchase in hand. You may have the copy of your charge card transaction, and you know that somewhere in the store's data files, one pair of shoes has been removed from their inventory and the price you just paid has been added to their cash balance. But what else might have been recorded? Did the sales clerk, for example, record your approximate age or ethnic or racial profile, or make a judgment as to your income level. Did you have children with you? Were you wearing a wedding band? What other general observations were made about you when the shoes were purchased? These items are of great importance in helping the shoe store replenish its supply of shoes, determining if they have attracted the type of customer they intended to attract and analyzing whether they are, in general, serving a growing or shrinking segment of the population. Without even knowing it, some information that you may consider private may have been used *without your knowledge* simply by the act of buying a new pair of shoes.

Derived Data

Finally, derived data is created by analyzing groups of dynamic transactions over time to build a profile of your behavior. Your standard way of living out your day, week, and month may be known by others even better than you may know it yourself. For example, you may, without even planning it, have dinner at a restaurant 22 Thursdays during the year. The other six days of the week, you may only dine out eight times in total. If you and others in your area fall into a given pattern, the restaurant community may begin to offer "specials" on Tuesday, or raise their prices slightly on Thursdays to accommodate the increased demand. In this case, your behavior is being recorded and used by your transaction partners in ways you do not even know or approve of. If you use an electronic toll recorder, as has become popular in many U.S. states, do you know if they are also computing the time it took to enter and exit the highway, and consequently your average speed? Most often, this derived data is being collected without even a hint to us, and certainly without our expressed permission.

Preserving Privacy

One place to start examining this issue is with a key subset of the first point on classifying data as private: what, exactly, is the data we are talking about? Start with the obvious: private data includes those items that we believe belong to us exclusively and it is not necessary for us to receive the product or service we wish to receive. To examine privacy in the context of computer technology today, we need to examine the following four questions:

1. Who owns the private data?
2. Who is responsible for security and accuracy?
3. Who decides how it can be used?
4. Does the owner need to be told when it is used or compromised?

> You already have zero privacy. Get over it.
>
> **— Scott McNealy, Chairman, Sun Microsystems, 1999**

Start with the first question about ownership. Cyber-consumers love to get offers tailored to them. Over 63 percent of the buying public in the United States bought from direct mail in 1998. Companies invest heavily in personalizing their marketing approach because it works. So what makes it so successful? By allowing the seller to know some pretty personal data about your preferences, a trust relationship is implied. (Remember that word "trust"; it will surface later.) The "real deal" is this: vendors do not know about your interests because they are your friend and want to make you happy. They want to take your trust and put together something private that will result in their product winding up in your home or office. Plain and simple: economics. And what does this cost them? If they have their way, practically nothing. You have given up your own private

information that they have used to exploit your buying habits or personal preferences. Once you give up ownership, you have let the cat out of the bag. Now they have the opportunity to do whatever they want with it.

"Are there any controls?" That brings us to the second question. The most basic control is to ask you clearly whether you want to give up something you own. That design method of having you "opt in" to their data collection gives you the opportunity to look further into their privacy protection methods, a stated or implied process for sharing (or not sharing) your information with other organizations and how your private information is to be removed. By simply adding this verification of your agreement, 85 percent of surveyed consumers would approve of having their profile used for marketing. Not that they ask, but they will be responsible for protecting your privacy. You must do some work to verify that they can keep their promise, but at least you know they have accepted some responsibility (their privacy policy should tell you how much). Their very mission will ensure accuracy. No product vendor wants to build its sales campaign on inaccurate data — at least not a second time.

Who decides use? If done right, both you and the marketer can decide based on the policy. If you are not sure if they are going to misuse their data, you can test them. Use a nickname, or some identifying initial to track where your profile is being used. I once tested an online information service by using my full middle name instead of an initial. Lo and behold, I discovered that my "new" name ended up on over 30 different mailing lists, and it took me several months to be removed from most of them. Some still are using my name, despite my repeated attempts to stop the vendors from doing so. Your method for deciding who to trust (there is that word again) depends on your preferences and the genre of services and products you are interested in buying. Vendors also tend to reflect the preferences of their customers. Those who sell cheap, ultra-low-cost commodities have a different approach than those who sell big-ticket luxuries to a well-educated executive clientele. Be aware and recognize the risks. Special privacy concerns have been raised in three areas: data on children, medical information, and financial information (including credit/debit cards). Be especially aware if these categories of data are collected and hold the collector to a more stringent set of protection standards. You, the public, are the judge.

If your data is compromised, it is doubtful that the collector will know. This situation is unfortunate. Even if it is known, it could cost them their business. Now the question of ethics comes into play. I actually know of a company that had its customer credit card files "stolen" by hackers. Rather than notify the affected customers and potentially cause a mass exodus to other vendors, the company decided to keep quiet. That company may be only buying some time. It is a far greater mistake to know that a customer is at risk and not inform them that they should check their records carefully than it is to have missed a technical component and, as a result, their system was compromised. The bottom line is that *you* are expected to report errors, inconsistencies, and suspected privacy violations to them. If you do, you have a right to expect immediate correction.

Where Is the Data to Be Protected?

Much ado has been made about the encryption of data while connected to the Internet. This is a concern; but to be really responsive to privacy directives, more than transmitting encrypted data is required. For a real privacy policy to be developed, the data must be protected when it is:

- Captured
- Transmitted
- Stored
- Processed
- Archived

That means more than using SSL or sending data over a VPN. It also goes beyond strong authentication using biometrics or public/private keys. It means developing a privacy architecture that protects data when it is sent, even internally; while stored in databases, with access isolated from those who can see other data in the same database; and while it is being stored in program work areas. All these issues can be solved with technology and should be discussed with the appropriate network, systems development, or data center managers. Despite all best efforts to make technology respond to the issues of privacy, the most effective use of resources and effort is in developing work habits that facilitate data privacy protection.

Good Work Habits

Privacy does not just happen. Everyone has certain responsibilities when it comes to protecting the privacy of one's own data or the data that belongs to others. In some cases, the technology exists to make that responsibility easier to carry out.

Vendor innovations continue to make this technology more responsive, for both data "handlers" and data "owners." For the owners, smart cards carry a record of personal activity that never leaves the wallet-sized token itself. For example, smart cards can be used to record selection of services (video, phone, etc.) without divulging preferences. They can maintain complex medical information (e.g., health, drug interactions) and can store technical information in the form of x-rays, nuclear exposure time (for those working in the nuclear industry), and tanning time (for those who do not).

For the handlers, smart cards can record electronic courier activities when data is moved from one place to another. They can enforce protection of secret data and provide proper authentication, either using a biometric such as a fingerprint or a traditional personal identification number (PIN). There are even cards that can scan a person's facial image and compare it to a digitized photo stored on the card. They are valuable in providing a digital signature that does not reside on one's office PC, subject to theft or compromise by office procedures that are less than effective.

In addition to technology, privacy can be afforded through diligent use of traditional data protection methods. Policies can develop into habits that force employees to understand the sensitivity of what they have access to on their desktops and personal storage areas. Common behavior such as protecting one's territory before leaving that area and when returning to one's area is as important as protecting privacy while in one's area.

Stories about privacy, the compromise of personal data, and the legislation (both U.S. and international) being enacted or drafted are appearing daily. Some are redundant and some are downright scary. One's mission is to avoid becoming one of those stories.

Recommendations

For all 21st-century organizations (and all people who work in those organizations), a privacy policy is a must and adherence to it is expected. Here are several closing tips:

1. If your organization has a privacy coordinator (or chief privacy officer), contact that person or a compliance person if you have questions. Keep their numbers handy.
2. Be aware of the world around you. Monitor national and international developments, as well as all local laws.
3. Be proactive; anticipate privacy issues before they become a crisis.
4. Much money can be made or lost by being ahead of the demands for privacy or being victimized by those who capitalize on your shortcomings.
5. Preserve your reputation and that of your organization. As with all bad news, violations of privacy will spread like wildfire. Everyone is best served by collective attention to maintaining an atmosphere of respect for the data being handled.
6. Communicate privacy throughout all areas of your organization.
7. Imbed privacy in existing processes — even older legacy applications.
8. Provide notification and allow your customers/clients/constituents to opt out or opt in.
9. Conduct audits and consumer inquiries.
10. Create a positive personalization image of what you are doing (how does this *really* benefit the data owner).
11. Use your excellent privacy policies and behavior as a competitive edge.

6

Biometric Identification

Donald R. Richards

Envision a day when the door to a secured office building can be opened using an automated system for identification based on a person's physical presence, although that person left his or her ID or access card on the kitchen counter at home. Imagine ticket-less airline travel, whereby a person can enter the aircraft based on a positive identification verified biometrically at the gateway. Picture getting into a car, starting the engine by flipping down the driver's visor, and glancing into the mirror and driving away, secure in the knowledge that only authorized individuals can make the vehicle operate.

The day when these actions are routine is rapidly approaching. Actually, implementation of fast, accurate, reliable, and user-acceptable biometric identification systems is already under way. Societal behavior patterns result in ever-increasing requirements for automated positive identification systems, and these are growing even more rapidly. The potential applications for these systems are limited only by a person's imagination. Performance claims cover the full spectrum from realistic to incredible. System implementation problems with these new technologies have been predictably high. User acceptance obstacles are on the rise. Security practitioners contemplating use of these systems are faced with overwhelming amounts of often contradictory information provided by manufacturers and dealers.

This chapter provides the security professional with the knowledge necessary to avoid potential pitfalls in selecting, installing, and operating a biometric identification system. The characteristics of these systems are introduced in sufficient detail to enable determination as to which are most important for particular applications. Historical problems experienced in organizational use of biometric systems are also discussed. Finally, the specific technologies available in the marketplace are described, including the data acquisition process, enrollment procedure, data files, user interface actions, speed, anti-counterfeit information, accuracy, and unique system aspects.

Background and History Leading to Biometric Development

Since the early days of mankind, humans have struggled with the problem of protecting their assets. How can unauthorized persons effectively and efficiently be prevented from making off with the things that are considered valuable, even a cache of food? Of course, the immediate solution then, as it has always been for the highest-value assets, was to post a guard. Then, as now, it was realized that the human guard is an inefficient and sometimes ineffective method of protecting resources.

The creation of a securable space, for example, a room with no windows or other openings except a sturdy door, was a step in the right direction. From there, the addition of the lock and key was a small but very effective move that enabled the removal of the continuous guard. Those with authorized access to the protected assets were given keys, which was the beginning of the era of identification of authorized persons based on the fact that they had such keys. Over centuries, locks and keys were successively improved to provide better security. The persistent problem was lost and stolen keys. When these events occurred, the only solution was the replacement of the lock (later just the cylinder) and of all keys, which was time consuming and expensive.

The next major breakthrough was the advent of electronic locks, controlled by cardreaders with plastic cards as keys. This continued the era of identification of authorized persons based on things that they had (e.g., coded plastic cards). The great advancement was the ability to electronically remove the ability of lost

or stolen (key) cards to unlock the door. Therefore, no locks or keys had to be changed, with considerable savings in time and cost. However, as time passed, experience proved that assets were sometimes removed before authorized persons even realized that their cards had been lost or stolen.

The addition of a Personal Identification Number (PIN) keypad to the cardreader was the solution to the unreported lost or stolen card problem. Thus began the era of identification of authorized persons based on things they had and on things they knew (e.g., a PIN). This worked well until the "bad guys" figured out that most people chose PINs that were easy for them to remember, such as birthdays, anniversaries, or other numbers significant in their lives. With a lost or stolen card, and a few trials, "bad guys" were sometimes successful in guessing the correct PIN and accessing the protected area.

The obvious solution was to use only random numbers as PINs, which solved the problem of PINs being guessed or found through trial and error. However, the difficulty in remembering random numbers caused another predictable problem. PINs (and passwords) were written on pieces of paper, Post-It notes, driver's licenses, blotters, bulletin boards, computers, or wherever they were convenient to find when needed. Sometimes they were written on the access cards themselves. In addition, because it is often easy to observe PINs being entered, "bad guys" planning a theft were sometimes able to obtain the number prior to stealing the associated card. These scenarios demonstrate that cardreaders, even those with PINs, cannot positively authenticate the identity of persons with authorized entry.

The only way to be truly positive in authenticating identity for access is to base the authentication on the physical attributes of the persons themselves (i.e., biometric identification). Because most identity authentication requirements take place when people are fully clothed (neck to feet and wrists), the parts of the body conveniently available for this purpose are the hands, face, and eyes.

Biometric Development

Once it became apparent that truly positive identification could only be based on the physical attributes of the person, two questions had to be answered. First, what part of the body could be used? Second, how could identification be accomplished with sufficient accuracy, reliability, and speed so as to be viable in field performance? However, had the pressures demanding automated personal identification not been rising rapidly at the highest levels (making necessary resources and funds available), this research would not have occurred.

At the time, the only measurable characteristic associated with the human body that was universally accepted as a positive identifier was the fingerprint. Contact data collected using special inks, dusting powders, and tape, for example, are matched by specially trained experts. Uniquely positioned whorls, ridge endings, and bifurcations were located and compared against templates. A sensor capable of reading a print made by a finger pressed against a piece of glass was required. Matching the collected print against a stored template is a classic computer task. Fortuitously, at the time these identification questions were being asked, computer processing capabilities and speed were increasing rapidly, while size and cost were falling. Had this not been the case, even the initial development of biometric systems would not have taken place. It has taken an additional 25 years of computer and biometric advancement, and cost reduction, for biometrics to achieve widespread acceptability and field proliferation.

Predictably, the early fingerprint-identifying verification systems were not successful in the marketplace, but not because they could not do what they were designed to do. They did. Key problems were the slow decision speed and the lack of ability to detect counterfeit fingerprints. Throughput of two to three people per minute results in waiting lines, personal frustration, and lost productive time. Failure to detect counterfeit input (i.e., rubber fingers, photo images) can result in false acceptance of impostors.

Continued comprehensive research and development and advancements in sensing and data processing technologies enabled production of systems acceptable in field use. Even these systems were not without problems, however. Some systems required high levels of maintenance and adjustment for reliable performance. Some required lengthy enrollment procedures. Some required data templates of many thousands of bytes, requiring large amounts of expensive storage media and slowing processing time. Throughput was still relatively slow (though acceptable). Accuracy rates (i.e., false accept and mostly false reject) were higher than would be acceptable today. However, automated biometric identifying verification systems were now performing needed functions in the field.

The value of fast, accurate, and reliable biometric identity verification was rapidly recognized, even if it was not yet fully available. Soon, the number of organized biometric research and development efforts exceeded

20. Many were fingerprint spinoffs: thumb print; full finger print; finger pattern (i.e., creases on the underside of the finger); and palm print. Hand topography (i.e., the side-view elevations of the parts of the hand placed against a flat surface) proved not sufficiently unique for accurate verification, but combined with a top view of the hand (i.e., hand geometry) it became one of the most successful systems in the field. Two-finger geometry is a recently marketed variation.

Other technologies that have achieved at least some degree of market acceptance include voice patterns, retina scan (i.e., the blood-vessel pattern inside the eyeball), signature dynamics (i.e., the speed, direction, and pressure of pen strokes), and iris recognition (i.e., the pattern of features in the colored portion of the eye around the pupil). Others that have reached the market, but have not remained, include keystroke dynamics (i.e., the measurable pattern of speed and time in typing words) and signature recognition (i.e., matching). Other physical characteristics that have been and are currently being investigated as potential biometric identifiers include finger length (though not sufficiently unique), wrist veins (underside), hand veins (back of the hand), knuckle creases (when grasping a bar), fingertip structure (blood vessel pattern under the skin), finger sections (between first and second joint), ear shape, and lip shape. One organization has been spending significant amounts of money and time investigating biometric identification based on body odor.

Another biometric identifying verification area receiving significant attention (and funding) is facial recognition. This partially results from the ease of acquiring facial images with standard video technology and from the perceived high payoff to be enjoyed by a successful facial recognition system. Facial thermography (i.e., heat patterns of the facial tissue) is an expensive variation because of high camera cost.

The history of the development of biometric identifying verification systems is far from complete. Entrepreneurs continue to see rich rewards for faster, more accurate and reliable technology, and advanced development will continue. However, advancements are expected to be improvements or variations of current technologies. These will be associated with the hands, eyes, and face for the "what we are" systems, and the voice and signature for the "what we do" systems.

Characteristics of Biometric Systems

These are the important factors necessary for any effective biometric system: accuracy, speed and throughput rate, acceptability to users, uniqueness of the biometric organ and action, resistance to counterfeiting, reliability, data storage requirements, enrollment time, intrusiveness of data collection, and subject and system contact requirements.

Accuracy

Accuracy is the most critical characteristic of a biometric identifying verification system. If the system cannot accurately separate authentic persons from impostors, it should not even be termed a biometric identification system.

False Reject Rate

The rate, generally stated as a percentage, at which authentic, enrolled persons are rejected as unidentified or unverified persons by a biometric system is termed the false reject rate. False rejection is sometimes called a Type I error. In access control, if the requirement is to keep the "bad guys" out, false rejection is considered the least important error. However, in other biometric applications, it may be the most important error. When used by a bank or retail store to authenticate customer identity and account balance, false rejection means that the transaction or sale (and associated profit) is lost, and the customer becomes upset. Most bankers and retailers are willing to allow a few false accepts as long as there are no false rejects.

False rejections also have a negative effect on throughput, frustrations, and unimpeded operations because they cause unnecessary delays in personnel movements. An associated problem that is sometimes incorrectly attributed to false rejection is failure to acquire. Failure to acquire occurs when the biometric sensor is not presented with sufficient usable data to make an authentic or impostor decision. Examples include smudged prints on a fingerprint system, improper hand positioning on a hand geometry system, improper alignment on a retina or iris system, or mumbling on a voice system. Subjects cause failure-to-acquire problems, either accidentally or on purpose.

False Accept Rate

The rate, generally stated as a percentage, at which unenrolled persons or impostors are accepted as authentic, enrolled persons by a biometric system is termed the false accept rate. False acceptance is sometimes called a Type II error. This is usually considered the most important error for a biometric access control system.

Crossover Error Rate (CER)

This is also called the equal error rate and is the point, generally stated as a percentage, at which the false rejection rate and the false acceptance rate are equal. This has become the most important measure of biometric system accuracy.

All biometric systems have sensitivity adjustment capability. If false acceptance is not desired, the system can be set to require (nearly) perfect matches of enrollment data and input data. If tested in this configuration, the system can truthfully be stated to achieve a (near) zero false accept rate. If false rejection is not desired, this system can be readjusted to accept input data that only approximates a match with enrollment data. If tested in this configuration, the system can be truthfully stated to achieve a (near) zero false rejection rate. However, the reality is that biometric systems can operate on only one sensitivity setting at a time.

The reality is also that when system sensitivity is set to minimize false acceptance, closely matching data will be spurned and the false rejection rate will go up significantly. Conversely, when system sensitivity is set to minimize false rejects, the false acceptance rate will go up notably. Thus, the published (i.e., truthful) data tells only part of the story. Actual system accuracy in field operations may even be less than acceptable. This is the situation that created the need for a single measure of biometric system accuracy.

The crossover error rate (CER) provides a single measurement that is fair and impartial in comparing the performance of the various systems. In general, the sensitivity setting that produces the equal error will be close to the setting that will be optimal for field operation of the system. A biometric system that delivers a CER of 2 percent will be more accurate than a system with a CER of 5 percent.

Speed and Throughput Rate

The speed and throughput rate are the most important biometric system characteristics. Speed is often related to the data processing capability of the system and is stated as how fast the accept or reject decision is annunciated. In actuality, it relates to the entire authentication procedure: stepping up to the system; inputting the card or PIN (if a verification system); inputting the physical data by inserting a hand or finger, aligning an eye, speaking access words, or signing a name; processing and matching of data files; annunciation of the accept or reject decision; and, if a portal system, moving through and closing the door.

Generally accepted standards include a system speed of five seconds from start-up through decision annunciation. Another standard is a portal throughput rate of six to ten/minute, which equates to six to ten seconds/person through the door. Only in recent years have biometric systems become capable of meeting these speed standards, and, even today, some marketed systems do not maintain this rapidity. Slow speed and the resultant waiting lines and movement delays have frequently caused the removal of biometric systems and even the failure of biometric companies.

Acceptability to Users

System acceptability to the people who must use it has been a little noticed but increasingly important factor in biometric identification operations. Initially, when there were few systems, most were of high security and the few users had a high incentive to use the systems; user acceptance was of little interest. In addition, little user threat was seen in fingerprint and hand systems.

Biometric system acceptance occurs when those who must use the system — organizational managers and any union present — all agree that there are assets that need protection, the biometric system effectively controls access to these assets, system usage is not hazardous to the health of the users, system usage does not inordinately impede personnel movement and cause production delays, and the system does not enable management to collect personal or health information about the users. Any of the parties can effect system success or removal. Uncooperative users will overtly or covertly compromise, damage, or sabotage system equipment. The cost of union inclusion of the biometric system in their contracts may become too costly. Moreover, management has the final decision on whether the biometric system benefits outweigh its liabilities.

Uniqueness of Biometric Organ and Action

Because the purpose of biometric systems is positive identification of personnel, some organizations (e.g., elements of the government) are specifying systems based only on a unique (i.e., no duplicate in the world) physical characteristic. The rationale is that when the base is a unique characteristic, a file match is a positive identification rather than a statement of high probability that this is the right person. Only three physical characteristics or human organs used for biometric identification are unique: the fingerprint, the retina of the eye (i.e., the blood-vessel pattern inside the back of the eyeball), and the iris of the eye (i.e., random pattern of features in the colored portion of the eye surrounding the pupil). These features include freckles, rings, pits, striations, vasculature, coronas, and crypts.

Resistance to Counterfeiting

The ability to detect or reject counterfeit input data is vital to a biometric access control system meeting high security requirements. These include use of rubber, plastic, or even hands or fingers of the deceased in hand or fingerprint systems, and mimicked or recorded input to voice systems. Entertainment media, such as the James Bond or Terminator films, have frequently shown security system failures when the heads or eyes of deceased (i.e., authentic) persons were used to gain access to protected assets or information. Because most of the early biometric identifying verification systems were designed for high-security access control applications, failure to detect or reject counterfeit input data was the reason for several system or organization failures. Resistance to counterfeit data remains a criterion of high-quality, high-accuracy systems. However, the proliferation of biometric systems into other non-high-security type applications means that lack of resistance to counterfeiting is not likely to cause the failure of a system in the future.

Reliability

It is vital that biometric identifying verification systems remain in continuous, accurate operation. The system must allow authorized persons access while precluding others, without breakdown or deterioration in performance accuracy or speed. In addition, these performance standards must be sustained without high levels of maintenance or frequent diagnostics and system adjustments.

Data Storage Requirements

Data storage requirements are a far less significant issue today than in the earlier biometric systems when storage media were very expensive. Nevertheless, the size of biometric data files remains a factor of interest. Even with current ultra-high-speed processors, large data files take longer to process than small files, especially in systems that perform full identification, matching the input file against every file in the database. Biometric file size varies between 9 and 10,000 bytes, with most falling in the 256- to 1000-byte range.

Enrollment Time

Enrollment time is also a less significant factor today. Early biometric systems sometimes had enrollment procedures requiring many repetitions and several minutes to complete. A system requiring a five-minute enrollment instead of two minutes causes 50 hours of expensive nonproductive time if 1000 users must be enrolled. Moreover, when line waiting time is considered, the cost increases several times. The accepted standard for enrollment time is two minutes per person. Most of the systems in the marketplace today meet this standard.

Intrusiveness of Data Collection

Originally, this factor developed because of user concerns regarding collection of biometric data from inside the body, specifically the retina inside the eyeball. Early systems illuminated the retina with a red light beam. However, this coincided with increasing public awareness of lasers, sometimes demonstrated as red light beams cutting steel. There has never been an allegation of user injury from retina scanning, but user sensitivity expanded from resistance to red lights intruding inside the body to include any intrusion inside the body. This user sensitivity has now increased to concerns about intrusions into perceived personal space.

Subject and System Contact Requirements

This factor could possibly be considered as a next step or continuation of intrusiveness. Indications are that biometric system users are becoming increasingly sensitive to being required to make firm physical contact with surfaces where up to hundreds of other unknown (to them) persons are required to make contact for biometric data collection. These concerns include voice systems that require holding and speaking into a handset close to the lips.

There seems to be some user feeling that "if I choose to do something, it is OK, but if an organization, or society, requires me to do the same thing, it is wrong." Whether or not this makes sense, it is an attitude spreading through society that is having an impact on the use of biometric systems. Systems using video camera data acquisition do not fall into this category.

Historical Biometric Problems

A variety of problems in the field utilization of biometric systems over the past 25 years have been identified. Some have been overcome and are seldom seen today; others still occur. These problems include performance, hardware and software robustness, maintenance requirements, susceptibility to sabotage, perceived health maladies because of usage, private information being made available to management, and skill and cooperation required to use the system.

Performance

Field performance of biometric identifying verification systems is often different from from experienced in manufacturers' or laboratory tests. There are two ways to avoid being stuck with a system that fails to deliver promised performance. First, limit consideration to technologies and systems that have been tested by an independent, unbiased testing organization. Sandia National Laboratories, located in Albuquerque, New Mexico, has done biometric system testing for the Department of Energy for many years, and some of their reports are available. Second, any system manufacturer or sales representative should be able to provide a list of organizations currently using their system. They should be able to point out those users whose application is similar to that currently contemplated (unless the planned operation is a new and unique application). Detailed discussions, and perhaps a site visit, with current users with similar application requirements should answer most questions and prevent many surprises.

Hardware and Software Robustness

Some systems and technologies that are very effective with small- to medium-sized user databases have a performance that is less than acceptable with large databases. Problems that occur include system slowdown and accuracy degradation. Some biometric system users have had to discard their systems and start over because their organizations became more successful, grew faster than anticipated, and the old system could not handle the growth. If they hope to "grow" their original system with the organization, system managers should at least double the most optimistic growth estimate and plan for a system capable of handling that load.

Another consideration is hardware capability to withstand extended usage under the conditions expected. An example is the early signature dynamics systems, which performed adequately during testing and early fielding periods. However, the pen and stylus sensors used to detect stroke direction, speed, and pressure were very tiny and sensitive. After months or a year of normal public use, the system performance had deteriorated to the point that the systems were no longer effective identifiers.

Maintenance Requirements

Some sensors and systems have required very high levels of preventive maintenance or diagnostics and adjustment to continue effective operations. Under certain operating and user conditions (e.g., dusty areas or with frequent users of hand lotions or creams), some fingerprint sensors needed cleaning as frequently as every day to prevent deterioration of accuracy. Other systems demanded weekly or monthly connection of diagnostic equipment, evaluation of performance parameters, and careful adjustment to retain productive performance. These human interventions not only disrupt the normal security process, but significantly increase operational costs.

Susceptibility to Sabotage

Systems with data acquisition sensors on pedestals protruding far out from walls or with many moving parts are often susceptible to sabotage or disabling damage. Spinning floor polisher handles or hammers projecting out of pockets can unobtrusively or accidentally affect sensors. These incidents have most frequently occurred when there was widespread user or union resistance to the biometric system.

Perceived Health Maladies Due to Usage

As new systems and technologies were developed and public sensitivity to new viruses and diseases such as AIDS, Ebola, and *E. coli* increased by orders of magnitude, acceptability became a more important issue. Perceptions of possible organ damage and potential spread of disease from biometric system usage ultimately had such a devastating effect on sales of one system that it had to be totally redesigned. Although thousands of the original units had been successfully fielded, whether or not the newly packaged technology regains popularity or even survives remains to be seen. All of this occurred without even one documented allegation of a single user becoming sick or injured as a result of system utilization.

Many of the highly contagious diseases recently publicized can be spread by simple contact with a contaminated surface. As biometric systems achieve wider market penetration in many applications, user numbers are growing logarithmically. There are developing indications that users are becoming increasingly sensitive about systems and technologies that require firm physical contact for acquisition of the biometric data.

Private Information Made Available to Management

Certain health events can cause changes in the blood vessel pattern (i.e., retina) inside the eyeball. These include diabetes and strokes. Allegations have been made that the retina-based biometric system enables management to improperly obtain health information that may be used to the detriment of system users. The scenario begins with the system failing to identify a routine user. The user is easily authenticated and re-enrolled. As a result, management will allegedly note the re-enrollment report and conclude that this user had a minor health incident (minor because the user is present the next working day). In anticipation that this employee's next health event could cause major medical cost, management might find (or create) a reason for termination. Despite the fact that there is no recorded case of actual occurrence of this alleged scenario, this folklore continues to be heard within the biometric industry.

Skill and Cooperation Required to Use the System

The performance of some biometric systems is greatly dependent on the skill or careful cooperation of the subject in using the system. Although there is an element of this factor required for data acquisition positioning for all biometric systems, it is generally attributed to the "what we do" type of systems.

Benefits of Biometric Identification as Compared with Card Systems

Biometric identifying verification systems control people. If the person with the correct hand, eye, face, signature, or voice is not present, the identification and verification cannot take place and the desired action (i.e., portal passage, data or resource access) does not occur.

As has been demonstrated many times, adversaries and criminals obtain and successfully use access cards, even those that require the addition of a PIN. This is because these systems control only pieces of plastic (and sometimes information), rather than people. Real asset and resource protection can only be accomplished by people, not cards and information, because unauthorized persons can (and do) obtain the cards and information.

Further, life-cycle costs are significantly reduced because no card or PIN administration system or personnel are required. The authorized person does not lose physical characteristics (i.e., hands, face, eyes, signature, or voice), but cards and PINs are continuously lost, stolen, or forgotten. This is why card access systems require systems and people to administer, control, record, and issue (new) cards and PINs. Moreover, the cards are an expensive and recurring cost.

Card System Error Rates

The false accept rate is 100 percent when the access card is in the wrong hands, lost, or stolen. It is a false reject when the right card is swiped incorrectly or just does not activate the system. (Think about the number of times to retry hotel room access cards to get the door to unlock.) Actually, it is also a false reject when a card is forgotten and that person cannot get through the door.

Biometric Data Updates

Some biometric systems, using technologies based on measuring characteristics and traits that may vary over time, work best when the database is updated with every use. These are primarily the "what we do" technologies (i.e., voice, signature, and keystroke). Not all systems do this. The action measured by these systems changes gradually over time. The voice changes as people age. It is also affected by changes in weight and by certain health conditions. Signature changes over time are easily documented. For example, look at a signature of Franklin D. Roosevelt at the beginning of his first term as president. Each name and initial is clearly discernible. Then, compare it with his signature in his third term, just eight years later. To those familiar with it, the strokes and lines are clearly the president's signature; but to others, they bear no relationship to his name or any other words. Keystroke patterns change similarly over time, particularly depending on typing frequency.

Systems that update the database automatically average the current input data into the database template after the identification transaction is complete. Some also delete an earlier data input, making that database a moving average. These gradual changes in input data may not affect user identification for many months or years. However, as the database file and the input data become further apart, increasingly frequent false rejections will cause enough inconvenience that re-enrollment is dictated, which is another inconvenience.

Different Types of Biometric Systems and Their Characteristics

This section describes the different types of biometric systems: fingerprint systems, hand geometry systems, voice pattern systems, retina pattern systems, iris pattern systems, and signature dynamics systems. For each system, the following characteristics are described: the enrollment procedure and time, the template or file size, the user action required, the system response time, any anti-counterfeit method, accuracy, field history, problems experienced, and unique system aspects.

Fingerprint Systems

The information in this section is a compilation of information about several biometric identifying verification systems whose technology is based on the fingerprint.

Data Acquisition

Fingerprint data is acquired when subjects firmly press their fingers against a glass or polycarbonate plate. The fingerprint image is not stored. Information on the relative location of the ridges, whorls, lines, bifurcations, and intersections is stored as an enrolled user database file and later compared with user input data.

Enrollment Procedure and Time

As instructed, subject enters a one- to nine-digit PIN on the keypad. As cued, the finger is placed on the reader plate and then removed. A digitized code is created. As cued, the finger is placed and removed four more times for calibration. The total enrollment time required is less than two minutes.

Template or File Size

Fingerprint user files are generally between 500 and 1500 bytes.

User Actions Required

Nearly all fingerprint-based biometrics are verification systems. The user states identification by entering a PIN through a keypad or by using a card reader, and then places a finger on the reader plate.

System Response Time

Visual and audible annunciation of the confirmed and not confirmed decision occurs in five to seven seconds.

Accuracy

Some fingerprint systems can be adjusted to achieve a false accept rate of 0.0 percent. Sandia National Laboratories tests of a top-rated fingerprint system in 1991 and 1993 produced a three-try false reject rate of 9.4 percent and a crossover error rate of 5 percent.

Field History

Thousands of units have been fielded for access control and identity verification for disbursement of government benefits, for example.

Problems Experienced

System operators with large user populations are often required to clean sensor plates frequently to remove built-up skin oil and dirt that adversely affect system accuracy.

Unique System Aspects

To avoid the dirt build-up problem, a newly developed fingerprint system acquires the fingerprint image with ultrasound. Claims are made that this system can acquire the fingerprint of a surgeon wearing latex gloves. A number of companies are producing fingerprint-based biometric identification systems.

Hand Geometry System

Hand geometry data, the three-dimensional record of the length, width, and height of the hand and fingers, is acquired by simultaneous vertical and horizontal camera images.

Enrollment Procedure and Time

The subject is directed to place the hand flat on a grid platen, positioned against pegs between the fingers. Four finger-position lights ensure proper hand location. A digital camera records a single top and side view from above, using a 45-degree mirror for the side view. The subject is directed to withdraw and then reposition the hand twice more. The readings are averaged into a single code and given a PIN. Total enrollment time is less than two minutes.

Template or File Size

The hand geometry user file size is nine bytes.

User Actions Required

The hand geometry system operates only as an identification verifier. The user provides identification by entering a PIN on a keypad or by using a cardreader. When the "place hand" message appears on the unit display, the user places his or her hand flat on the platen against the pegs. When all four lights confirm correct hand position, the data is acquired and a "remove hand" message appears.

System Response Time

Visual and audible annunciation of the confirm or not confirm decision occurs in three to five seconds.

Anticounterfeit Method

The manufacturer states that "the system checks to ensure that a live hand is used."

Accuracy

Sandia National Laboratories tests have produced a one-try false accept rate less than 0.1 percent, a three-try false reject rate less than 0.1 percent, and crossover error rates of 0.2 and 2.2 percent (i.e., two tests).

Field History

Thousands of units have been fielded for access control, college cafeterias and dormitories, and government facilities. Hand geometry was the original biometric system of choice of the Department of Energy and the Immigration and Naturalization Service. It was also used to protect the Athlete's Village at the 1996 Olympics in Atlanta.

Problems Experienced

Some of the field applications did not perform up to the accuracy results of the initial Sandia test. There have been indications that verification accuracy achieved when user databases are in the hundreds deteriorates when the database grows into the thousands.

Unique System Aspects

The hand geometry user file code of nine bytes is, by far, the smallest of any current biometric system. Hand geometry identification systems are manufactured by Recognition Systems, Inc. A variation, a two-finger geometry identification system, is manufactured by BioMet Partners.

Voice Pattern Systems

Up to seven parameters of nasal tones, larynx and throat vibrations, and air pressure from the voice are captured by audio and other sensors.

Enrollment Procedure and Time

Most voice systems use equipment similar to a standard telephone. As directed, the subject picks up the handset and enters a PIN on the telephone keypad. When cued through the handset, the subject speaks his or her access phrase, which may be his or her PIN and name or some other four- to six-word phrase. The cue and the access phrase are repeated up to four times. Total enrollment time required is less than two minutes.

Template or File Size

Voice user files vary from 1000 to 10,000 bytes, depending on the system manufacturer.

User Actions Required

Currently, voice systems operate only as identification verifiers. The user provides identification by entering the PIN on the telephone-type keypad. As cued through the handset (i.e., recorded voice stating "please say your access phrase"), the user speaks into the handset sensors.

System Response Time

Audible response (i.e., "accepted, please enter" or "not authorized") is provided through the handset. Some systems include visual annunciation (e.g., red and green lights or LEDs). Total transaction time requires up to 10 to 14 seconds.

Anti-counterfeit Method

Various methods are used, including measuring increased air pressure when "p" or "t" sounds are spoken. Some sophisticated systems require the user to speak different words from a list of ten or more enrolled words in a different order each time the system is used.

Accuracy

Sandia National Laboratories has reported crossover errors greater 10 percent for two systems they have tested. Other voice tests are being planned.

Field History

More than 100 systems have been installed, with over 1000 door access units, at colleges, hospitals, laboratories, and offices.

Problems Experienced

Background noise can affect the accuracy of voice systems. Access systems are located at entrances, hallways, and doorways, which tend to be busy, high-traffic, and high-noise-level sites.

Unique System Aspects

Some voice systems can also be used as an intercom or to leave messages for other system users. There are several companies producing voice-based biometric identification systems.

Retina Pattern System

The system records elements of the blood-vessel pattern of the retina on the inside rear portion of the eyeball using a camera to acquire the image.

Enrollment Procedure and Time

The subject is directed to position his or her eye an inch or two from the system aperture, keeping a pulsing green dot inside the unit centered in the aperture, and remain still. An ultra-low-intensity invisible light enables reading 320 points on a 450-degree circle on the retina. A PIN is entered on a unit keypad. Total enrollment time required is less than two minutes.

Template or File Size

The retina pattern digitized waveform is stored as a 96-byte template.

User Actions Required

If verifying, the user enters the PIN on the keypad. The system automatically acquires data when an eye is positioned in front of the aperture and centered on the pulsing green dot. Acceptance or nonacceptance is indicated in the LCD display.

System Response Time

Verification system decision time is about 1.5 seconds. Recognition decision time is less than five seconds with a 1,500-file data base. Average throughput time is four to seven seconds.

Anticounterfeit Method.

The system "requires a live, focusing eye to acquire pattern data," according to the manufacturer.

Accuracy

Sandia National Laboratories' test of the previous retina model produced no false accepts and a crossover error rate of 1.5 percent. The new model, System 2001, is expected to perform similarly.

Field History

Hundreds of the original binocular-type units were fielded before those models were discontinued. They were used for access control and identification in colleges, laboratories, government facilities, and jails. The new model, System 2001, is now on sale.

Problems Experienced

Because persons perspiring or having watery eyes could leave moisture on the eyecups of the previous models, some users were concerned about acquiring a disease through the transfer of body fluids. Because the previous models used a red light beam to acquire pattern data, some users were concerned about possible eye damage from the "laser." No allegations were made that any user actually became injured or diseased through the use of these systems. Because some physical conditions such as diabetes and heart attacks can cause changes in the retinal pattern, which can be detected by this system, some users were concerned that management would gain unauthorized medical information that could be used to their detriment. No cases of detrimental employee personnel actions resulting from retina system information have been reported.

Unique System Aspects

Some potential system users remain concerned about potential eye damage from using the new System 2001. They state that, even if they cannot see it, the system projects a beam inside the eye to read the retina pattern. Patents for retina-based identification are owned by EyeDentify Inc.

Iris Pattern System

The iris (i.e., the colored portion of the eye surrounding the pupil) has rich and unique patterns of striations, pits, freckles, rifts, fibers, filaments, rings, coronas, furrows, and vasculature. The images are acquired by a standard 1/3-inch CCD video camera capturing 30 images per second, similar to a camcorder.

Enrollment Procedure and Time

The subject looks at a mirror-like LCD feedback image of his or her eye, centering and focusing the image as directed. The system creates zones of analysis on the iris image, locates the features within the zones, and creates an IrisCode. The system processes three images, selects the most representative, and stores it upon approval of the operator. A PIN is added to the administrative (i.e., name, address) data file. Total enrollment time required is less than two minutes.

Template or File Size

The IrisCode occupies 256 bytes.

User Actions Required

The IriScan system can operate as a verifier, but is normally used in full identification mode because it performs this function faster than most systems verify. The user pushes the start button, tilts the optical unit if necessary to adjust for height, and looks at the LCD feedback image of his or her eye, centering and focusing the image. If the system is used as a verifier, a keypad or cardreader is interconnected.

System Response Time

Visual and audible annunciation of the identified or not identified decision occurs in one to two seconds, depending on the size of the database. Total throughput time (i.e., start button to annunciation) is 2.5 to 4 seconds with experienced users.

Anti-counterfeit Method

The system ensures that data input is from a live person by using naturally occurring physical factors of the eye.

Accuracy

Sandia National Laboratories' test of a preproduction model had no false accepts, low false rejects, and the system "performed extremely well." Sandia has a production system currently in testing. British Telecommunications recently tested the system in various modes and will publish a report in its engineering journal. They report 100 percent correct performance on over 250,000 IrisCode comparisons. "Iris recognition is a reliable and robust biometric. Every eye presented was enrolled. There were no false accepts, and every enrolled eye was successfully recognized." Other tests have reported a crossover error rate of less than 0.5 percent.

Field History

Units have been fielded for access control and personnel identification at military and government organizations, banks, telecommunications firms, prisons and jails, educational institutions, manufacturing companies, and security companies.

Problems Experienced

Because this is a camera-based system, the optical unit must be positioned such that the sun does not shine directly into the aperture.

Unique System Aspects

The iris of the eye is a stable organ that remains virtually unchanged from one year of age throughout life. Therefore, once enrolled, a person will always be recognized, absent certain eye injuries or diseases. IriScan Inc. has the patents worldwide on iris recognition technology.

Signature Dynamics Systems

The signature penstroke speed, direction, and pressure are recorded by small sensors in the pen, stylus, or writing tablet.

Enrollment Procedure and Time

As directed, the subject signs a normal signature by using the pen, stylus, or sensitive tablet provided. Five signatures are required. Some systems record three sets of coordinates versus time patterns as the template.

Templates are encrypted to preclude signature reproduction. A PIN is added using a keypad. Total enrollment time required is less than two minutes.

Template or File Size

Enrollment signature input is averaged into a 1000- to 1500-byte template.

User Actions Required

The user provides identification through PIN entry on a keypad or cardreader. The signature is then written using the instrument or tablet provided. Some systems permit the use of a stylus without paper if a copy of the signature is not required for a record.

System Response Time

Visual and audible annunciation of the verified or not verified decision is annunciated after about one second. The total throughput time is in the five to ten-second range, depending on the time required to write the signature.

Anticounterfeit Method

This feature is not applicable for signature dynamics systems.

Accuracy

Data collection is underway at pilot projects and beta test sites. Current signature dynamics biometric systems have not yet been tested by an independent agency.

Field History

Approximately 100 units are being used in about a dozen systems operated by organizations in the medical, pharmaceutical, banking, manufacturing, and government fields.

Problems Experienced

Signature dynamics systems, which previously performed well during laboratory and controlled tests, did not stand up to rigorous operational field use. Initially acceptable accuracy and reliability rates began to deteriorate after months of system field use. Although definitive failure information is not available, it is believed that the tiny, super-accurate sensors necessary to measure the minute changes in pen speed, pressure, and direction did not withstand the rough handling of the public. It is too early to tell whether the current generation of signature systems has overcome these shortcomings.

Unique System Aspects

Among the various biometric identification systems, bankers and lawyers advocate signature dynamics because legal documents and financial drafts historically have been validated by signature. Signature dynamics identification systems are not seen as candidates for access control and other security applications. There are several companies producing signature dynamics systems.

Information Security Applications

The use of biometric identification systems in support of information security applications falls into two basic categories: controlling access to hard-copy documents and to rooms where protected information is discussed, and controlling computer use and access to electronic data.

Access Control

Controlling access to hard-copy documents and to rooms where protected information is discussed can be accomplished using the systems and technologies previously discussed. This applies also to electronic data tape and disk repositories.

Computer and Electronic Data Protection

Controlling access to computers, the data they access and use, and the functions they can perform is becoming more vitally important with each passing day. Because of the ease of electronic access to immense amounts of

information and funds, losses in these areas have rapidly surpassed losses resulting from physical theft and fraud. Positive identification of the computer operators who are accessing vital programs and data files and performing vital functions is becoming imperative as it is the only way to eliminate these losses.

The use of passwords and PINs to control computer boot-up and program and data file call-up is better than no control at all, but is subject to all the shortcomings previously discussed. Simple, easy-to-remember codes are easy for the "bad guys" to figure out. Random or obtuse codes are difficult to remember and nearly always get written down in some convenient and vulnerable place. In addition, and just as important, is that these controls are only operative at the beginning of the operation or during access to the program or files.

What is needed is a biometric system capable of providing continuing, transparent, and positive identification of the person sitting at the computer keyboard. This system would interrupt the computer boot-up until the operator is positively identified as a person authorized to use that computer or terminal. This system would also prevent the use of controlled programs or data files until the operator is positively identified as a person authorized for such access. This system would also provide continuing, periodic (e.g., every 30 seconds) positive identification of the operator as long as these controlled programs or files were in use. If this system did not verify the presence of the authorized operator during a periodic check, the screen could be cleared of data. If this system verified the presence of an unauthorized or unidentified operator, the file and program could be closed.

Obviously, the viability of such a system depends on software with effective firewalls and programmer access controls to prevent tampering, insertion of unauthorized identification files, or bypasses. However, such software already exists. Moreover, a biometric identification system replacing the log-on password already exists. Not yet available is a viable, independently tested, continuing, and transparent operator identification system.

System Currently Available

Identix' TouchSafe™ provides verification of enrolled persons who log on or off the computer. It comes with an IBM-compatible plug-in electronics card and a $5.4 \times 2.5 \times 3.6$-inch fingerprint reader unit with cable. This unit can be expected to be even more accurate than the normal fingerprint access control systems previously described because of a more controlled operating environment and limited user list. However, it does not provide for continuing or transparent identification. Every time that identification is required, the operator must stop activity and place a finger on the reader.

Systems Being Developed

Only a camera-based system can provide the necessary continuing and transparent identification. With a small video camera mounted on a top corner of the computer monitor, the system could be programmed to check operator identity every 30 or 60 seconds. Because the operator can be expected to look at the screen frequently, a face or iris identification system would be effective without ever interrupting the operator's work. Such a system could be set to have a 15-second observation window to acquire an acceptable image and identify the operator. If the operator did not look toward the screen or was not present during the 15-second window, the screen would be cleared with a screen saver. The system would remain in the observation mode so that when the operator returned to the keyboard or looked at the screen and was identified, the screen would be restored. If the operator at the keyboard was not authorized or was unidentified, the program and files would be saved and closed.

The first development system that seems to have potential for providing these capabilities is a face recognition system from Miros Inc. Miros is working on a line of products called TrueFace. At this time, no independent test data are available concerning the performance and accuracy of Miros' developing systems. Face recognition research has been under way for many years, but no successful systems have yet reached the marketplace. Further, the biometric identification industry has a history of promising developments that have failed to deliver acceptable results in field use. Conclusions regarding Miros' developments must wait for performance and accuracy tests by a recognized independent organization.

IriScan Inc. is in the initial stages of developing an iris recognition system capable of providing the desired computer or information access control capabilities. IriScan's demonstrated accuracy gives this development the potential to be the most accurate information user identification system.

Summary

The era of fast, accurate, cost-effective biometric identification systems has arrived. Societal activities increasingly threaten individuals' and organizations' assets, information, and, sometimes, even their existence. Instant, positive personal identification is a critically important step in controlling access to and protecting society's resources. Effective tools are now available.

There are more than a dozen companies manufacturing and selling significant numbers of biometric identification systems today. Even more organizations are conducting biometric research and development and hoping to break into the market or are already selling small numbers of units. Not all biometric systems and technologies are equally effective in general, nor specifically in meeting all application requirements. Security managers are advised to be cautious and thorough in researching candidate biometric systems before making a selection. Independent test results and the reports of current users with similar applications are recommended. On-site tests are desirable. Those who are diligent and meticulous in their selection and installation of a biometric identification system will realize major increases in asset protection levels.

7

Single Sign-On for the Enterprise

Ross A. Leo, CISSP

Corporations everywhere have made the functional shift from the mainframe-centered data processing environment to the client/server configuration. With this conversion have come new economies, a greater variety of operational options, and a new set of challenges. In the mainframe-centric installation, systems management was often the administrative twin of the computing complex itself: the components of the system were confined to one area, as were those who performed the administration of the system. In the distributed client/server arrangement, those who manage the systems are again arranged in a similar fashion. This distributed infrastructure has complicated operations, even to the extent of making the simple act of logging in more difficult.

Users need access to many different systems and applications to accomplish their work. Getting them set up to do this simply and easily is frequently time-consuming, requiring coordination between several individuals across multiple systems. In the mainframe environment, switching between these systems and applications meant returning to a main menu and making a new selection. In the client/server world, this can mean logging in to an entirely different system. New loginid, new password, and both very likely different than the ones used for the previous system — the user is inundated with these, and the problem of keeping them un-confused to prevent failed log-in attempts. It was because of this and related problems that the concept of the **Single Sign-On**, or SSO, was born.

Evolution

Given the diversity of computing platforms, operating systems, and access control software (and the many loginids and passwords that go with them), having the capability to log on to multiple systems once and simultaneously through a single transaction would seem an answer to a prayer. Such a prayer is one offered by users and access control administrators everywhere. When the concept arose of a method to accomplish this, it became clear that integrating it with the different forms of system access control would pose a daunting challenge with many hurdles.

In the days when applications software ran on a single platform, such as the early days of the mainframe, there was by default only a single login that users had to perform. Whether the application was batch oriented or interactive, the user had only a single loginid and password combination to remember. When the time came for changing passwords, the user could often make up his own. The worst thing to face was the random password generator software implemented by some companies that served up number/letter combinations. Even then, there was only one of them.

The next step was the addition of multiple computers of the same type on the same network. While these machines did not always communicate with each other, the user had to access more than one of them to fulfill all data requirements. Multiple systems, even of the same type, often had different rules of use. Different groups within the data processing department often controlled these disparate systems and sometimes completely separate organizations with the same company. Of course, the user had to have a different loginid and password for each one, although each system was reachable from the same terminal.

Then, the so-called "departmental computer" appeared. These smaller, less powerful processors served specific groups in the company to run unique applications specific to that department. Examples include materials management, accounting and finance applications, centralized word-processing, and shop-floor applications. Given the limited needs of these areas, and the fact that they frequently communicated electronically internal to themselves, tying these systems together on the same network was unnecessary. This state of affairs did not last long.

It soon became obvious that tying these systems together, and allowing them to communicate with each other over the network would speed up the information flow from one area to another. Instead of having to wait until the last week of the month to get a report through internal mail, purchasing records could be reconciled weekly with inventory records for materials received the same week from batched reports sent to purchasing. This next phase in the process of information flow did not last long either.

As systems became less and less batch oriented and more interactive, and business pressures to record the movement of goods, services, and money mounted, more rapid access was demanded. Users in one area needed direct access to information in another. There was just one problem with this scenario — and it was not a small one.

Computers have nearly always come in predominantly two different flavors: the general-purpose machines and specific-use machines. Initially called "business processing systems" and "scientific and engineering systems," these computers began the divergence from a single protocol and single operating system that continues today. For a single user to have access to both often required two separate networks because each ran on a different protocol. This of course meant two different terminals on that user's desk. That all the systems came from the same manufacturer was immaterial: the systems could not be combined on the same wire or workstation.

The next stage in the evolution was to hook in various types of adapters, multiple screen "windowed" displays, protocol converters, etc. These devices sometimes eliminated the second terminal. Then came the now-ubiquitous personal computer, or "PC" as it was first called when it was introduced by IBM on August 12, 1981. Within a few short years, adapters appeared that permitted this indispensable device to connect and display information from nearly every type of larger host computer then in service. Another godsend had hit the end user!

This evolution has continued to the present day. Most proprietary protocols have gone the way of the woolly Mammoth, and have resolved down to a precious few, nearly all of them speaking TCP/IP in some form. This convergence is extremely significant: the basic method of linking all these different computing platforms together with a common protocol on the same wire exists.

The advent of Microsoft Windows pushed this convergence one very large step further. Just as protocols had come together, so too the capability of displaying sessions with the different computers was materializing. With refinement, the graphical user interface ("GUI" — same as gooey) enabled simultaneous displays from different hosts. Once virtual memory became a reality on the PC, this pushed this envelope further still by permitting simultaneous active displays and processing.

Users were getting capabilities they had wanted and needed for years. Now impossible tasks with impossible deadlines were rendered normal, even routine. But despite all the progress that had been made, the real issue had yet to be addressed. True to form, users were grateful for all the new toys and the ease of use they promised … until they woke up and found that none of these innovations fixed the thing they had complained most and loudest about: multiple loginids and passwords.

So what is single sign-on?

What Single Sign-On Is: The Beginning

Beginning nearly 50 years ago, system designers realized that a method of tracking interaction with computer systems was needed, and so a form of identification — the loginid — was conceived. Almost simultaneously with this came the password — that sometimes arcane companion to the loginid that authenticates, or confirms the identity of, the user. And for most of the past five decades, a single loginid and its associated password was sufficient to assist the user in gaining access to virtually all the computing power then available, and to all the applications and systems that user was likely to use. Yes, those were the days… simple, straightforward, and easy to administer. And now they are all but gone, much like the club moss, the vacuum tube, and MS/DOS (perhaps).

Today's environment is more distributed in terms of both geography and platform. Although some will dispute, the attributes differentiating one operating system from another are being obscured by both network access and graphical user interfaces (the ubiquitous GUI). Because not every developer has chosen to offer his or her particular application on every computing platform (and networks have evolved to the point of being seemingly oblivious to this diversity), users now have access to a broader range of tools spread across more platforms, more transparently than at any time in the past. And yet all is not paradise.

Along with this wealth of power and utility comes the same requirement as before: to identify and authenticate the user. But now this must be done across all these various systems and platforms, and (no surprise) they all have differing mechanisms to accomplish this. The result is that users now have multiple loginids, each with its own unique password, quite probably governed by its equally unique set of rules. The CISSP knows that users complain bitterly about this situation, and will often attempt to circumvent it by whatever means necessary. To avoid this, the CISSP had to find a solution. To facilitate this, and take advantage of a marketing opportunity, software vendors saw a vital need, and thus the single sign-on (SSO) was conceived to address these issues.

Exhibit 7.1 shows where SSO was featured in the overall security program when it first appeared. As an access control method, SSO addressed important needs across multiple platforms (user identification and authentication). It was frequently regarded as a "user convenience" that was difficult and costly to implement, and of questionable value in terms of its contribution to the overall information protection and control structure.

The Essential Problem

In simplest terms, too many loginids and passwords, and a host of other user access administration issues. With complex management structures requiring a geographically dispersed matrix approach to oversee employee work, distributed and often very different systems are necessary to meet operational objectives and reporting requirements.

In the days of largely mainframe-oriented systems, a problem of this sort was virtually nonexistent. Standards were made and enforcement was not complex. In these days, such conditions carry the same mandate for the establishment and enforcement of various system standards. Now, however, such conditions, and the systems arising in them, are of themselves not naturally conducive to this.

As mentioned above, such systems have different built-in systems for tracking user activity. The basic concepts are similar: audit trail, access control rule sets, Access Control Lists (ACLs), parameters governing system privilege levels, etc. In the end, it becomes apparent that one set of rules and standards, while sound in theory, may be exceedingly difficult to implement across all platforms without creating unmanageable complexity. It is however the "Holy Grail" that enterprise-level user administrators seek.

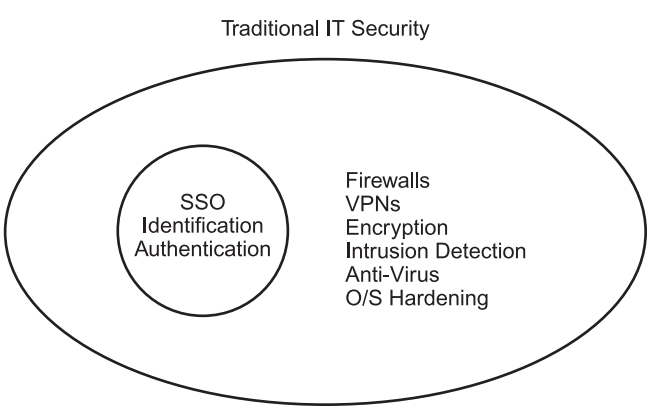


EXHIBIT 7.1 Single sign-on: in the beginning.

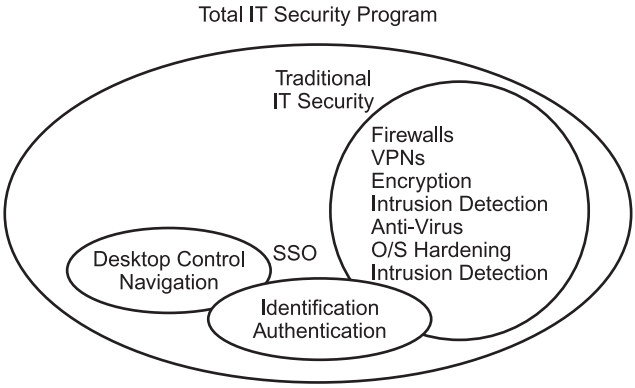


EXHIBIT 7.2 The evolution of SSO.

Despite the seeming simplicity of this problem, it represents only the tip of a range of problems associated with user administration. Such problems exist wherever the controlling access of users to resources is enforced: local in-house, remote WAN nodes, remote dial-in, and Web-based access.

As compared with Exhibit 7.1, Exhibit 7.2 illustrates how SSO has evolved into a broader scope product with greater functionality. Once considered merely a "user convenience," SSO has been more tightly integrated with other, more traditional security products and capabilities. This evolution has improved SSO's image measurably, but has not simplified its implementation.

In addition to the problem mentioned above, the need for this type of capability manifests itself in a variety of ways, some of which include:

1. As the number of entry points increases (Internet included), there is a need to implement improved and auditable security controls.
2. The management of large numbers of workstations is dictating that some control be placed over how they are used to avoid viruses, limit user-introduced problems, minimize help desk resources, etc.
3. As workstations have become electronic assistants, there has likewise arisen a need for end users to be able to use various workstations along their work path to reach their electronic desktop.
4. The proliferation of applications has made getting to all the information that is required too difficult, too cumbersome, or too time-consuming, even after passwords are automated.
5. The administration of security needs to move from an application focus to a global focus to improve compliance with industry guidelines and to increase efficiency.

Mechanisms

The mechanisms used to implement SSO have varied over time. One method uses the Kerberos product to authenticate users and resources to each other through a "ticketing" system, tickets being the vehicle through which authorization to systems and resources is granted. Another method has been shells and scripting: primary authentication to the shell, which then initiated various platform-specific scripts to activate account and resource access on the target platforms.

For those organizations not wanting to expend the time and effort involved with a Kerberos implementation, the final solution was likely to be a variation of the shell-and-script approach. This had several drawbacks. It did not remove the need to set up user accounts individually on each platform. It also did not provide password synchronization or other management features. Shell-and-scripting was a half-step at best, and although it simplified user login, that was about the extent of the automation it facilitated. That was "then."

Today, different configuration approaches and options are available when implementing an SSO platform, and the drawbacks of the previous attempts have largely been well-addressed. Regardless, from the security engineering perspective, the design and objectives (i.e., the problem one is trying to solve) for the implementation plan must be evaluated in a risk analysis, and then mitigated as warranted. In the case of SSO, the operational concerns should also be evaluated, as discussed below.

One form of implementation allows one login session, which concludes with the user being actively connected to the full range of their authorized resources until logout. This type of configuration allows for reauthentication based on time (every … minutes or hours) or can be event driven (i.e., system boundary crossing).

One concern with this configuration is resource utilization. This is because a lot of network traffic is generated during login, directory/ACL accesses are performed, and several application/system sessions are established. This level of activity will degrade overall system performance substantially, especially if several users engage their login attempts simultaneously. Prevention of session loss (due to inactivity timeouts) would likely require an occasional "ping" to prevent this, if the feature itself cannot be deactivated. This too consumes resources with additional network traffic.

The other major concern with this approach would be that "open sessions" would exist, regardless of whether the user is active in a given application or not. This might make possible "session stealing" should the data stream be invaded, penetrated, or rerouted.

Another potential configuration would perform the initial identification/authentication to the network service, but would not initialize access to a specific system or application until the user explicitly requests it (i.e., double-click the related desktop icon). This would reduce the network traffic level, and would invoke new sessions only when requested. The periodic reauthentication would still apply.

What Single Sign-On Provides

SSO products have moved beyond simple end-user authentication and password management to more complex issues that include addressing the centralized administration of endpoint systems, the administration of end users through a role-based view that allows large populations of end users to be affected by a single system administration change (e.g., adding a new application to all office workers), and the monitoring of end users' usage of sensitive applications.

The next section describes many of the capabilities and features that an ideal single sign-on product might offer. Some of the items that mention cost refer expressly to the point being made, and not to the software performing the function. The life-cycle cost of a product such as that discussed here can and does vary widely from one installation to the next. The extent of such variation is based on many factors, and is well beyond the scope of this discussion.

A major concern with applying the SSO product to achieve the potential economies is raised when consideration is given to the cost of the product, and comparing it to the cost of how things were done pre-SSO, and contrasting this with the cost of how things will be done post-SSO, the cost of putting SSO in, and all other dollars expended in the course of project completion.

By comparing the before-and-after expenditures, the ROI (return on investment) for installing the SSO can be calculated and used as part of the justification for the project. It is recommended that this be done using equivalent formulas, constraints, and investment/ROI objectives the enterprise applies when considering any project. When the analysis and results are presented (assuming they favor this undertaking), the audience will have better insight into the soundness of the investment in terms of real costs and real value contribution. Such insight fosters endorsement, and favors greater acceptance of what will likely be a substantial cost and lengthy implementation timeline.

Regardless, it is reasonably accurate to say that this technology is neither cheap to acquire nor to maintain. In addition, as with any problem-solution set, the question must be asked, "Is this problem worth the price of the solution?" The next section discusses some of the features to assist in making such a decision.

Internal Capability Foundation

Having GUI-based central administration offers the potential for simplified user management, and thus possibly substantial cost-savings in reduced training, reduced administrative effort, and lower life-cycle cost for user management. This would have beneath it a logging capability that, based on some DBMS engine and a set of report generation tools, would enhance and streamline the data reduction process for activity reporting and forensic analysis derived through the SSO product.

The basic support structure must include direct (standard customary login) and Web-based access. This would be standard, especially now that the Internet has become so prolific and also since an increasing number of applications are using some form of Web-enabled/aware interface. This means that the SSO implementation

would necessarily limit the scope or depth of the login process to make remote access practical, whether direct dial-up or via the Web.

One aspect of concern is the intrusiveness of the implementation. Intrusiveness is the extent to which the operating environment must be modified to accommodate the functionality of the product. Another is the retrofitting of legacy systems and applications. Installation of the SSO product on the various platforms in the enterprise would generally be done through APIs to minimize the level of custom code.

Not surprisingly, most SSO solutions vendors developed their product with the retrofit of legacy systems in mind. For example, the Platinum Technologies (now CA) product AutoSecure SSO supported RACF, ACF2, and TopSecret — all of which are access control applications born and bred in the legacy systems world. It also supports Windows NT, Novell, and TCP/IP network-supported systems. Thus, it covers the range from present day to legacy.

General Characteristics

The right SSO product should provide all the required features and sustain itself in an enterprise production environment. Products that operate in an open systems distributed computing environment, complete with parallel network servers, are better positioned to address enterprise needs than more narrow NOS-based SSO products.

It is obvious then that SSO products must be able to support a fairly broad array of systems, devices, and interfaces if the promise of this technology is to be realized. Given that, it is clear some environments will require greater modification than others; that is, the SSO configuration is more complex and modifies the operating environment to a greater extent. Information derived through the following questions will assist in pre-implementation analysis:

1. Is the SSO nonintrusive; that is, can it manage access to all applications, without a need to change the applications in any way?
2. Does the SSO product dictate a single common logon and password across all applications?
3. What workstations are supported by the SSO product?
4. On what operating systems can SSO network servers operate?
5. What physical identification technologies are supported (e.g., Secure-ID card)?
6. Are dial-up end users supported?
7. Is Internet access supported? If so, are authentication and encryption enforced?
8. Can the SSO desktop optionally replace the standard desktop to more closely control the usage of particular workstations (e.g., in the production area)?
9. Can passwords be automatically captured the first time an end user uses an endpoint application under the SSO product's control?
10. Can the look of the SSO desktop be replaced with a custom site-specific desktop look?
11. How will the SSO work with the PKI framework already installed?

End-User Management Facilities

These features and options include the normal suite of functions for account creation, password management, etc. The performance of end-user identification and authentication is obvious. Password management includes all the normal features: password aging, histories, and syntax rules. To complete the picture, support for the wide variety of token-type devices (Secure-ID cards), biometric devices, and the like should be considered, especially if remote end users are going to be using the SSO product. At the very least, optional modules providing this support should exist and be available.

Some additional attributes that should be available are:

- *Role-based privileges.* This functionality makes it possible to administer a limited number of roles that are in turn shared by a large population of end users. This would not necessarily have any effect on individual users working outside the authority scope of that role.
- *Desktop control.* This allows the native desktop to be replaced by an SSO-managed desktop, thereby preventing end users from using the workstation in such a way as to create support problems (e.g., introducing unauthorized software). This capability is particularly important in areas where workstations are shared by end users (e.g., production floor).

- *Application authorization.* This ensures that any launched application is registered and cleared by the SSO product and records are kept of individual application usage.
- *Mobile user support.* This capability allows end users to reach their desktop, independent of their location or the workstation they are using. It should also include configuring the workstation to access the proper domain server and bringing the individual's preferences to the workstation before launching applications.

Application Management Facilities

Application management in the context of SSO refers to the treatment of an application in a manner similar to how it manages or treats users. As shown in Exhibit 7.2, the evolved state of SSO has moved beyond the simplistic identification/authentication of users, and now encompasses certain aspects of application management. This management capability relates to the appearance of user desktops and navigation through application menus and interfaces rather than with the maintenance and upgrading of application functionality.

Context management ensures that when multiple sessions that relate to a common subject are simultaneously active, each session is automatically updated when another related session changes position (e.g., in a healthcare setting, the lab and pharmacy sessions must be on the same patient if the clinician is to avoid mixing two patients' records when reaching a clinical decision).

Application monitoring is particularly useful when it is desirable to monitor the usage of particular rows of information in an application that is not programmed to provide that type of information (e.g., access to particular constituents' records in a government setting).

Application positioning is a feature that relates to personalized yet centrally controlled desktops. This allows configuration of an end-user start-up script to open an application (possibly chosen from a set of options) on initialization, and specify even what screen is loaded.

One other feature that binds applications together is application fusing. This allows applications to operate in unison such that the end user is only aware of a single session. The view to the end user can range from a simple automated switching between applications up to and including creating an entirely new view for the end user.

Endpoint Management Facilities

Endpoint administration is an essential component of an SSO product because, without it, administration is forced to input the same information twice; once in the SSO and once in the endpoint each time a change is made to the SSO database. Two methods of input into the endpoint should be supported: (1) API-based agents to update endpoint systems that support an API, and (2) session animation agents to update endpoint systems that do not support an API. Services provided by the SSO to accomplish this administrative goal should include:

- *Access control.* This is the vehicle used by end users to gain access to applications and, based on each application's capabilities, to define to the application the end user's privileges within it. Both API-based and session-based applications should be supported.
- *Audit services.* These should be made available through an API to endpoint applications that wish to publish information into the SSO product's logging system.
- *Session encryption.* This feature ensures information is protected from disclosure and tampering as it moves between applications and end users. This capability should be a requirement in situations where sensitive applications only offer cleartext facilities.

Mobile Users

The capability for end users to use any available workstation to reach information sources is mandatory in environments where end users are expected to function in a number of different locations. Such users would include traveling employees, healthcare providers (mobile nurses, physicians, and technicians), consultants, and sales staff. In the highly mobile workforce of today's world, it is unlikely that a product not offering this feature would be successful.

Another possible feature would facilitate workstation sharing; that is, the sharing of the device by multiple simultaneous users, each one with their own active session separate from all others. This capability would entail the use of a form of screen swapping so that loginids and passwords would not be shared. When the

first user finishes his session, rather than log out, he locks the session, a hot-key combination switches to the next open login screen, and the second user initiates his session, etc.

When investigating the potential needs in this regard, the questions to ask yourself and the vendors of such products should include:

1. Can a workstation in a common area be shared by many end users (e.g., production floor)?
2. If someone wants to use a workstation already in use by another end user, can the SSO product gracefully close the existing end user's applications (including closing open documents) and turn control over to the new end user?
3. Can end users adjust the organization of their desktop, and if so, does it travel with them, independent of the workstation they use?
4. Can individual applications preferences travel with the end user to other workstations (e.g., MS Word preferences)?
5. Can the set of available applications be configured to vary based on the entry point of the end user into the network?
6. If a Novell end user is logging in at a workstation that is assigned to a different Novell domain, how does the end user get back to his or her domain?
7. Given that Windows 95 and Windows NT rely on a locally stored password for authentication, what happens when the end user logs onto another workstation?
8. Is the date and time of the last successful sign-on shown at the time the end user signs on to highlight unauthorized sign-ons?
9. Is the name of the logged in end user prominently displayed to avoid inadvertent use of workstations by other end users?

Authentication

Authentication ensures that users are who are who they claim to be. It also ensures that all processes and transactions are initiated only by authorized end users. User authentication couples the loginid and the password, providing an identifier for the user, a mechanism for assigning access privileges, and an auditing "marker" for the system against which to track all activity, such as file accesses, process initiation, and other actions (e.g., attempted logons). Thus, through the process of authentication, one has the means to control and track the "who" and the "what."

The SSO products take this process and enable it to be used for additional services that enhance and extend the applications of the loginid/password combination. Some of these applications provide a convenience for the user that also improves security: the ability to lock the workstation just before stepping away briefly means the user is more likely to do it, rather than leave his workstation open for abuse by another. Some are extensions of audit tools: display of last login attempt, and log entry of all sign-ons. These features are certainly not unique to SSO, but they extend and enhance its functionality, and thus make it more user friendly.

As part of a Public Key Infrastructure (PKI) installation, the SSO should have the capability to support digital certificate authentication. Through a variety of methods (token, password input, biometrics possibly), the SSO supplies a digital certificate for the user that the system then uses as both an authenticator and an access privilege "license" in a fashion similar to the Kerberos ticket. The vital point here is not how this functionality is actually performed (that is another lengthy discussion), but that the SSO supports and integrates with a PKI, and that it uses widely recognized standards in doing so.

It should noted, however, that any SSO product that offers less than the standard suite of features obtainable through the more common access control programs should *not* be considered. Such a product may be offered as an alternative to the more richly featured SSO products on the premise that "simpler is better." Simpler is not better in this case because it means reduced effectiveness.

To know whether the candidates measure up, an inquiry should be made regarding these aspects:

1. Is authentication done at a network server or in the workstation?
2. Is authentication done with a proven and accepted standard (e.g., Kerberos)?
3. Are all sign-on attempts logged?
4. After a site-specified number of failed sign-on attempts, can all future sign-on attempts be unconditionally rejected?

5. Is an inactivity timer available to lock or close the desktop when there is a lack of activity for a period of time?
6. Can the desktop be easily locked or closed when someone leaves a workstation (e.g., depression of single key)?
7. Is the date and time of the last successful sign-on shown at the time the end user signs on to highlight unauthorized sign-ons?

Encryption

Encryption ensures that information that flows between the end users and the security server(s) and endpoint applications they access is not intercepted through spying, line-tapping, or some other method of eavesdropping. Many SSO products encrypt traffic between the end user and the security server but let cleartext pass between the end user and the endpoint applications, causing a potential security gap to exist. Some products by default encrypt all traffic between workstation and server, some do not, and still others provide this feature as an option that is selectable at installation.

Each installation is different in its environment and requirements. The same holds true when it comes to risks and vulnerabilities. Points to cover that address this include:

- Is all traffic between the workstation and the SSO server encrypted?
- Can the SSO product provide encryption all the way to the endpoint applications (e.g., computer room) without requiring changes to the endpoint applications?
- Is the data stream encrypted using an accepted and proven standard algorithm (e.g., DES, Triple DES, IDEA, AES, or other)?

Access Control

End users should only be presented with the applications they are authorized to access. Activities required to launch these applications should be carefully evaluated because many SSO products assume that only API-based endpoint applications can participate, or that the SSO is the owner of a single password that all endpoint applications must comply with. These activities include automatically inputting and updating application passwords when they expire.

Exhibit 7.3 shows how the SSO facilitates automatic login and acquisition of all resources to which a user is authorized. The user logs into the authentication server (centrally positioned on the network). This then validates the user and his access rights. The server then sends out the validated credentials and activates the required scripts to log the user in and attach his resources to the initiated session.

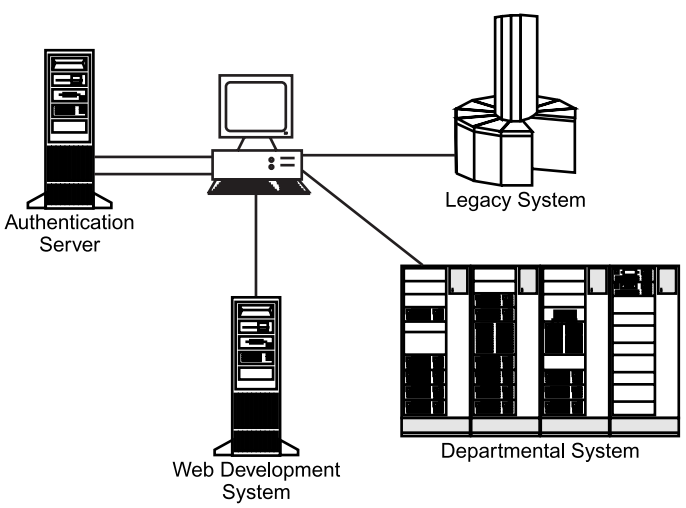


EXHIBIT 7.3 Automated login.

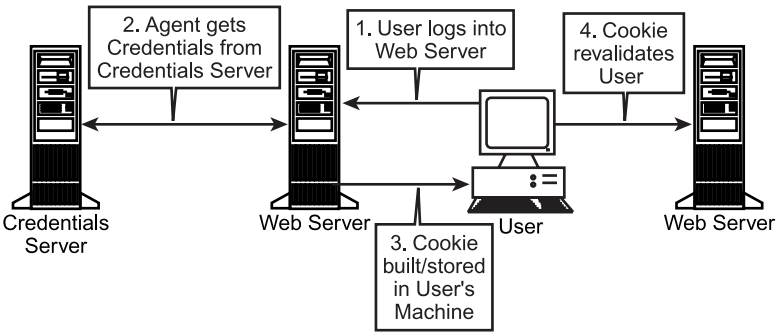


EXHIBIT 7.4 SSO: Web with cookies.

While it is certainly true that automatically generated passwords might make the user's life easier, current best practice is to allow users to create and use their own passwords. Along with this should be a rule set governing the syntax of those passwords; for example, no dictionary words, a combination of numbers and letters, a mixture of case among the letters, no repetition within a certain number of password generations, proscribed use of special characters (#, $, &, ?, %, etc.), and other rules. The SSO should support this function across all intended interfaces to systems and applications.

Exhibit 7.4 shows how the SSO facilitates login over the World Wide Web (WWW) by making use of cookies — small information packets shipped back and forth over the Web. The user logs into the initial Web server (1), which then activates an agent that retrieves the user's credentials from the credentials server (2). This server is similar in function to a name server or an LDAP server, except that this device provides authorization and access privileges information specifically. The cookie is then built and stored in the user's machine (3), and is used to revalidate the user each time a page transition is made.

This process is similar to verification of application-level privileges inside a DBMS. While moving within the database system, each time the user accesses a new region or transaction, access privileges must be reverified to ensure correct authorization. Page transitions on the Web equate to new regions or transactions within the DBMS.

In this area, the following points should be covered:

1. Can all applications, regardless of platform, be nonintrusively supported (i.e., without changing them, either extensively or at all)?
2. What types of adapters are available to mechanize the application launching process without having to adjust the individual applications? Are API-based, OLE-based, DDE-based, scripting-based, and session-simulation adapters available?
3. Are all application activations and deactivations logged?
4. When application passwords expire, does the SSO product automatically generate new expired one-time passwords or are users able to select and enter their own choices?
5. When an application is activated, can information be used to navigate to the proper position in the application (e.g., order entry application is positioned to the order entry screen)?
6. Can the application activation procedure be hidden from the end user, or does the end user have to see the mechanized process as it progresses?
7. Are inactivity timers available to terminate an application when there is a lack of activity for a period of time?

Application Control

Application control limits end users' use of applications in such a way that only particular screens within a given application are visible, only specific records can be requested, and particular uses of the applications can be recorded for audit purposes, transparently to the endpoint applications so no changes are needed to the applications involved.

As a way in which user navigation is controlled, this is another feature that can assist with enhancing the overall security posture of an installation. Again, this would be as an adjunct feature — not the key method. The determination of the usefulness of this capability can be made through the following questions.

1. Can applets be incorporated into the desktop's presentation space (e.g., list of major accounts)?
2. Can applet information (e.g., particular account) be used to navigate to the proper position within an application (e.g., list of orders outstanding for a particular customer)?
3. Can each application's view be adjusted to show only the information that is appropriate for a particular end user?
4. Can the SSO product log end users' activities inside applications (e.g., which accounts have been accessed)?
5. Can application screens be enhanced with new capabilities without having to change the applications themselves (e.g., additional validation of input as it is captured)?
6. Can the SSO product log attempt to reach areas of applications that go beyond permitted areas (e.g., confidential patient information)?
7. Can multiple applications be fused into a single end-user session to eliminate the need for end users to learn each application?
8. Can applications be automatically coordinated such that end-user movement in one application (e.g., billing) automatically repositions subordinate application sessions (e.g., current orders, accounts receivable)?

Administration

The centralized administration capabilities offered by the SSO are — if not the main attraction — the "Holy Grail" mentioned earlier. The management (creation, modification, deletion) of user accounts and resource profiles through an SSO product can streamline and simplify this function within an organization or enterprise. The power of the administration tools is key because the cost of administering a large population of end users can easily overshadow the cost of the SSO product itself.

The product analysis should take the following attributes into consideration:

1. Does the SSO product allow for the central administration of all endpoint systems? (That is, changes to the central administration database are automatically reflected in endpoint systems.)
2. Is administration done at an "end-user" or a "role within the enterprise" level? (This is a critical element because an end-user focus can result in disproportional administration effort.)
3. Does each workstation have to be individually installed? If so, what is the estimated time required?
4. Can end users' roles in the organization be easily changed (to deal with people that perform mixed roles)?
5. Is the desktop automatically adjusted if the end user's roles are changed, or does the desktop view have to be adjusted manually?
6. Can an administrator see a list of active end users by application?
7. Can an administrator access all granted passwords to specific endpoint applications?
8. Does the product gracefully deal with network server failures?

Services for Desktop-Aware Applications

In cases where it is possible to modify existing endpoint applications, the ability for them to cooperatively share responsibilities with the desktop is very attractive. What is required is a published desktop API and associated services.

The circumstance can and does arise where the end user wants to customize a standard product in the enterprise suite for his own use in a way that affects only him and does not change the basic application itself. Such customization may include display formats, scripts, and processes relating to specific tasks the individual user wants or needs to use in conjunction with the server-supplied application. Through the supplied API, the user can make the custom changes necessary without impediment, and this allows other users to proceed without affecting them or their workstations.

In such cases, the user wanting the changes may require specific access and other controls to lock out other users. An example might be one where the user requiring the changes works on sensitive or restricted information, and others in the same area do not, and are not permitted access to such. This then may necessitate

the use of access controls embedded in the scripts used to change his desktop to meet his additional security needs.

That being the case, the API should provide the capability to access the SSO, and perform the access/privilege checking, without the user (the one making the localized changes) having any direct access to the SSO access/privilege database. This should likewise be true to facilitate the logging of access attempts, transactions, and data access authorizations to track the use of the local workstation. To determine the existence of this facility in the SSO, questions should be asked regarding such services, APIs, and related capabilities, such as:

1. Can desktop-aware applications interrogate end-user permissions managed by the SSO product?
2. Can desktop-aware applications make use the SSO product's logging facilities for their own use?
3. Do API services exist that enable desktop customization?
4. Do these APIs facilitate this without compromising overall system integrity by providing "back-door" access to the resident security information database?

Reliability and Performance

Given that an SSO product is, by necessity, positioned between the end users and the applications they need access to get their jobs done, it has a very high visibility within the enterprise and any unexpected reliability or performance problems can have serious consequences. This issue points directly back at the original business case made to justify the product.

Concerns with regard to reliability and performance generally focus on the additional layering of one software upon another ("yet another layer"), the interfaces between the SSO and other access control programs it touches, the complexity of these interactions, etc. One aspect of concern is the increased latency introduced by this new layer. The time from power-on to login screen has steadily increased over the years, and the addition of the SSO may increase it yet again. This can exacerbate user frustration.

The question of reliability arises when considering the interaction between the SSO and the other security front ends. The complexity of the interfaces, if very great, may lead to increased service problems; the more complex the code, the more likely failure is to result more frequently. This may manifest itself by passwords and changes in them losing synchronization, not being reliably passed, or privilege assignment files not being updated uniformly or rapidly. Such problems as these call into question whether SSO was such a good idea, even if it truly was. Complex code is costly to maintain, and the SSO is nothing if not complex. Even the best programming can be rendered ineffective or, worse yet, counterproductive if it is not implemented properly.

An SSO product requires more of this type of attention than most because of its feature-rich complexity. It is clear that the goal of SSO is access control, and in that regard achieves the same goals of confidentiality, integrity, and availability as any other access control system does. SSO products are designed to provide more functionality, but in so doing can adversely affect the environments in which they are installed. If they do, the impacts will most likely appear against factors of reliability, integrity, and performance; and if large enough, the impacts will negate the benefits the SSO provides elsewhere.

Requirements

This section presents the contents of a requirements document that the Georgia Area RACF Users Group (GARUG) put together regarding things it would like to see in an SSO application.

Objectives

The focus of this list is to present a set of functional requirements for the design and development of a trusted single sign-on and security administration product. It is the intention that this be used by security practitioners to determine the effectiveness of the security products they may be reviewing.

It contains many requirements that experienced security users feel are very important to the successful protection of multi-platform systems. It also contains several functional requirements that may not be immediately available at this time. Having said that, the list can be used as a research and development tool because the requirements are being espoused by experienced, working security practitioners in response to real-world problems.

This topic was brought to the forefront by many in the professional security community, and the GARUG members that prepared this list in response. This is not a cookbook to use in the search for security products. In many ways, this list is visionary, which is to say that many of the requirements stated here do not exist. But just because they do not exist now does not deter their inclusion now. As one member noted, "If we don't ask for it, we won't get it."

Functional Requirements

The following is a listing of the functional requirements of an ideal security product on the market. The list also includes many features that security practitioners want to see included in future products. The requirements are broken down in four major categories: security administration management, identification and authorization, access control, and data integrity/confidentiality/encryption. Under each category the requirements are listed in most critical to least critical order.

Assumptions

There are three general assumptions that follow throughout this document.

1. All loginids are unique; no two loginids can be the same. This prevents two users from having the same loginid.
2. The vendor should provide the requisite software to provide functionality on all supported platforms.
3. All vendor products are changing. All products will have to work with various unlike platforms.

Security Administration Management

Single Point of Administration

All administration of the product should be done from a single point. This enables an administrator to provide support for the product from any one platform device.

Ability to Group Users

The product should enable the grouping of like users where possible. These groups should be handled the same way individual users are handled. This will enable more efficient administration of access authority.

Ability to Enforce Enterprise/Global Security Rules

The product should provide the ability to enforce security rules over the entire enterprise, regardless of platform. This will ensure consistent security over resources on all protected platforms.

Audit Trail

All changes, modifications, additions, and deletions should be logged. This ensures that all security changes are recorded for review at a later time.

Ability to Recreate

Information logged by the system should be able to be used to "back out" changes to the security system. Example: used to recreate deleted resources or users. This enables mass changes to be "backed out" of production or enables mass additions or changes to be made based on logged information.

Ability to Trace Access

The product should enable the administrator to be able to trace access to systems, regardless of system or platform.

Scoping and Decentralization of Control

The product should be able to support the creation of spans of control so that administrators can be excluded from or included in certain security control areas within the overall security setup. This enables an administrator to decentralize the administration of security functions based on the groups, nodes, domains, and enterprises over which the decentralized administrator has control.

Administration for Multiple Platforms

The product should provide for the administration of the product for any of the supported platforms. This enables the administrator to support the product for any platform of his or her choice.

Synchronization across All Entities

The product should be synchronizing security data across all entities and all platforms. This ensures that all security decisions are made with up-to-date security information.

Real-Time and Batch Update

All changes should be made online/real-time. The ability to batch changes together is also important to enable easy loading or changing of large numbers of security resources or users.

Common Control Language across All Platforms

The product should feature a common control language across all serviced platforms so that administrators do not have to learn and use different commands on different platforms.

One Single Product

The product should be a single product — not a compendium of several associated products. Modularity for the sake of platform-to-platform compatibility is acceptable and favored.

Flexible Cost

The cost of the product should be reasonable. Several cost scenarios should be considered, such as per seat, CPU, site licensing, and MIPS pricing. Pricing should include disaster recovery scenarios.

Physical Terminal/Node/Address Control

The product should have the ability to restrict or control access on the basis of a terminal, node, or network address. This ability will enable users to provide access control by physical location.

Release Independent/Backward Compatible

All releases of the product should be backward compatible or release independent. Features of new releases should coexist with current features and not require a total reinstallation of the product. This ensures that the time and effort previously invested in the prior release of the product is not lost when a new release is installed.

Software Release Distribution

New releases of the product should be distributed via the network from a single distribution server of the administrator's choice. This enables an administrator to upgrade the product on any platform without physically moving from platform to platform.

Ability to Do Phased Implementation

The product should support a phased implementation to enable administrators to implement the product on individual platforms without affecting other platforms. This will enable installation on a platform-by-platform basis if desired.

Ability to Interface with Application/Database/Network Security

The product should be able to interface with existing application, database, or network security by way of standard security interfaces. This will ensure that the product will mesh with security products already installed.

SQL Reporting

The product should have the ability to use SQL query and reporting tools to produce security setup reports/queries. This feature will enable easy access to security information for administrators.

Ability to Create Security Extract Files

The product should have a feature to produce an extract file of the security structure and the logging/violation records. This enables the administrator to write his or her own reporting systems via SAS or any other language.

Usage Counter per Application/Node/Domain/Enterprise

The product should include an internal counter to maintain the usage count of each application, domain, or enterprise. This enables an administrator to determine which applications, nodes, domains, or enterprises are being used and to what extent they are being used.

Test Facility

The product should include a test facility to enable administrators to test security changes before placing them into production. This ensures that all security changes are fully tested before being placed into production.

Ability to Tag Enterprise/Domain/Node/Application

The product should be able to add a notation or "tag" an enterprise/domain/node/application in order to provide the administrator with a way identify the entity. This enables the administrator to denote the tagged entity and possibly perform extra or nonstandard operations on the entity based on that tag.

Platform Inquiries

The product should support inquiries to the secured platforms regarding the security setup, violations, and other logged events. This will enable an administrator to inquire on security information without having to sign on/log on.

Customize in Real-Time

It is important to have a feature that enables the customization of selected features (those features for which customization is allowed) without reinitializing the product. This feature will ensure that the product is available for 24-hour, seven-day-a-week processing.

GUI Interface

The product should provide a user interface via a Windows-like user interface. The interface may vary slightly between platforms (i.e., Windows, OS/2, X-Windows, etc.) but should retain the same functionality. This facilitates operating consistency and lowers operator and user training requirements.

User-Defined Fields

The product should have a number of user customizable/user-defined fields. This enables a user to provide for informational needs that are specific to his or her organization.

Identification and Authorization

Support RACF Pass Ticket Technology

The product should support IBM's RACF Pass Ticket technology, ensuring that the product can reside in an environment using Pass Ticket technology to provide security identification and authorization.

Support Password Rules (i.e., Aging, Syntax, etc.)

All common password rules should be supported:

- Use or non-use of passwords
- Password length rules
- Password aging rules
- Password change intervals
- Password syntax rules
- Password expiration warning message
- Save previous passwords
- Password uniqueness rules
- Limited number of logons after a password expires
- Customer-defined rules

Logging of All Activity Including Origin/Destination/Application/Platform

All activity should be logged, or able to be logged, for all activities. The logging should include the origin of the logged item or action, the destination, the application involved, and the platform involved. This enables the administrator to provide a concise map of all activity on the enterprise. The degree of logging should be controlled by the administrator.

Single Revoke/Resume for All Platforms

The product should support a single revoke or resume of a loginid, regardless of the platform. This ensures that users can be revoked or resumed with only one command from one source or platform.

Support a Standard Primary loginid Format

The administrator should define all common loginid syntax rules. The product should include features to translate unlike loginids from different platforms so that they can be serviced. This enables the product to handle loginids from systems that support different loginid syntax that cannot be supported natively.

Auto Revoke after X Attempts

Users should be revoked from system access after a specified number of invalid attempts. This threshold should be set by the administrator. This ensures that invalid users are prevented from retrying sign-ons indefinitely.

Capture Point of Origin Information, Including Caller ID/Phone Number for Dial-In Access

The product should be able to capture telephone caller ID (ANI) information if needed. This will provide an administrator increased information that can be acted upon manually or via an exit to provide increased security for chosen ports.

Authorization Server Should Be Portable (Multi-platform)

The product should provide for the authentication server to reside on any platform that the product can control. This provides needed portability if there is a need to move the authentication server to another platform for any reason.

Single Point of Authorization

All authorizations should be made a single point (i.e., an authentication server). The product should not need to go to several versions of the product on several platforms to gain the needed access to a resource. This provides not only a single point of administration for the product, but also reduced network security traffic.

Support User Exits/Options

The product should support the addition of user exits, options, or application programming interfaces (APIs) that could be attached to the base product at strategically identified points of operation. The points would include sign-on, sign-off, resource access check, etc. The enables an administrator or essential technical support personnel to add exit/option code to the package to provide for specific security needs above and beyond the scope of the package.

Ensure loginid Uniqueness

The product should ensure that all loginids are unique; no two loginids can be the same. This prevents two users from having the same loginid.

Source Sign-On Support

The product should support sign-ons from a variety of sources. These sources should include LAN/WAN, workstations, portables (laptops and notebooks), dial-in, and dumb terminals. This would ensure that all potential login sources are enabled to provide login capability and facilitate support for legacy systems.

Customizable Messages

The product should support the use of customized security messages. The will enable an administrator to customize messages to fit the needs of his or her organization.

Access Control

Support Smart Card Tokens

The product should support the use of the common smart card security tokens (i.e., SecureID cards) to enable their use on any platform. The enables the administrator to provide for increased security measures where they are needed for access to the systems.

Ability to Support Scripting — Session Manager Menus

The product should support the use of session manager scripting. This enables the use of a session manager script in those sites and instances where they are needed or required.

Privileges at the Group and System Level

The product should support administration privileges at a group level (based on span of control) or on the system level. This enables the product to be administered by several administrators without the administrators' authority overlapping.

Default Protection Unless Specified

The product should provide for the protection of all resources and entities as the default unless the opposite of protection for only those resources profiled is specified. The enables each organization to determine the best way to install the product based on is own security needs.

Support Masking/Generics

The product should support security profiles containing generic characters that enable the product to make security decisions based on groups of resources as opposed to individual security profiles. The enables the administrator to provide security profiles over many like-named resources with the minimum amount of administration.

Allow Delegation within Power of Authority

The product should allow an administrator to delegate security administration authority to others at the discretion of the administrator within his or her span of authority. An administrator would have the ability to give some of his or her security authority to another administrator for backup purposes.

Data Integrity/Confidentiality/Encryption

No Cleartext Passwords (Net or DB) — Dumb Terminal Exception

At no time should any password be available on the network or in the security database in clear, human-readable form. The only exception is the use of dumb terminals where the terminal does not support encryption techniques. This will ensure the integrity of the users' passwords in all cases with the exception of dumb terminals.

Option to Have One or Distributed Security DBs

The product should support the option of having a single security database or several distributed security databases on different platforms. This enables an administrator to use a distributed database on a platform that may be sensitive to increased activity rather than a single security database. The administrator will control who can and if they can update distributed databases.

Inactive User Timeout

All users who are inactive for a set period during a session should be timed out and signed off of all sessions. This ensures that a user who becomes inactive for whatever reason does not compromise the security of the system by providing an open terminal to a system. This feature should be controlled by the administrator and have two layers:

1. At the session manager/screen level
2. At the application/platform level

Inactive User Revoke

All users who have not signed on within a set period should be revoked. This period should be configurable by the administrator. This will ensure that loginids are not valid if not used within a set period of time.

Ability to Back Up Security DBs to Choice of Platforms/Media

The product should be able to back up its security database to a choice of supported platforms or storage media. This enables the user to have a variety of destinations available for the security database backup.

Encryption Should Be Commercial Standard (Presently DES)

The encryption used in the product should be standard. That standard is presently DES but could change as new encryption standards are made. This will ensure that the product will be based on a tested, generally accepted encryption base.

Integrity of Security DB(s)

The database used by the product to store security information and parameters should be protected from changes via any source other than the product itself. Generic file edit tools should not be able to view or update the security database.

Optional Application Data Encryption

The product should provide the optional ability to interface to encrypted application data if the encryption techniques are provided. This enables the product to interact with encrypted data from existing applications.

Failsoft Ability

The product should have the ability to perform at a degraded degree without access to the security database. This ability should rely on administrator input on an as-needed basis to enable a user to sign on, access resources, and sign off. This enables the product to at least work in a degraded mode in an emergency in such a fashion that security is not compromised.

Conclusion

Single sign-on (SSO) can indeed be the answer to an array of user administration and access control problems. For the user, it *might* be a godsend. It is, however, not a straightforward or inexpensive solution. As with other so-called "enterprise security solutions," there remain the problems of scalability and phasing-in. There is generally no half-step to be taken in terms of how such a technology as this is rolled out. It is of course possible to limit it to a single platform, but that negates the whole point of doing SSO in the first place.

Like all solutions, SSO must have a real problem that it addresses. Initially regarded as a solution looking for a problem, SSO has broadened its scope to address more than simply the avalanche of loginids and passwords users seem to acquire in their systems travels. This greater functionality can provide much needed assistance and control in managing the user, his access rights, and the trail of activity left in his wake. This however comes at a cost.

Some significant observations made by others regarding SSO became apparent from an informal survey conducted by this author. The first is that it can be very expensive, based mostly on the scope of the implementation. The second is that it can be a solution looking for a problem — meaning that it sounds like a "really neat" technology (which it is) that proffers religion on some. This "religion" tends to be a real cause for concern in the manager or CIO over the IT function, for reasons that are well-understood. When the first conjoins with the second, the result is frequently substantial project scope creep — usually a very sad story with an unhappy ending in the IT world.

The third observation was more subtle, but more interesting. Although several vendors still offer an SSO product as an add-on, the trend appears to be more toward SSO slowly disappearing as a unique product. Instead, this capability is being included in platform or enterprise IT management solution software such as Tivoli (IBM) and Unicenter-TNG (Computer Associates). Given the fact that SSO products support most of the functions endemic to PKI, the other likelihood in the author's opinion is that SSO will be subsumed into the enterprise PKI solution and thus become a "feature" rather than a "product."

It does seem certain that this technology will continue to mature and improve, and eventually become more widely used. As more and more experience is gained in implementation endeavors, the files of "lessons learned"

will grow large with many painful implementation horror stories. Such stories often arise from "bad products badly constructed." Just as often, they arise from poorly managed implementation projects. SSO will suffer, and has, from the same bad rap — partially deserved, partially not. The point here is: do your homework, select the right tool for the right job, plan your work carefully, and execute thoroughly. It will probably still be difficult, but one might actually get the results one wants.

In the mystical and arcane practice of information security, many different tools and technologies have acquired that rarified and undeserved status known as "panacea." In virtually no case has any one of them fully lived up to this unreasonable expectation, and the family of products providing the function known as "single sign-on" is no exception.

Single Sign-on

Ross Leo

CORPORATIONS EVERYWHERE HAVE MADE THE FUNCTIONAL SHIFT FROM THE MAINFRAME-CENTERED DATA PROCESSING ENVIRONMENT TO THE CLIENT/SERVER CONFIGURATION. With this conversion have come new economies, a greater variety of operational options, and a new set of challenges. In the mainframe-centric installation, systems management was often the administrative twin of the computing complex itself: the components of the system were confined to one area, as were those who performed the administration of the system. In the distributed client/server arrangement, those who manage the systems are again arranged in a similar fashion. This distributed infrastructure has complicated operations, even to the extent of making the simple act of logging in more difficult.

Users need access to many different systems and applications to accomplish their work. Getting them set up to do this simply and easily is frequently time-consuming, requiring coordination between several individuals across multiple systems. In the mainframe environment, switching between these systems and applications meant returning to a main menu and making a new selection. In the client/server world, this can mean logging in to an entirely different system. New loginid, new password, and both very likely different than the ones used for the previous system — the user is inundated with these, and the problem of keeping them un-confused to prevent failed log-in attempts. It was because of this and related problems that the concept of the **Single Sign-on**, or SSO, was born.

EVOLUTION

Given the diversity of computing platforms, operating systems, and access control software (and the many loginids and passwords that go with them), having the capability to log on to multiple systems once and simultaneously through a single transaction would seem an answer to a prayer. Such a prayer is one offered by users and access control administrators everywhere. When the concept arose of a method to accomplish this, it became clear that integrating it with the different forms of system access control would pose a daunting challenge with many hurdles.

In the days when applications software ran on a single platform, such as the early days of the mainframe, there was by default only a single login that users had to perform. Whether the application was batch oriented or interactive, the user had only a single loginid and password combination to remember. When the time came for changing passwords, the user could often make up his own. The worst thing to face was the random password generator software implemented by some companies that served up number/letter combinations. Even then, there was only one of them.

The next step was the addition of multiple computers of the same type on the same network. While these machines did not always communicate with each other, the user had to access more than one of them to fulfill all data requirements. Multiple systems, even of the same type, often had different rules of use. Different groups within the Data Processing Department often controlled these disparate systems and sometimes completely separate organizations with the same company. Of course, the user had to have a different loginid and password for each one, although each system was reachable from the same terminal.

Then, the so-called "departmental computer" appeared. These smaller, less powerful processors served specific groups in the company to run unique applications specific to that department. Examples include materials management, accounting and finance applications, centralized word-processing, and shop-floor applications. Given the limited needs of these areas, and the fact that they frequently communicated electronically internal to themselves, tying these systems together on the same network was unnecessary. This state of affairs did not last long.

It soon became obvious that tying these systems together, and allowing them to communicate with each other over the network would speed up the information flow from one area to another. Instead of having to wait until the last week of the month to get a report through internal mail, purchasing records could be reconciled weekly with inventory records for materials received the same week from batched reports sent to Purchasing. This next phase in the process of information flow did not last long either.

As systems became less and less batch oriented and more interactive, and business pressures to record the movement of goods, services, and money mounted, more rapid access was demanded. Users in one area needed direct access to information in another. There was just one problem with this scenario — and it was not a small one.

Computers have nearly always come in predominantly two different flavors: the general-purpose machines and specific-use machines. Initially called "business processing systems" and "scientific and engineering systems", these computers began the divergence from a single protocol and single operating system that continues today. For a single user to have access

to both often required two separate networks because each ran on a different protocol. This of course meant two different terminals on that user's desk. That all the systems came from the same manufacturer was immaterial: the systems could not be combined on the same wire or workstation.

The next stage in the evolution was to hook in various types of adapters, multiple screen "windowed" displays, protocol converters, etc. These devices sometimes eliminated the second terminal. Then came the now-ubiquitous personal computer, or "PC" as it was first called when it was introduced by IBM on August 12, 1981. Within a few short years, adapters appeared that permitted this indispensable device to connect and display information from nearly every type of larger host computer then in service. Another godsend had hit the end user!

This evolution has continued to the present day. Most proprietary protocols have gone the way of the woolly Mammoth, and have resolved down to a precious few, nearly all of them speaking TCP/IP in some form. This convergence is extremely significant: the basic method of linking all these different computing platforms together with a common protocol on the same wire exists.

The advent of Microsoft Windows pushed this convergence one very large step further. Just as protocols had come together, so too the capability of displaying sessions with the different computers was materializing. With refinement, the graphical user interface ("GUI" — same as gooey) enabled simultaneous displays from different hosts. Once virtual memory became a reality on the PC, this pushed this envelope further still by permitting simultaneous active displays and processing.

Users were getting capabilities they had wanted and needed for years. Now impossible tasks with impossible deadlines were rendered normal, even routine. But despite all the progress that had been made, the real issue had yet to be addressed. True to form, users were grateful for all the new toys and the ease of use they promised … until they woke up and found that none of these innovations fixed the thing they had complained most and loudest about: multiple loginids and passwords.

So what is single sign-on?

WHAT SINGLE SIGN-ON IS: THE BEGINNING

Beginning nearly 50 years ago, system designers realized that a method of tracking interaction with computer systems was needed, and so a form of identification — the loginid — was conceived. Almost simultaneously with this came the password — that sometimes arcane companion to the loginid that authenticates, or confirms the identity of, the user. And for most of the past five decades, a single loginid and its associated password was sufficient to assist the user in gaining access to virtually all the computing power then

available, and to all the applications and systems that user was likely to use. Yes, those were the days… simple, straightforward, and easy to administer. And now they are all but gone, much like the club moss, the vacuum tube, and MS/DOS (perhaps).

Today's environment is more distributed in terms of both geography and platform. Although some will dispute, the attributes differentiating one operating system from another are being obscured by both network access and graphical user interfaces (the ubiquitous GUI). Because not every developer has chosen to offer his or her particular application on every computing platform (and networks have evolved to the point of being seemingly oblivious to this diversity), users now have access to a broader range of tools spread across more platforms, more transparently than at any time in the past. And yet all is not paradise.

Along with this wealth of power and utility comes the same requirement as before: to identify and authenticate the user. But now this must be done across all these various systems and platforms, and (no surprise) they all have differing mechanisms to accomplish this. The result is that users now have multiple loginids, each with its own unique password, quite probably governed by its equally unique set of rules. The CISSP knows that users complain bitterly about this situation, and will often attempt to circumvent it by whatever means necessary. To avoid this, the CISSP had to find a solution. To facilitate this, and take advantage of a marketing opportunity, software vendors saw a vital need, and thus the single sign-on (SSO) was conceived to address these issues.

Exhibit 1-1 shows where SSO was featured in the overall security program when it first appeared. As an access control method, SSO addressed important needs across multiple platforms (user identification and authentication). It was frequently regarded as a "user convenience" that was difficult and costly to implement, and of questionable value in terms of its contribution to the overall information protection and control structure.

THE ESSENTIAL PROBLEM

In simplest terms, too many loginids and passwords, and a host of other user access administration issues. With complex management structures requiring a geographically dispersed matrix approach to oversee employee work, distributed and often very different systems are necessary to meet operational objectives and reporting requirements.

In the days of largely mainframe-oriented systems, a problem of this sort was virtually nonexistent. Standards were made and enforcement was not complex. In these days, such conditions carry the same mandate for the establishment and enforcement of various system standards. Now, however, such conditions, and the systems arising in them, are of themselves not naturally conducive to this.

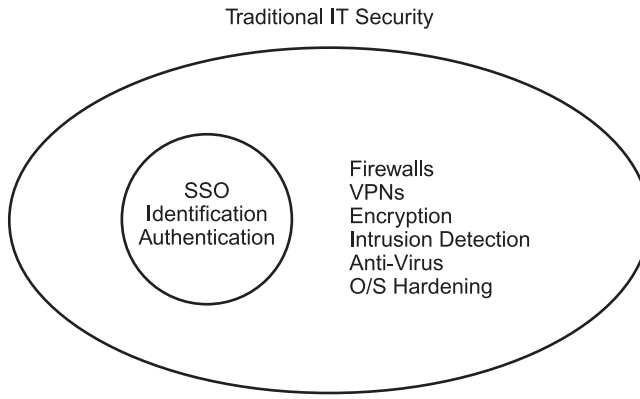


Exhibit 1-1. Single sign-on: in the beginning.

As mentioned above, such systems have different built-in systems for tracking user activity. The basic concepts are similar: audit trail, access control rule sets, Access Control Lists (ACLs), parameters governing system privilege levels, etc. In the end, it becomes apparent that one set of rules and standards, while sound in theory, may be exceedingly difficult to implement across all platforms without creating unmanageable complexity. It is however the "Holy Grail" that enterprise-level user administrators seek.

Despite the seeming simplicity of this problem, it represents only the tip of a range of problems associated with user administration. Such problems exist wherever the controlling access of users to resources is enforced: local in-house, remote WAN nodes, remote dial-in, and Web-based access.

As compared with Exhibit 1-1, Exhibit 1-2 illustrates how SSO has evolved into a broader scope product with greater functionality. Once considered merely a "user convenience," SSO has been more tightly integrated with other, more traditional security products and capabilities. This evolution has improved SSO's image measurably, but has not simplified its implementation.

In addition to the problem mentioned above, the need for this type of capability manifests itself in a variety of ways, some of which include:

1. As the number of entry points increases (Internet included), there is a need to implement improved and auditable security controls.
2. The management of large numbers of workstations is dictating that some control be placed over how they are used to avoid viruses, limit user-introduced problems, minimize help desk resources, etc.
3. As workstations have become electronic assistants, there has likewise arisen a need for end users to be able to use various workstations along their work path to reach their electronic desktop.

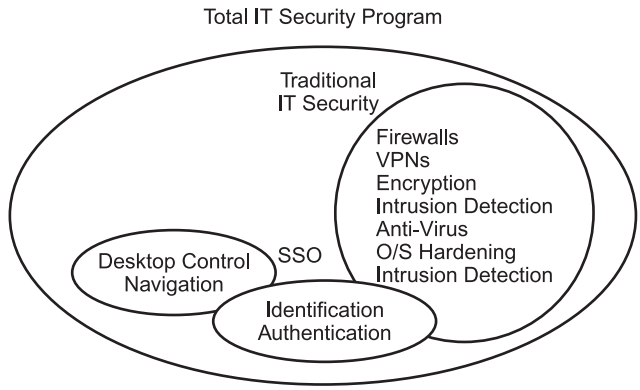


Exhibit 1-2. The evolution of SSO.

4. The proliferation of applications has made getting to all the information that is required too difficult, too cumbersome, or too time-consuming, even after passwords are automated.
5. The administration of security needs to move from an application focus to a global focus to improve compliance with industry guidelines and to increase efficiency.

MECHANISMS

The mechanisms used to implement SSO have varied over time. One method uses the Kerberos product to authenticate users and resources to each other through a "ticketing" system; tickets being the vehicle through which authorization to systems and resources is granted. Another method has been shells and scripting: primary authentication to the shell, which then initiated various platform-specific scripts to activate account and resource access on the target platforms.

For those organizations not wanting to expend the time and effort involved with a Kerberos implementation, the final solution was likely to be a variation of the shell-and-script approach. This had several drawbacks. It did not remove the need to set up user accounts individually on each platform. It also did not provide password synchronization or other management features. Shell-and-scripting was a half-step at best, and although it simplified user login, that was about the extent of the automation it facilitated. That was "then."

Today, different configuration approaches and options are available when implementing an SSO platform, and the drawbacks of the previous attempts have largely been well-addressed. Regardless, from the security engineering perspective, the design and objectives (i.e., the problem one is trying to solve) for the implementation plan must be evaluated in a risk

analysis, and then mitigated as warranted. In the case of SSO, the operational concerns should also be evaluated, as discussed below.

One form of implementation allows one login session, which concludes with the user being actively connected to the full range of their authorized resources until logout. This type of configuration allows for reauthentication based on time (every … minutes or hours) or can be event driven (i.e., system boundary crossing).

One concern with this configuration is resource utilization. This is because a lot of network traffic is generated during login, directory/ACL accesses are performed, and several application/system sessions are established. This level of activity will degrade overall system performance substantially, especially if several users engage their login attempts simultaneously. Prevention of session loss (due to inactivity timeouts) would likely require an occasional "ping" to prevent this, if the feature itself cannot be deactivated. This too consumes resources with additional network traffic.

The other major concern with this approach would be that "open sessions" would exist, regardless of whether the user is active in a given application or not. This might make possible "session stealing" should the data stream be invaded, penetrated or rerouted.

Another potential configuration would perform the initial identification/authentication to the network service, but would not initialize access to a specific system or application until the user explicitly requests it (i.e., double-click the related desktop icon). This would reduce the network traffic level, and would invoke new sessions only when requested. The periodic reauthentication would still apply.

What Single Sign-on Provides

SSO products have moved beyond simple end-user authentication and password management to more complex issues that include addressing the centralized administration of endpoint systems, the administration of end users through a role-based view that allows large populations of end users to be affected by a single system administration change (e.g., adding a new application to all office workers), and the monitoring of end users' usage of sensitive applications.

The next section describes many of the capabilities and features that an ideal single sign-on product might offer. Some of the items that mention cost refer expressly to the point being made, and not to the software performing the function. The life-cycle cost of a product such as that discussed here can and does vary widely from one installation to the next. The extent of such variation is based on many factors, and is well beyond the scope of this discussion.

A major concern with applying the SSO product to achieve the potential economies is raised when consideration is given to the cost of the product, and comparing it to the cost of how things were done pre-SSO, and contrasting this with the cost of how things will be done post-SSO, the cost of putting SSO in, and all other dollars expended in the course of project completion.

By comparing the before-and-after expenditures, the ROI (return on investment) for installing the SSO can be calculated and used as part of the justification for the project. It is recommended that this be done using equivalent formulas, constraints, and investment/ROI objectives the enterprise applies when considering any project. When the analysis and results are presented (assuming they favor this undertaking), the audience will have better insight into the soundness of the investment in terms of real costs and real value contribution. Such insight fosters endorsement, and favors greater acceptance of what will likely be a substantial cost and lengthy implementation timeline.

Regardless, it is reasonably accurate to say that this technology is neither cheap to acquire nor to maintain. In addition, as with any problem-solution set, the question must be asked, "Is this problem worth the price of the solution?" The next section discusses some of the features to assist in making such a decision.

Internal Capability Foundation

Having GUI-based central administration offers the potential for simplified user management, and thus possibly substantial cost-savings in reduced training, reduced administrative effort, and lower life-cycle cost for user management. This would have beneath it a logging capability that, based on some DBMS engine and a set of report generation tools, would enhance and streamline the data reduction process for activity reporting and forensic analysis derived through the SSO product.

The basic support structure must include direct (standard customary login) and Web-based access. This would be standard, especially now that the Internet has become so prolific and also since an increasing number of applications are using some form of Web-enabled/aware interface. This means that the SSO implementation would necessarily limit the scope or depth of the login process to make remote access practical, whether direct dial-up or via the Web.

One aspect of concern is the intrusiveness of the implementation. Intrusiveness is the extent to which the operating environment must be modified to accommodate the functionality of the product. Another is the retrofitting of legacy systems and applications. Installation of the SSO product

on the various platforms in the enterprise would generally be done through APIs to minimize the level of custom code.

Not surprisingly, most SSO solutions vendors developed their product with the retrofit of legacy systems in mind. For example, the Platinum Technologies (now CA) product AutoSecure SSO supported RACF, ACF2, and TopSecret — all of which are access control applications born and bred in the legacy systems world. It also supports Windows NT, Novell, and TCP/IP network-supported systems. Thus, it covers the range from present day to legacy.

General Characteristics

The right SSO product should provide all the required features and sustain itself in an enterprise production environment. Products that operate in an open systems distributed computing environment, complete with parallel network servers, are better positioned to address enterprise needs than more narrow NOS-based SSO products.

It is obvious then that SSO products must be able to support a fairly broad array of systems, devices, and interfaces if the promise of this technology is to be realized. Given that, it is clear some environments will require greater modification than others; that is, the SSO configuration is more complex and modifies the operating environment to a greater extent. Information derived through the following questions will assist in pre-implementation analysis:

1. Is the SSO nonintrusive; that is, can it manage access to all applications, without a need to change the applications in any way?
2. Does the SSO product dictate a single common logon and password across all applications?
3. What workstations are supported by the SSO product?
4. On what operating systems can SSO network servers operate?
5. What physical identification technologies are supported (e.g., Secure-ID card)?
6. Are dial-up end users supported?
7. Is Internet access supported? If so, are authentication and encryption enforced?
8. Can the SSO desktop optionally replace the standard desktop to more closely control the usage of particular workstations (e.g., in the production area)?
9. Can passwords be automatically captured the first time an end user uses an endpoint application under the SSO product's control?
10. Can the look of the SSO desktop be replaced with a custom site-specific desktop look?
11. How will the SSO work with the PKI framework already installed?

End-User Management Facilities

These features and options include the normal suite of functions for account creation, password management, etc. The performance of end-user identification and authentication is obvious. Password management includes all the normal features: password aging, histories, and syntax rules. To complete the picture, support for the wide variety of token-type devices (Secure-ID cards), biometric devices, and the like should be considered, especially if remote end users are going to be using the SSO product. At the very least, optional modules providing this support should exist and be available.

Some additional attributes that should be available are:

- *Role-based privileges:* this functionality makes it possible to administer a limited number of roles that are in turn shared by a large population of end users. This would not necessarily have any effect on individual users working outside the authority scope of that role.
- *Desktop control:* this allows the native desktop to be replaced by an SSO-managed desktop, thereby preventing end users from using the workstation in such a way as to create support problems (e.g., introducing unauthorized software). This capability is particularly important in areas where workstations are shared by end users (e.g., production floor).
- *Application authorization:* this ensures that any launched application is registered and cleared by the SSO product and records are kept of individual application usage.
- *Mobile user support:* this capability allows end users to reach their desktop, independent of their location or the workstation they are using. It should also include configuring the workstation to access the proper domain server and bringing the individual's preferences to the workstation before launching applications.

Application Management Facilities

Application management in the context of SSO refers to the treatment of an application in a manner similar to how it manages or treats users. As shown in Figure 1-2, the evolved state of SSO has moved beyond the simplistic identification/authentication of users, and now encompasses certain aspects of application management. This management capability relates to the appearance of user desktops and navigation through application menus and interfaces rather than with the maintenance and upgrading of application functionality.

Context management ensures that when multiple sessions that relate to a common subject are simultaneously active, each session is automatically updated when another related session changes position (e.g., in a healthcare setting, the lab and pharmacy sessions must be on the same patient if

the clinician is to avoid mixing two patients' records when reaching a clinical decision).

Application monitoring is particularly useful when it is desirable to monitor the usage of particular rows of information in an application that is not programmed to provide that type of information (e.g., access to particular constituents' records in a government setting).

Application positioning is a feature that relates to personalized yet centrally controlled desktops. This allows configuration of an end-user start-up script to open an application (possibly chosen from a set of options) on initialization, and specify even what screen is loaded.

One other feature that binds applications together is application fusing. This allows applications to operate in unison such that the end user is only aware of a single session. The view to the end user can range from a simple automated switching between applications up to and including creating an entirely new view for the end user.

Endpoint Management Facilities

Endpoint administration is an essential component of an SSO product because, without it, administration is forced to input the same information twice; once in the SSO and once in the endpoint each time a change is made to the SSO database. Two methods of input into the endpoint should be supported: (1) API-based agents to update endpoint systems that support an API, and (2) session animation agents to update endpoint systems that do not support an API. Services provided by the SSO to accomplish this administrative goal should include:

- *Access control:* this is the vehicle used by end users to gain access to applications and, based on each application's capabilities, to define to the application the end user's privileges within it. Both API-based and session-based applications should be supported.
- *Audit services:* these should be made available through an API to endpoint applications that wish to publish information into the SSO product's logging system.
- *Session encryption:* this feature ensures information is protected from disclosure and tampering as it moves between applications and end users. This capability should be a requirement in situations where sensitive applications only offer cleartext facilities.

Mobile Users

The capability for end users to use any available workstation to reach information sources is mandatory in environments where end users are expected to function in a number of different locations. Such users would include traveling employees, health care providers (mobile nurses,

physicians, and technicians), consultants, and sales staff. In the highly mobile workforce of today's world, it is unlikely that a product not offering this feature would be successful.

Another possible feature would facilitate workstation sharing; that is, the sharing of the device by multiple simultaneous users, each one with their own active session separate from all others. This capability would entail the use of a form of screen swapping so that loginids and passwords would not be shared. When the first user finishes his session, rather than logout, he locks the session, a hot-key combination switches to the next open login screen, and the second user initiates his session, etc.

When investigating the potential needs in this regard, the questions to ask yourself and the vendors of such products should include:

1. Can a workstation in a common area be shared by many end users (e.g., production floor)?
2. If someone wants to use a workstation already in use by another end user, can the SSO product gracefully close the existing end user's applications (including closing open documents) and turn control over to the new end user?
3. Can end users adjust the organization of their desktop, and if so, does it travel with them, independent of the workstation they use?
4. Can individual applications preferences travel with the end user to other workstations (e.g., MS Word preferences)?
5. Can the set of available applications be configured to vary based on the entry point of the end user into the network?
6. If a Novell end user is logging in at a workstation that is assigned to a different Novell domain, how does the end user get back to his or her domain?
7. Given that Windows 95 and Windows NT rely on a locally stored password for authentication, what happens when the end user logs onto another workstation?
8. Is the date and time of the last successful sign-on shown at the time the end user signs on to highlight unauthorized sign-ons?
9. Is the name of the logged in end user prominently displayed to avoid inadvertent use of workstations by other end users?

Authentication

Authentication ensures that users are who are who they claim to be. It also ensures that all processes and transactions are initiated only by authorized end users. User authentication couples the loginid and the password, providing an identifier for the user, a mechanism for assigning access privileges, and an auditing "marker" for the system against which to track all activity, such as file accesses, process initiation, and other actions

(e.g., attempted logons). Thus, through the process of authentication, one has the means to control and track the "who" and the "what."

The SSO products take this process and enable it to be used for additional services that enhance and extend the applications of the loginid/password combination. Some of these applications provide a convenience for the user that also improves security: the ability to lock the workstation just before stepping away briefly means the user is more likely to do it, rather than leave his workstation open for abuse by another. Some are extensions of audit tools: display of last login attempt, and log entry of all sign-ons. These features are certainly not unique to SSO, but they extend and enhance its functionality, and thus make it more user friendly.

As part of a Public Key Infrastructure (PKI) installation, the SSO should have the capability to support digital certificate authentication. Through a variety of methods (token, password input, biometrics possibly), the SSO supplies a digital certificate for the user that the system then uses as both an authenticator and an access privilege "license" in a fashion similar to the Kerberos ticket. The vital point here is not how this functionality is actually performed (that is another lengthy discussion), but that the SSO supports and integrates with a PKI, and that it uses widely recognized standards in doing so.

It should noted, however, that any SSO product that offers less than the standard suite of features obtainable through the more common access control programs should *not* be considered. Such a product may be offered as an alternative to the more richly featured SSO products on the premise that "simpler is better." Simpler is not better in this case because it means reduced effectiveness.

To know whether the candidates measure up, an inquiry should be made regarding these aspects:

1. Is authentication done at a network server or in the workstation?
2. Is authentication done with a proven and accepted standard (e.g., Kerberos)?
3. Are all sign-on attempts logged?
4. After a site-specified number of failed sign-on attempts, can all future sign-on attempts be unconditionally rejected?
5. Is an inactivity timer available to lock or close the desktop when there is a lack of activity for a period of time?
6. Can the desktop be easily locked or closed when someone leaves a workstation (e.g., depression of single key)?
7. Is the date and time of the last successful sign-on shown at the time the end user signs on to highlight unauthorized sign-ons?

Encryption

Encryption ensures that information that flows between the end users and the security server(s) and endpoint applications they access is not intercepted through spying, line-tapping, or some other method of eavesdropping. Many SSO products encrypt traffic between the end user and the security server but let cleartext pass between the end user and the endpoint applications, causing a potential security gap to exist. Some products by default encrypt all traffic between workstation and server, some do not, and still others provide this feature as an option that is selectable at installation.

Each installation is different in its environment and requirements. The same holds true when it comes to risks and vulnerabilities. Points to cover that address this include:

- Is all traffic between the workstation and the SSO server encrypted?
- Can the SSO product provide encryption all the way to the endpoint applications (e.g., computer room) without requiring changes to the endpoint applications?
- Is the data stream encrypted using an accepted and proven standard algorithm (e.g., DES, Triple DES, IDEA, AES, or other)?

Access Control

End users should only be presented with the applications they are authorized to access. Activities required to launch these applications should be carefully evaluated because many SSO products assume that only API-based endpoint applications can participate, or that the SSO is the owner of a single password that all endpoint applications must comply with. These activities include automatically inputting and updating application passwords when they expire.

Exhibit 1-3 shows how the SSO facilitates automatic login and acquisition of all resources to which a user is authorized. The user logs into the authentication server (centrally positioned on the network). This then validates the user and his access rights. The server then sends out the validated credentials and activates the required scripts to log the user in and attach his resources to the initiated session.

While it is certainly true that automatically generated passwords might make the user's life easier, current best practice is to allow users to create and use their own passwords. Along with this should be a rule set governing the syntax of those passwords; for example, no dictionary words, a combination of numbers and letters, a mixture of case among the letters, no repetition within a certain number of password generations, proscribed use of special characters (#, $, &, ?, %, etc.), and other rules. The SSO

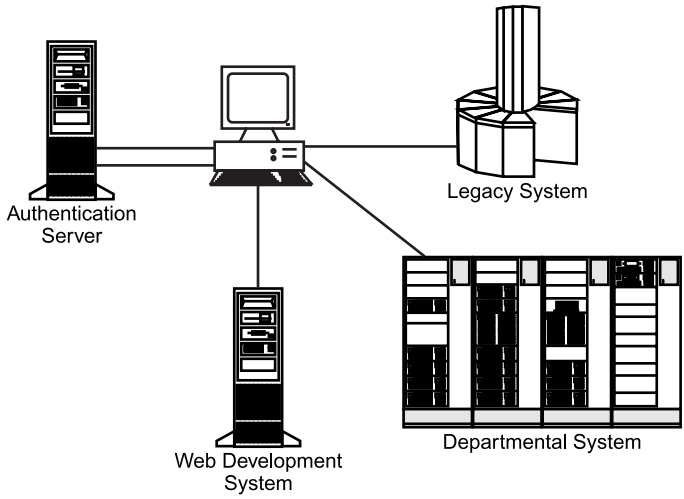


Exhibit 1-3. Automated login.

should support this function across all intended interfaces to systems and applications.

Exhibit 1-4 shows how the SSO facilitates login over the World Wide Web (WWW) by making use of cookies — small information packets shipped back and forth over the Web. The user logs into the initial Web server (1), which then activates an agent that retrieves the user's credentials from the credentials server (2). This server is similar in function to a name server or an LDAP server, except that this device provides authorization and access privileges information specifically. The cookie is then built and stored in the user's machine (3), and is used to revalidate the user each time a page transition is made.

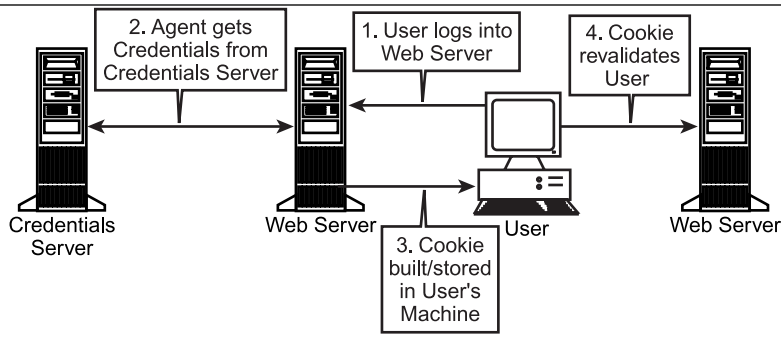


Exhibit 1-4. SSO: Web with cookies.

This process is similar to verification of application-level privileges inside a DBMS. While moving within the database system, each time the user accesses a new region or transaction, access privileges must be reverified to ensure correct authorization. Page transitions on the Web equate to new regions or transactions within the DBMS.

In this area, the following points should be covered:

1. Can all applications, regardless of platform, be nonintrusively supported (i.e., without changing them, either extensively or at all)?
2. What types of adapters are available to mechanize the application launching process without having to adjust the individual applications? Are API-based, OLE-based, DDE-based, scripting-based, and session-simulation adapters available?
3. Are all application activations and deactivations logged?
4. When application passwords expire, does the SSO product automatically generate new expired one-time passwords or are users able to select and enter their own choices?
5. When an application is activated, can information be used to navigate to the proper position in the application (e.g., order entry application is positioned to the order entry screen)?
6. Can the application activation procedure be hidden from the end user or does the end user have to see the mechanized process as it progresses?
7. Are inactivity timers available to terminate an application when there is a lack of activity for a period of time?

Application Control

Application control limits end users' use of applications in such a way that only particular screens within a given application are visible, only specific records can be requested, and particular uses of the applications can be recorded for audit purposes, transparently to the endpoint applications so no changes are needed to the applications involved.

As a way in which user navigation is controlled, this is another feature that can assist with enhancing the overall security posture of an installation. Again, this would be as an adjunct feature — not the key method. The determination of the usefulness of this capability can be made through the following questions.

1. Can applets be incorporated into the desktop's presentation space (e.g., list of major accounts)?
2. Can applet information (e.g., particular account) be used to navigate to the proper position within an application (e.g., list of orders outstanding for a particular customer)?

3. Can each application's view be adjusted to show only the information that is appropriate for a particular end user?
4. Can the SSO product log end users' activities inside applications (e.g., which accounts have been accessed)?
5. Can application screens be enhanced with new capabilities without having to change the applications themselves (e.g., additional validation of input as it is captured)?
6. Can the SSO product log attempt to reach areas of applications that go beyond permitted areas (e.g., confidential patient information)?
7. Can multiple applications be fused into a single end-user session to eliminate the need for end users to learn each application?
8. Can applications be automatically coordinated such that end-user movement in one application (e.g., billing) automatically repositions subordinate application sessions (e.g., current orders, accounts receivable)?

Administration

The centralized administration capabilities offered by the SSO are — if not the main attraction — the "Holy Grail" mentioned earlier. The management (creation, modification, deletion) of user accounts and resource profiles through an SSO product can streamline and simplify this function within an organization or enterprise. The power of the administration tools is key because the cost of administering a large population of end users can easily overshadow the cost of the SSO product itself.

The product analysis should take the following attributes into consideration:

1. Does the SSO product allow for the central administration of all endpoint systems? (That is, changes to the central administration database are automatically reflected in endpoint systems.)
2. Is administration done at an "end-user" or a "role within the enterprise" level? (This is a critical element because an end-user focus can result in disproportional administration effort.)
3. Does each workstation have to be individually installed? If so, what is the estimated time required?
4. Can end users' roles in the organization be easily changed (to deal with people that perform mixed roles)?
5. Is the desktop automatically adjusted if the end user's roles are changed, or does the desktop view have to be adjusted manually?
6. Can an administrator see a list of active end users by application?
7. Can an administrator access all granted passwords to specific endpoint applications?
8. Does the product gracefully deal with network server failures?

Services for Desktop-Aware Applications

In cases where it is possible to modify existing endpoint applications, the ability for them to cooperatively share responsibilities with the desktop is very attractive. What is required is a published desktop API and associated services.

The circumstance can and does arise where the end user wants to customize a standard product in the enterprise suite for his own use in a way that affects only him and does not change the basic application itself. Such customization may include display formats, scripts, and processes relating to specific tasks the individual user wants or needs to use in conjunction with the server-supplied application. Through the supplied API, the user can make the custom changes necessary without impediment, and this allows other users to proceed without affecting them or their workstations.

In such cases, the user wanting the changes may require specific access and other controls to lock out other users. An example might be one where the user requiring the changes works on sensitive or restricted information, and others in the same area do not, and are not permitted access to such. This then may necessitate the use of access controls embedded in the scripts used to change his desktop to meet his additional security needs.

That being the case, the API should provide the capability to access the SSO, and perform the access/privilege checking, without the user (the one making the localized changes) having any direct access to the SSO access/privilege database. This should likewise be true to facilitate the logging of access attempts, transactions, and data access authorizations to track the use of the local workstation. To determine the existence of this facility in the SSO, questions should be asked regarding such services, APIs, and related capabilities, such as

1. Can desktop-aware applications interrogate end-user permissions managed by the SSO product?
2. Can desktop-aware applications make use the SSO product's logging facilities for their own use?
3. Do API services exist that enable desktop customization?
4. Do these APIs facilitate this without compromising overall system integrity by providing "back-door" access to the resident security information database?

Reliability and Performance

Given that an SSO product is, by necessity, positioned between the end users and the applications they need access to get their jobs done, it has a very high visibility within the enterprise and any unexpected reliability or performance problems can have serious consequences. This issue points directly back at the original business case made to justify the product.

Concerns with regard to reliability and performance generally focus on the additional layering of one software upon another ("yet another layer"), the interfaces between the SSO and other access control programs it touches, the complexity of these interactions, etc. One aspect of concern is the increased latency introduced by this new layer. The time from power-on to login screen has steadily increased over the years, and the addition of the SSO may increase it yet again. This can exacerbate user frustration.

The question of reliability arises when considering the interaction between the SSO and the other security frontends. The complexity of the interfaces, if very great, may lead to increased service problems; the more complex the code, the more likely failure is to result more frequently. This may manifest itself by passwords and changes in them losing synchronization, not being reliably passed, or privilege assignment files not being updated uniformly or rapidly. Such problems as these call into question whether SSO was such a good idea, even if it truly was. Complex code is costly to maintain, and the SSO is nothing if not complex. Even the best programming can be rendered ineffective, or worse yet counterproductive, if it is not implemented properly.

An SSO product requires more of this type of attention than most because of its feature-rich complexity. It is clear that the goal of the sso is access control, and in that regard achieves the same goals of confidentiality, integrity, and availability as any other access control system does. SSO products are designed to provide more functionality, but in so doing can adversely affect the environments in which they are installed. If they do, the impacts will most likely appear against factors of reliability, integrity, and performance; and if large enough, the impacts will negate the benefits the SSO provides elsewhere.

REQUIREMENTS

This section presents the contents of a requirements document that the Georgia Area RACF Users Group (GARUG) put together regarding things they would like to see in an SSO application.

Objectives

The focus of this list is to present a set of functional requirements for the design and development of a trusted single sign-on and security administration product. It is the intention that this be used by security practitioners to determine the effectiveness of the security products they may be reviewing.

It contains many requirements that experienced security users feel are very important to the successful protection of multi-platform systems. It also contains several functional requirements that may not be immediately

available at this time. Having said that, the list can be used as a research and development tool because the requirements are being espoused by experienced, working security practitioners in response to real-world problems.

This topic was brought to the forefront by many in the professional security community, and the GARUG members that prepared this list in response. This is not a cookbook to use in the search for security products. In many ways, this list is visionary, which is to say that many of the requirements stated here do not exist. But just because they do not exist now does not deter their inclusion now. As one member noted, "If we don't ask for it, we won't get it."

Functional Requirements

The following is a listing of the functional requirements of an ideal security product on the market. The list also includes many features that security practitioners want to see included in future products. The requirements are broken down in four major categories: security administration management, identification and authorization, access control, and data integrity/confidentiality/encryption. Under each category the requirements are listed in most critical to least critical order.

Assumptions

There are three general assumptions that follow throughout this document.

1. All loginids are unique; no two loginids can be the same. This prevents two users from having the same loginid.
2. The vendor should provide the requisite software to provide functionality on all supported platforms.
3. All vendor products are changing. All products will have to work with various unlike platforms.

Security Administration Management

Single Point of Administration. All administration of the product should be done from a single point. This enables an administrator to provide support for the product from any one platform device.

Ability to Group Users. The product should enable the grouping of like users where possible. These groups should be handled the same way individual users are handled. This will enable more efficient administration of access authority.

Ability to Enforce Enterprise/Global Security Rules. The product should provide the ability to enforce security rules over the entire enterprise,

regardless of platform. This will ensure consistent security over resources on all protected platforms.

Audit Trail. All changes, modifications, additions, and deletions should be logged. This ensures that all security changes are recorded for review at a later time.

Ability to Recreate. Information logged by the system should be able to be used to "backout" changes to the security system. Example: used to recreate deleted resources or users. This enables mass changes to be "backed out" of production or enables mass additions or changes to be made based on logged information.

Ability to Trace Access. The product should enable the administrator to be able to traced access to systems, regardless of system or platform.

Scoping and Decentralization of Control. The product should be able to support the creation of spans of control so that administrators can be excluded from or included in certain security control areas within the overall security setup. This enables an administrator to decentralize the administration of security functions based on the groups, nodes, domains, and enterprises over which the decentralized administrator has control.

Administration for Multiple Platforms. The product should provide for the administration of the product for any of the supported platforms. This enables the administrator to support the product for any platform of his or her choice.

Synchronization Across All Entities. The product should be synchronizing security data across all entities and all platforms. This ensures that all security decisions are made with up-to-date security information.

Real-Time and Batch Update. All changes should be made online/real-time. The ability to batch changes together is also important to enable easy loading or changing of large numbers of security resources or users.

Common Control Language Across All Platforms. The product should feature a common control language across all serviced platforms so that administrators do not have to learn and use different commands on different platforms.

One Single Product. The product should be a single product — not a compendium of several associated products. Modularity for the sake of platform-to-platform compatibility is acceptable and favored.

Flexible Cost. The cost of the product should be reasonable. Several cost scenarios should be considered, such as per seat, CPU, site licensing, and MIPS pricing. Pricing should include disaster recovery scenarios.

Physical Terminal/Node/Address Control. The product should have the ability to restrict or control access on the basis of a terminal, node, or network address. This ability will enable users to provide access control by physical location.

Release Independent/Backward Compatible. All releases of the product should be backward compatible or release independent. Features of new releases should coexist with current features and not require a total reinstallation of the product. This ensures that the time and effort previously invested in the prior release of the product is not lost when a new release is installed.

Software Release Distribution. New releases of the product should be distributed via the network from a single distribution server of the administrator's choice. This enables an administrator to upgrade the product on any platform without physically moving from platform to platform.

Ability to Do Phased Implementation. The product should support a phased implementation to enable administrators to implement the product on individual platforms without affecting other platforms. This will enable installation on a platform-by-platform basis if desired.

Ability to Interface with Application/Database/Network Security. The product should be able to interface with existing application, database, or network security by way of standard security interfaces. This will ensure that the product will mesh with security products already installed.

SQL Reporting. The product should have the ability to use SQL query and reporting tools to produce security setup reports/queries. This feature will enable easy access to security information for administrators.

Ability to Create Security Extract Files. The product should have a feature to produce an extract file of the security structure and the logging/violation records. This enables the administrator to write his or her own reporting systems via SAS or any other language.

Usage Counter per Application/Node/Domain/Enterprise. The product should include an internal counter to maintain the usage count of each application, domain, or enterprise. This enables an administrator to determine which applications, nodes, domains, or enterprises are being used and to what extent they are being used.

Test Facility. The product should include a test facility to enable administrators to test security changes before placing them into production. This ensures that all security changes are fully tested before being placed into production.

Ability to Tag Enterprise/Domain/Node/Application. The product should be able to add a notation or "tag" an enterprise/domain/node/application in order to provide the administrator with a way identify the entity. This enables the administrator to denote the tagged entity and possibly perform extra or nonstandard operations on the entity based on that tag.

Platform Inquiries. The product should support inquiries to the secured platforms regarding the security setup, violations, and other logged events. This will enable an administrator to inquire on security information without having to signon/logon.

Customize in Real-Time. It is important to have a feature that enables the customization of selected features (those features for which customization is allowed) without reinitializing the product. This feature will ensure that the product is available for 24-hour, seven-day-a-week processing.

GUI Interface. The product should provide a user interface via a Windows-like user interface. The interface may vary slightly between platforms (i.e., Windows, OS/2, X-windows, etc.) but should retain the same functionality. This facilitates operating consistency and lowers operator and user training requirements.

User Defined Fields. The product should have a number of user customizable/user-defined fields. This enables a user to provide for informational needs that are specific to his or her organization.

Identification and Authorization

Support RACF Pass Ticket Technology. The product should support IBM's RACF Pass Ticket technology, ensuring that the product can reside in an environment using Pass Ticket technology to provide security identification and authorization.

Support Password Rules (i.e. Aging, Syntax, etc.). All common password rules should be supported:

- use or non-use of passwords
- password length rules
- password aging rules
- password change intervals
- password syntax rules
- password expiration warning message
- Save previous passwords
- Password uniqueness rules
- Limited number of logons after a password expires
- Customer-defined rules

Logging of All Activity Including Origin/Destination/Application/Platform. All activity should be logged, or able to be logged, for all activities. The logging should include the origin of the logged item or action, the destination, the application involved, and the platform involved. This enables the administrator to provide a concise map of all activity on the enterprise. The degree of logging should be controlled by the administrator.

Single Revoke/Resume for All Platforms. The product should support a single revoke or resume of a loginid, regardless of the platform. This ensures that users can be revoked or resumed with only one command from one source or platform.

Support a Standard Primary loginid Format. The administrator should define all common loginid syntax rules. The product should include features to translate unlike loginids from different platforms so that they can be serviced. This enables the product to handle loginids from systems that support different loginid syntax that cannot be supported natively.

Auto Revoke after X Attempts. Users should be revoked from system access after a specified number of invalid attempts. This threshold should set by the administrator. This ensures that invalid users are prevented from retrying sign-ons indefinitely.

Capture Point of Origin Information, Including Caller ID/Phone Number for Dial-in Access. The product should be able to capture telephone caller ID (ANI) information if needed. This will provide an administrator increased information that can be acted upon manually or via an exit to provide increased security for chosen ports.

Authorization Server Should be Portable (Multi-platform). The product should provide for the authentication server to reside on any platform that the product can control. This provides needed portability if there is a need to move the authentication server to another platform for any reason.

Single Point of Authorization. All authorizations should be made a single point (i.e., an authentication server). The product should not need to go to several versions of the product on several platforms to gain the needed access to a resource. This provides not only a single point of administration for the product, but also reduced network security traffic.

Support User Exits/Options. The product should support the addition of user exits, options, or application programming interfaces (APIs) that could be attached to the base product at strategically identified points of operation. The points would include sign-on, sign-off, resource access check, etc. The enables an administrator or essential technical support

personnel to add exit/option code to the package to provide for specific security needs above and beyond the scope of the package.

Insure loginid Uniqueness. The product should ensure that all loginids are unique; no two loginids can be the same. This prevents two users from having the same loginid.

Source Sign-on Support. The product should support sign-ons from a variety of sources. These sources should include LAN/WAN, workstations, portables (laptops and notebooks), dial-in, and dumb terminals. This would ensure that all potential login sources are enabled to provide login capability, and facilitate support for legacy systems.

Customizable Messages. The product should support the use of customized security messages. The will enable an administrator to customize messages to fit the needs of his or her organization.

Access Control

Support Smart Card Tokens. The product should support the use of the common smart card security tokens (i.e., SecurID cards) to enable their use on any platform. The enables the administrator to provide for increased security measures where they are needed for access to the systems.

Ability to Support Scripting — Session Manager Menus. The product should support the use of session manager scripting. This enables the use of a session manager script in those sites and instances where they are needed or required.

Privileges at the Group and System Level. The product should support administration privileges at a group level (based on span of control) or on the system level. This enables the product to be administered by several administrators without the administrators' authority overlapping.

Default Protection Unless Specified. The product should provide for the protection of all resources and entities as the default unless the opposite of protection for only those resources profiled is specified. The enables each organization to determine the best way to install the product based on their own security needs.

Support Masking/Generics. The product should support security profiles containing generic characters that enable the product to make security decisions based on groups of resources as opposed to individual security profiles. The enables the administrator to provide security profiles over many like-named resources with the minimum amount of administration.

Allow Delegation Within Power of Authority. The product should allow an administrator to delegate security administration authority to others at the discretion of the administrator within his or her span of authority. An administrator would have the ability to give some of his or her security authority to another administrator for backup purposes.

Data Integrity/Confidentiality/Encryption

No Cleartext Passwords (Net or DB) — Dumb Terminal Exception. At no time should any password be available on the network or in the security database in clear, human-readable form. The only exception is the use of dumb terminals where the terminal does not support encryption techniques. This will ensure the integrity of the users' passwords in all cases with the exception of dumb terminals.

Option to Have One or Distributed Security DBs. The product should support the option of having a single security database or several distributed security databases on different platforms. This enables an administrator to use a distributed database on a platform that may be sensitive to increased activity rather than a single security database. The administrator will control who can and if they can update distributed databases.

Inactive User Time-out. All users who are inactive for a set period during a session should be timed out and signed off of all sessions. This ensures that a user who becomes inactive for whatever reason does not compromise the security of the system by providing an open terminal to a system. This feature should be controlled by the administrator and have two layers:

- at the session manager/screen level
- at the application/platform level

Inactive User Revoke. All users who have not signed on within a set period should be revoked. This period should be configurable by the administrator. This will ensure that loginids are not valid if not used within a set period of time.

Ability to Back Up Security DBs to Choice of Platforms/Media. The product should be able to back up its security database to a choice of supported platforms or storage media. This enables the user to have a variety of destinations available for the security database backup.

Encryption Should be Commercial Standard (Presently DES). The encryption used in the product should be standard. That standard is presently DES but could change as new encryption standards are made. This will ensure that the product will be based on a tested, generally accepted encryption base.

Integrity of Security DB(s). The database used by the product to store security information and parameters should be protected from changes via any source other than the product itself. Generic file edit tools should not be able to view or update the security database.

Optional Application Data Encryption. The product should provide the optional ability to interface to encrypted application data if the encryption techniques are provided. This enables the product to interact with encrypted data from exiting applications.

Failsoft Ability. The product should have the ability to perform at a degraded degree without access to the security database. This ability should rely on administrator input on an as needed basis to enable a user to sign-on, access resources, and sign-off. This enables the product to at least work in a degraded mode in an emergency in such a fashion that security is not compromised.

CONCLUSION

Single sign-on (SSO) can indeed be the answer to an array of user administration and access control problems. For the user, it *might* be a godsend. It is, however, not a straightforward or inexpensive solution. As with other so-called "enterprise security solutions," there remain the problems of scalability and phasing-in. There is generally no half-step to be taken in terms of how such a technology as this is rolled out. It is of course possible to limit it to a single platform, but that negates the whole point of doing SSO in the first place.

Like all solutions, SSO must have a real problem that it addresses. Initially regarded as a solution looking for a problem, SSO has broadened its scope to address more than simply the avalanche of loginids and passwords users seem to acquire in their systems travels. This greater functionality can provide much needed assistance and control in managing the user, his access rights, and the trail of activity left in his wake. This however comes at a cost.

Some significant observations made by others regarding SSO became apparent from an informal survey conducted by this author. The first is that it can be very expensive, based mostly on the scope of the implementation. The second is that it can be a solution looking for a problem; meaning that it sounds like a "really neat" technology (which it is) that proffers religion on some. This "religion" tends to be a real cause for concern in the manager or CIO over the IT function, for reasons that are well-understood. When the first conjoins with the second, the result is frequently substantial project scope creep — usually a very sad story with an unhappy ending in the IT world.

The third observation was more subtle, but more interesting. Although several vendors still offer an SSO product as an add-on, the trend appears to be more toward SSO slowly disappearing as a unique product. Instead, this capability is being included in platform or enterprise IT management solution software such as Tivoli (IBM) and Unicenter-TNG (Computer Associates). Given the fact that SSO products support most of the functions endemic to PKI, the other likelihood in the author's opinion is that SSO will be subsumed into the enterprise PKI solution, and thus become a "feature" rather than a "product."

It does seem certain that this technology will continue to mature and improve, and eventually become more widely used. As more and more experience is gained in implementation endeavors, the files of "lessons learned" will grow large with many painful implementation horror stories. Such stories often arise from "bad products badly constructed." Just as often, they arise from poorly managed implementation projects. SSO will suffer, and has, from the same bad rap — partially deserved, partially not. The point here is: do your homework, select the right tool for the right job, plan your work carefully, and execute thoroughly. It will probably still be difficult, but one might actually get the results one wants.

In the mystical and arcane practice of Information Security, many different tools and technologies have acquired that rarified and undeserved status known as "panacea." In virtually no case has any one of these fully lived up to this unreasonable expectation, and the family of products providing the function known as "single sign-on" is no exception.

Relational Data Base Access Controls Using SQL

Ravi S. Sandhu

This chapter discusses access controls in relational data base management systems. Access controls have been built into relational systems since they first emerged. Over the years, standards have developed and are continuing to evolve. In recent years, products incorporating mandatory controls for multilevel security have also started to appear.

The chapter begins with a review of the relational data model and SQL language. Traditional discretionary access controls provided in various dialects of SQL are then discussed. Limitations of these controls and the need for mandatory access controls are illustrated, and three architectures for building multilevel data bases are presented. The chapter concludes with a brief discussion of role-based access control as an emerging technique for providing better control than do traditional discretionary access controls, without the extreme rigidity of traditional mandatory access controls.

RELATIONAL DATA BASES

A relational data base stores data in relations that are expected to satisfy some simple mathematical properties. Roughly speaking, a relation can be thought of as a table. The columns of the table are called attributes, and the rows are called tuples. There is no significance to the order of the columns or rows; however, duplicate rows with identical values for all columns are not allowed.

Relation schemes must be distinguished from relation instances. The relation scheme gives the names of attributes as well as their permissible values. The set of permissible values for an attribute is said to be the attribute's domain. The relation instance gives the tuples of the relation at a given instant.

For example, the following is a relation scheme for the EMPLOYEE relation:

EMPLOYEE (NAME, DEPT, RANK, OFFICE, SALARY, SUPERVISOR)

The domain of the NAME, DEPT, RANK, OFFICE, and SUPERVISOR attributes are character strings, and the domain of the SALARY attribute is integers. A particular instance of the EMPLOYEE relation, reflecting the employees who are currently employed, is as follows:

NAME	DEPT	RANK	OFFICE	SALARY	SUPERVISOR
Rao	Electrical Engineering	Professor	KH252	50,000	Jones
Kaplan	Computer Science	Researcher	ST125	35,000	Brown
Brown	Computer Science	Professor	ST257	55,000	Black
Jones	Electrical Engineering	Chair	KH143	45,000	Black
Black	Administration	Dean	ST101	60,000	NULL

The relation instance of EMPLOYEE changes with the arrival of new employees, changes to data for existing employees, and with their departure. The relation scheme, however, remains fixed. The NULL value in place of Black's supervisor signifies that Black's supervisor has not been defined.

Primary Key

A candidate key for a relation is a minimal set of attributes on which all other attributes depend functionally. In other words, two tuples may not have the same values of the candidate key in a relation instance. A candidate key is minimal — no attribute can be discarded without destroying this property. A candidate key always exists, because, in the extreme case, it consists of all the attributes.

In general, there can be more than one candidate key for a relation. If, for example in the EMPLOYEE previously described, duplicate names can never occur, NAME is a candidate key. If there are no shared offices, OFFICE is another candidate key. In the particular relation instance above there are no duplicate salary values. This, however, does not mean that salary is a candidate key. Identification of the candidate key is a property of the relation scheme and applies to every possible instance, not merely to the one that happens to exist at a given moment. SALARY would qualify as a candidate key only in the unlikely event that the organization forbids duplicate salaries.

The primary key of a relation is one of its candidate keys that has been designated as such. In the previous example, NAME is probably more appropriate than OFFICE as the primary key. Realistically, a truly unique identifier, such as social security number or employee identity number, rather than NAME should be used as the primary key.

Entity and Referential Integrity

The primary key uniquely identifies a specific tuple from a relation instance. It also links relations together. The relational model incorporates two application-independent integrity rules called entity integrity and referential integrity to ensure these purposes are properly served.

Entity integrity simply requires that no tuple in a relation instance can have NULL (i.e., undefined) values for any of the primary key attributes. This property guarantees that the value of the primary key can uniquely identify each tuple.

Referential integrity involves references from one relation to another. This property can be understood in context of the EMPLOYEE relation by assuming that there is a second relation with the scheme:

DEPARTMENT (DEPT, LOCATION, PHONE NUMBER)

DEPT is the primary key of DEPARTMENT. The DEPT attribute of the EMPLOYEE relation is said to be a foreign key from the EMPLOYEE relation to the DEPARTMENT relation. In general, a foreign key is an attribute, or set of attributes, in one relation R_1, whose values must match those of the primary key of a tuple in some other relation R_2. R_1 and R_2 need not be distinct. In fact, because supervisors are employees, the SUPERVISOR attribute in EMPLOYEE is a foreign key with $R_1 = R_2 =$ EMPLOYEE.

Referential integrity stipulates that if a foreign key *FK* of relation R_1 is the primary key *PK* of R_2, then for every tuple in R_1 the value of *FK* must either be NULL or equal to the value of *PK* of a tuple in R_2. Referential integrity requires the following in the EMPLOYEE example:

- Because of the DEPT foreign key, there should be tuples for the Electrical Engineering, Computer Science and Administration departments in the DEPARTMENT relation.
- Because of the SUPERVISOR foreign key, there should be tuples for Jones, Brown and Black in the EMPLOYEE relation.

The purpose of referential integrity is to prevent employees from being assigned to departments or supervisors who do not exist in the data base, though it is all right for employee Black to have a NULL supervisor or for an employee to have a NULL department.

SQL

Every data base management system (DBMS) needs a language for defining, storing, retrieving, and manipulating data. SQL is the de facto standard in relational DBMSs. SQL emerged from several projects at the IBM San Jose (now called Almaden) Research Center in the mid-1970s. Its official name now is Data Base Language SQL.

An official standard for SQL has been approved by the American National Standards Institute (ANSI) and accepted by the International Standards Organization (ISO) and the National Institute of Standards and Technology as a Federal Information Processing Standard. The standard has evolved and continues to do so. The base standard is generally known as SQL'89 and refers to the 1989 ANSI standard. SQL'92 is an enhancement of SQL'89 and refers to the 1992 ANSI standard. A third version SQL, commonly known as SQL3, is being developed under the ANSI and ISO aegis.

Although most relational DBMSs support some dialect of SQL, SQL compliance does not guarantee portability of a data base from one DBMS to another. This is true because DBMS vendors typically include enhancements not required by the SQL standard but not prohibited by it either. Most products are also not completely compliant with the standard.

The following sections provide a brief explanation of SQL. Unless otherwise noted, the version discussed is SQL'89.

The CREATE Statement

The relation scheme for the EMPLOYEE example, is defined in SQL by the following command:

```
CREATE   TABLE        EMPLOYEE
         (NAME        CHARACTER      NOT NULL,
         DEPT         CHARACTER,
         RANK         CHARACTER,
         OFFICE       CHARACTER,
         SALARY       INTEGER,
         SUPERVISOR   CHARACTER,
         PRIMARY KEY  (NAME),
         FOREIGN KEY  (DEPT)         REFERENCES DEPARTMENT,
         FOREIGN KEY  (SUPERVISOR)   REFERENCES EMPLOYEE)
```

This statement creates a table called EMPLOYEE with six columns. The NAME, DEPT, RANK, OFFICE, and SUPERVISOR columns have character strings (of unspecified length) as values, whereas the SALARY column has integer values. NAME is the primary key. DEPT is a foreign key that references the primary key of table DEPARTMENT. SUPERVISOR is a foreign key that references the primary key (i.e., NAME) of the EMPLOYEE table itself.

INSERT and DELETE Statements

The EMPLOYEE table is initially empty. Tuples are inserted into it by means of the SQL INSERT statement. For example, the last tuple of the relation instance previously discussed is inserted by the following statement:

```
INSERT
INTO     EMPLOYEE(NAME, DEPT, RANK, OFFICE, SALARY, SUPERVISOR)
VALUES   VALUES('Black', 'Administration', 'Dean', 'ST101', 60000, NULL)
```

The remaining tuples can be similarly inserted. Insertion of the tuples for Brown and Jones must respectively precede insertion of the tuples for Kaplan and Rao, so as to maintain referential integrity. Alternatively, these tuples can be inserted in any order with NULL managers that are later updated to their actual values. There is a DELETE statement to delete tuples from a relation.

The SELECT Statement

Retrieval of data is effected in SQL by the SELECT statement. For example, the NAME, SALARY, and SUPERVISOR data for employees in the computer science department is extracted as follows:

```
SELECT    NAME, SALARY, SUPERVISOR
FROM      EMPLOYEE
WHERE     DEPT = 'Computer Science'
```

This query applied to instance of EMPLOYEE previously given returns the following data:

NAME	SALARY	SUPERVISOR
Kaplan	35,000	Brown
Brown	55,000	Black

The WHERE clause in a SELECT statement is optional. SQL also allows the retrieved records to be grouped together for statistical computations by means of built-in statistical functions. For example, the following query gives the average salary for employees in each department:

```
SELECT      DEPT, AVG(SALARY)
FROM        EMPLOYEE
GROUP BY    DEPT
```

Data from two or more relations can be retrieved and linked together in a SELECT statement. For example, the location of employees can be retrieved by linking the data in EMPLOYEE with that in DEPARTMENT, as follows:

```
SELECT    NAME, LOCATION
FROM      EMPLOYEE, DEPARTMENT
WHERE     EMPLOYEE.DEPT = DEPARTMENT.DEPT
```

This query attempts to match every tuple in EMPLOYEE with every tuple in DEPARTMENT but selects only those pairs for which the DEPT attribute in the EMPLOYEE tuple matches the DEPT attribute in the DEPARTMENT

tuple. Because DEPT is a common attribute to both relations, every use of it is explicitly identified as occurring with respect to one of the two relations. Queries involving two relations in this manner are known as joins.

The UPDATE Statement

Finally, the UPDATE statement allows one or more attributes of existing tuples in a relation to be modified. For example, the following statement gives all employees in the Computer Science department a raise of $1000:

```
UPDATE    EMPLOYEE
SET       SALARY = SALARY + 1000
WHERE     DEPT = 'Computer Science'
```

This statement selects those tuples in EMPLOYEE that have the value of Computer Science for the DEPT attribute. It then increases the value of the SALARY attribute for all these tuples by $1000 each.

BASE RELATIONS AND VIEWS

The concept of a view has an important security application in relational systems. A view is a virtual relation derived by an SQL definition from base relations and other views. The data base stores the view definitions and materializes the view as needed. In contrast, a base relation is actually stored in the data base.

For example, the EMPLOYEE relation previously discussed is a base relation. The following SQL statement defines a view called COMPUTER_SCI_DEPT:

```
CREATE    VIEW COMPUTER_SCI_DEPT
AS        SELECT    NAME, SALARY, SUPERVISOR
          FROM      EMPLOYEE
          WHERE     DEPT = 'Computer Science'
```

This defines the virtual relation as follows:

NAME	SALARY	SUPERVISOR
Kaplan	35,000	Brown
Brown	55,000	Black

A user who has permission to access COMPUTER_SCI_DEPT is thereby restricted to retrieving information about employees in the computer science department. The dynamic aspect of views can be illustrated by an example in which a new employee, Turing, is inserted in base relation EMPLOYEE, modifying it as follows:

NAME	DEPT	RANK	OFFICE	SALARY	SUPERVISOR
Rao	Electrical Engineering	Professor	KH252	50,000	Jones
Kaplan	Computer Science	Researcher	ST125	35,000	Brown
Brown	Computer Science	Professor	ST257	55,000	Black
Jones	Electrical Engineering	Chairman	KH143	45,000	Black
Black	Administration	Dean	ST101	60,000	NULL
Turing	Computer Science	Genius	ST444	95,000	Black

The view COMPUTER_SCI_DEPT is automatically modified to include Turing, as follows:

NAME	SALARY	SUPERVISOR
Kaplan	35,000	Brown
Brown	55,000	Black
Turing	95,000	Black

In general, views can be defined in terms of other base relations and views.

Views can also provide statistical information. For example, the following view gives the average salary for each department:

```
CREATE    VIEW AVSAL(DEPT,AVG)
AS        SELECT      DEPT,AVG(SALARY)
          FROM        EMPLOYEE
          GROUP BY    DEPT
```

For retrieval purposes, there is no distinction between views and base relations. Views, therefore, provide a very powerful mechanism for controlling what information can be retrieved. When updates are considered, views and base relations must be treated quite differently. In general, users cannot directly update views, particularly when they are constructed from the joining of two or more relations. Instead, the base relations must be updated, with views thus being updated indirectly. This fact limits the usefulness of views for authorizing update operations.

DISCRETIONARY ACCESS CONTROLS

This section describes the discretionary access control (DAC) facilities included in the SQL standard, though the standard is incomplete and does not address several important issues. Some of these deficiencies are being addressed in the evolving standard. Different vendors have also provided more comprehensive facilities than the standard calls for.

SQL Privileges

The creator of a relation in an SQL data base is its owner and can grant other users access to that relation. The access privileges or modes recognized in SQL correspond directly to the CREATE, INSERT, SELECT,

DELETE, and UPDATE SQL statements discussed previously. In addition, a REFERENCES privilege controls the establishment of foreign keys to a relation.

The CREATE Statement

SQL does not require explicit permission for a user to create a relation, unless the relation is defined to have a foreign key to another relation. In this case, the user must have the REFERENCES privilege for appropriate columns of the referenced relation. To create a view, a user must have the SELECT privilege on every relation mentioned in definition of the view. If a user has INSERT, DELETE, or UPDATE privileges on these relations, corresponding privileges will be obtained on the view (if it is updatable).

The GRANT Statement

The owner of a relation can grant one or more access privileges to another user. This can be done with or without the GRANT OPTION. If the owner grants SELECT with the GRANT OPTION, the user receiving this grant can further grant SELECT to other users. The latter GRANT can be done with or without the GRANT OPTION at the granting user's discretion.

The general format of a grant operation in SQL is as follows:

```
GRANT    privileges
[ON      relation]
TO       users
[WITH    GRANT OPTION]
```

The GRANT command applies to base relations as well as to views. The brackets on the ON and WITH clauses denotes that these are optional and may not be present in every GRANT command. It is not possible to grant a user the grant option on a privilege, without allowing the grant option itself to be further granted.

INSERT, DELETE, and SELECT privileges apply to the entire relation as a unit. Because INSERT and DELETE are operations on entire rows, this is appropriate. SELECT, however, implies the ability to select on all columns. Selection on a subset of the columns can be achieved by defining a suitable view and granting SELECT on the view. This method is somewhat awkward, and there have been proposals to allow SELECT to be granted on a subset of the columns of a relation. In general, the UPDATE privilege applies to a subset of the columns. For example, a user can be granted the authority to update the OFFICE but not the SALARY of an EMPLOYEE. SQL'92 extends the INSERT privilege to apply to a subset of the columns. Thus, a clerical user, for example, can insert a tuple for a new employee with the NAME, DEPARTMENT, and RANK data. The OFFICE, SALARY, and SUPERVISOR

data can then be updated in this tuple by a suitably authorized supervisory user.

SQL'89 has several omissions in its access control facilities. These omissions have been addressed by different vendors in different ways. The following section identifies the major omissions and illustrates how they have been addressed in products and in the evolving standard.

The REVOKE Statement

One major shortcoming of SQL'89 is the lack of a REVOKE statement to take away a privilege granted by a GRANT. IBM's DB2 product provides a REVOKE statement for this purpose.

It is often necessary that revocation cascade. In a cascading revoke, not only is the privilege revoked, so too are all GRANTs based on the revoked privilege. For example, if user Tom grants Dick SELECT on relation R with the GRANT OPTION, Dick subsequently grants Harry SELECT on R, and Tom revokes SELECT on R from Dick, the SELECT on R privilege is taken away not only from Dick but also from Harry. The precise mechanics of a cascading revoke is somewhat complicated. If Dick had received the SELECT on R privilege (with GRANT OPTION) not only from Tom but also from Jane before Dick granted SELECT to Harry, Tom's revocation of the SELECT from R privilege from Dick would not cause either Dick or Tom to lose this privilege. This is because the GRANT from Jane remains valid.

Cascading revocation is not always desirable. A user's privileges to a given table are often revoked because the user's job functions and responsibilities have changed. For example, if Mary, the head of a department moves on to a different assignment, her privileges to her former department's data should be revoked. However, a cascading revoke could cause lots of employees of that department to lose their privileges. These privileges must then be regranted to keep the department functioning.

SQL'92 allows a revocation to be cascading or not cascading, as specified by the revoker. This is a partial solution to the more general problem of how to reassign responsibility for managing access to data from one user to another as their job assignments change.

Other Privileges

Another major shortcoming of SQL'89 is the lack of control over who can create relations. In SQL'89, every user is authorized to create relations. The Oracle DBMS requires possession of a RESOURCE privilege to create new relations. SQL'89 does not include a privilege to DROP a relation. Such a privilege is included in DB2.

SQL'89 does not address the issue of how new users are enrolled in a data base. Several DBMS products take the approach that a data base is

originally created to have a single user, usually called the DBA (data base administrator). The DBA essentially has all privileges with respect to this data base and is responsible for enrolling users and creating relations. Some systems recognize a special privilege (called DBA in Oracle and DBADM in DB2) that can be granted to other users at the original DBA's discretion and allows these users effectively to act as the DBA.

LIMITATIONS OF DISCRETIONARY CONTROLS

The standard access controls of SQL are said to be discretionary because the granting of access is under user control. Discretionary controls have a fundamental weakness, however. Even when access to a relation is strictly controlled, a user with SELECT access can create a copy of the relation, thereby circumventing these controls. Furthermore, even if users can be trusted not to engage deliberately in such mischief, programs infected with Trojan horses can have the same disastrous effect.

For example, in the following GRANT operation:

TOM: GRANT SELECT ON EMPLOYEE TO DICK

Tom has not conferred the GRANT option on Dick. Tom's intention is that Dick should not be allowed to further grant SELECT access on EMPLOYEE to other users. However, this intent is easily subverted as follows. Dick creates a new relation, COPY-OF-EMPLOYEE, into which he copies all the rows of EMPLOYEE. As the creator of COPY-OF-EMPLOYEE, Dick can grant any privileges for it to any user. Dick can therefore grant Harry access to COPY-OF-EMPLOYEE as follows:

DICK: GRANT SELECT ON COPY-OF-EMPLOYEE TO HARRY

At this point, Harry has access to all the information in the original EMPLOYEE relation. For all practical purposes, Harry has SELECT access to EMPLOYEE, so long as Dick keeps COPY-OF-EMPLOYEE reasonably up to date with respect to EMPLOYEE.

The problem, however, is actually worse than this scenario indicates. It portrays Dick as a cooperative participant in this process. For example, it might be assumed that Dick is a trusted confidant of Tom and would not deliberately subvert Tom's intentions regarding the EMPLOYEE relation. But if Dick were to use a text editor supplied by Harry, which Harry had programmed to create the COPY-OF-EMPLOYEE relation and execute the preceding GRANT operation, the situation might be different. Such software is said to be a Trojan horse because in addition to the normal functions expected by its user it also engages in surreptitious actions to subvert security. Thus, a Trojan horse executed by Tom could actually grant Harry the privilege to SELECT on EMPLOYEE.

Organizations trying to avoid such scenarios can require that all software they run on relational data bases be free of Trojan horses, but this is generally not considered a practical option. The solution is to impose mandatory controls that cannot be violated, even by Trojan horses.

MANDATORY ACCESS CONTROLS

Mandatory access controls (MACs) are based on security labels associated with each data item and each user. A label on a data item is called a security classification; a label on a user is called security clearance. In a computer system, every program run by a user inherits the user's security clearance.

In general, security labels form a lattice structure. This discussion assumes the simplest situation, in which there are only two labels — S for secret and U for unclassified. It is forbidden for S information to flow into U data items. Two mandatory access controls rules achieve this objective:

1. *Simple security property.* A U-user cannot read S-data.
2. *Star property.* A S-user cannot write U-data.

Some important points should be clearly understood in this context. First, the rules assume that a human being with S clearance can log in to the system as a S-user or a U-user. Otherwise, the star property prevents top executives from writing publicly readable data. Second, these rules prevent only the overt reading and writing of data. Trojan horses can still leak secret data by using devious means of communication called covert channels. Finally, mandatory access controls in relational data bases usually enforce a strong star property:

- *Strong star property.* A S-user cannot write U-data, and a U-user cannot write S-data.

The strong star property limits users to writing at their own level, for reasons of integrity. The (weak) star property allows a U-user to write S-data. This can result in overwriting, and therefore destruction, of S-data by U-users. The remainder of this chapter will assume the strong star property.

Labeling Granularity

Security labels can be assigned to data at different levels of granularity in relational data bases. Assigning labels to entire relations can be useful but is generally inconvenient. For example, if some salaries are secret but others are not, these salaries must be placed in different relations. Assigning labels to an entire column of a relation is similarly inconvenient in the general case.

The finest granularity of labeling is at the level of individual attributes of each tuple or row or at the level of individual element-level labeling. This

offers considerable flexibility. Most of the products emerging offer labeling at the level of a tuple. Although not so flexible as element-level labeling, this approach is definitely more convenient than using relation- or column-level labels. Products in the short term can be expected to offer tuple-level labeling.

MULTILEVEL DATA BASE ARCHITECTURES

In a multilevel system, users and data with different security labels coexist. Multilevel systems are said to be trusted because they keep data with different labels separated and ensure the enforcement of the simple security and strong star properties. Over the past fifteen years or so, considerable research and development has been devoted to the construction of multilevel data bases. Three viable architectures are emerging:

1. Integrated data architecture (also known as the trusted subject architecture).
2. Fragmented data architecture (also known as the kernelized architecture).
3. Replicated data architecture (also known as the distributed architecture).

The newly emerging relational data base products are basically integrated data architectures. This approach requires considerable modification of existing relational DBMSs and can be supported by DBMS vendors because they own the source code for their DBMSs and can modify it in new products.

Fragmented and replicated architectures have been demonstrated in laboratory projects. They promise greater assurance of security than does the integrated data architecture. Moreover, they can be constructed by using commercial off-the-shelf DBMSs as components. Therefore, non-DBMS vendors can build these products by integrating off-the-shelf trusted operating systems and non-trusted DBMSs.

Integrated Data Architecture

The integrated data architecture is illustrated in Exhibit 1. The bottom of the Exhibit shows three kinds of data coexisting in the disk storage of the illustrated systems:

1. *U-non-DBMS-data.* Unclassified data files are managed directly by the trusted operating system.
2. *S-non-DBMS-data.* Secret data files are managed directly by the trusted operating system.
3. *U+S-DBMS-data.* Unclassified and secret data are stored in files managed cooperatively by the trusted operating system and the trusted DBMS.

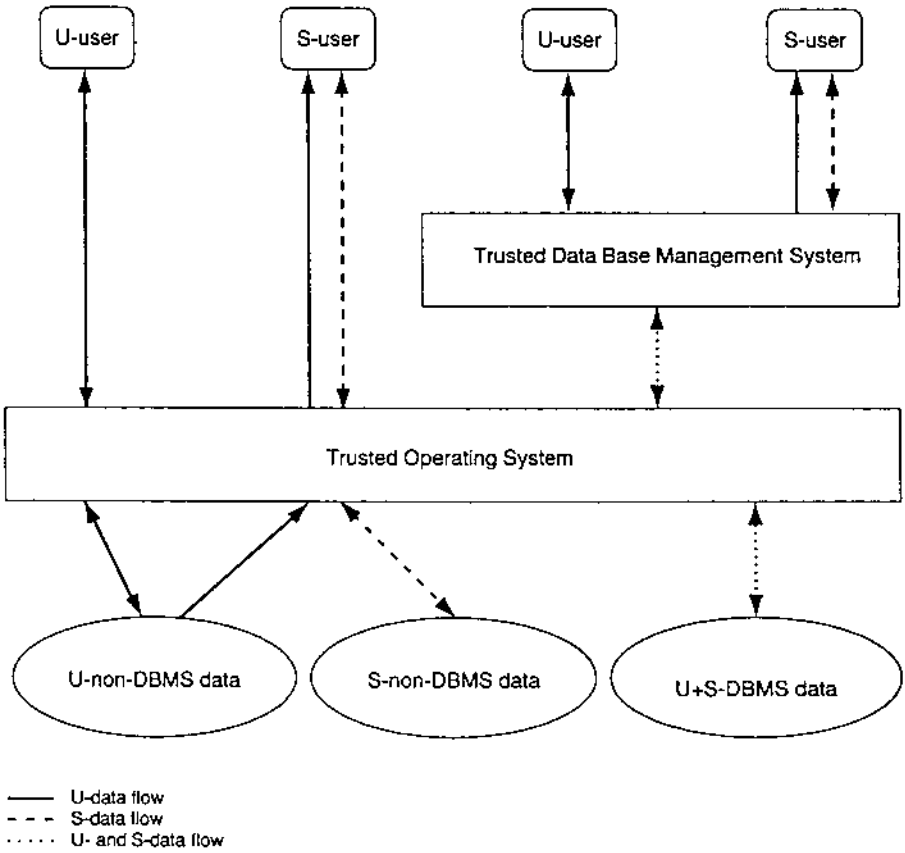


Exhibit 1. Integrated Data Architecture

At the top of the diagram on the left hand side a U-user and S-user interact directly with the trusted operating system. The trusted operating system allows these users to access only non-DBMS data in this manner. As according to the simple security and strong star properties, the U-user is allowed to read and write U-non-DBMS data, while the S-user is allowed to read U-non-DBMS data and read and write S-non-DBMS data. DBMS data must be accessed via the DBMS.

The right hand side of the diagram shows a U-user and S-user interacting with the trusted DBMS. The trusted DBMS enforces the simple security and strong star properties with respect to the DBMS data. The trusted DBMS relies on the trusted operating system to ensure that DBMS data cannot be accessed without intervention by the trusted DBMS.

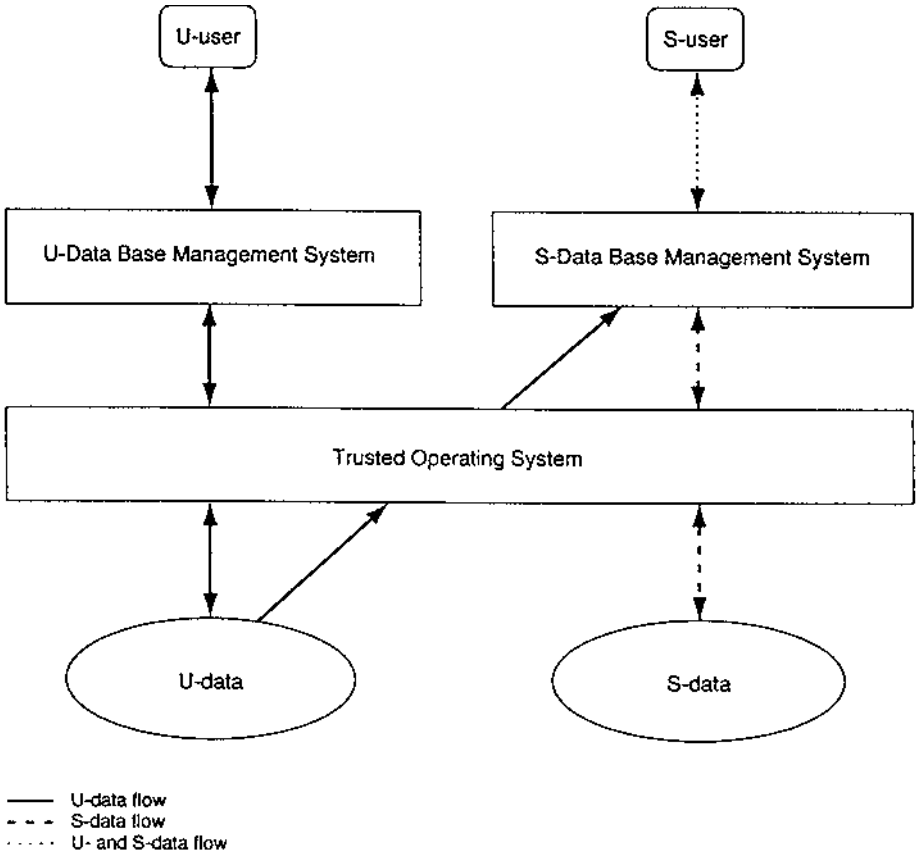


Exhibit 2. Fragmented Data Architecture

Fragmented Data Architecture

The fragmented data architecture is shown in Exhibit 2. In this architecture, only the operating system is multilevel and trusted. The DBMS is untrusted and interacts with users at a single level. The bottom of the exhibit shows two kinds of data coexisting in the disk storage of the system:

1. *U-data.* Unclassified data files are managed directly by the trusted operating system.
2. *S-data.* Secret data files are managed directly by the trusted operating system.

The trusted operating system does not distinguish between DBMS and non-DBMS data in this architecture. It supports two copies of the DBMS, one that can interact only with U-users and another that can interact only with S-users. These two copies run the same code but with different security

labels. The U-DBMS is restricted by the trusted operating system to reading and writing U-data. The S-DBMS, on other hand, can read and write S-data as well as read (but not write) U-data.

This architecture has great promise, but its viability depends on the availability of usable good-performance trusted operating systems. So far, there are few trusted operating systems, and these lack many of the facilities that users expect modern operating systems to provide. Development of trusted operating systems continues to be active, but progress has been slow. Emergency of strong products in this arena could make the fragmented data architecture attractive in the future.

Replicated Data Architecture

The replicated data architecture is shown in Exhibit 3. This architecture requires physical separation on backend data base servers to separate U- and S-users of the data base. The bottom half of the diagram shows two physically separated computers, each running a DBMS. The computer on the left hand side manages U-data, whereas the computer on the right hand side manages a mix of U- and S-data. The U-data on the left hand side is replicated on the right hand side.

The trusted operating system serves as a front end. It has two objectives. First, it must ensure that a U-user can directly access only the U-backend (left hand side) and that a S-user can directly access only the S-backend (right hand side). Second, the trusted operating system is the sole means for communication from the U-backend to the S-backend. This communication is necessary for updates to the U-data to be propagated to the U-data stored in the S-backend. Providing correct and secure propagation of these updates has been a major obstacle for this architecture, but recent research has provided solutions to this problem. The replicated architecture is viable for a small number of security labels, perhaps a few dozen, but it does not scale gracefully to hundreds or thousands of labels.

ROLE-BASED ACCESS CONTROLS

Traditional DACs are proving to be inadequate for the security needs of many organizations. At the same time, MACs based on security labels are inappropriate for many situations. In recent years, the notion of role-based access control (RBAC) has emerged as a candidate for filling the gap between traditional DAC and MAC.

One of weaknesses of DAC in SQL is that it does not facilitate the management of access rights. Each user must be explicitly granted every privilege necessary to accomplish his or her tasks. Often groups of users need similar or identical privileges. All supervisors in a department might require identical privileges; similarly, all clerks might require identical privileges,

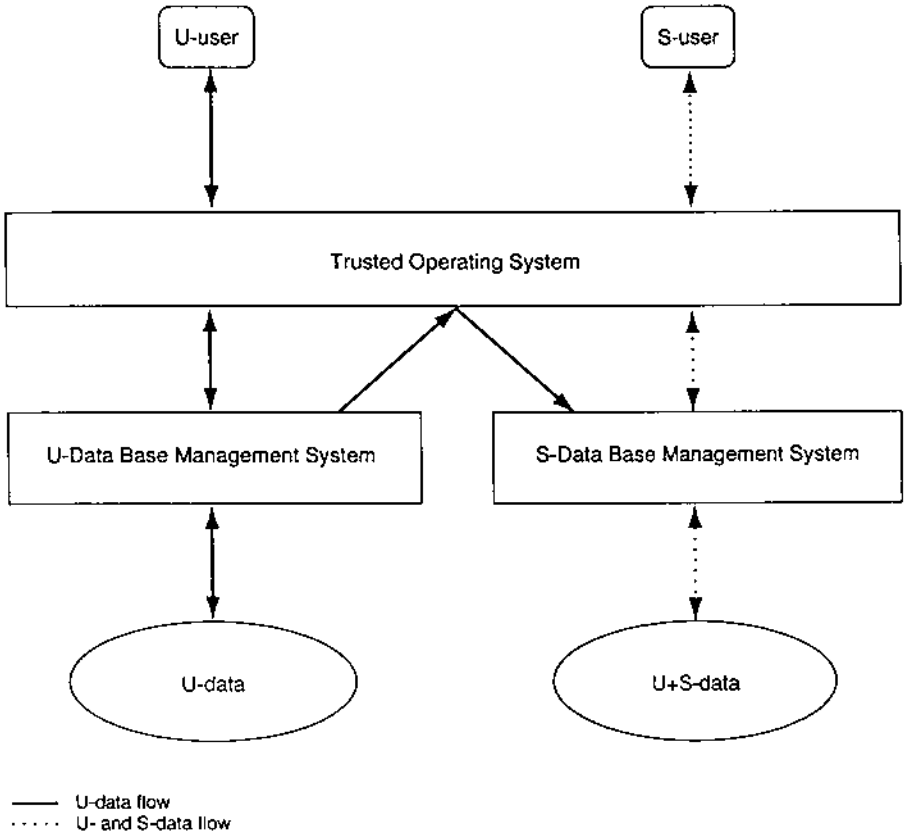


Exhibit 3. Replicated Data Architecture

different from those of the supervisors. RBAC allows the creation of roles for supervisors and clerks. Privileges appropriate to these roles are explicitly assigned to the role, and individual users are enrolled in appropriate roles from where they inherit these privileges. This arrangement separates two concerns: (1) what privileges should a role get and (2) which user should be authorized to each role. RBAC eases the task of reassigning users from one role to another or altering the privileges for an existing role.

Current efforts at evolving SQL, commonly called SQL3, have included proposals for RBAC based on vendor implementations, such as in Oracle. In the future, consensus on a standard approach to RBAC in relational data bases should emerge. However, this is a relatively new area, and a number of questions remain to be addressed before consensus on standards is obtained.

SUMMARY

Access controls have been an integral part of relational data base management systems from their introduction. There are, however, major weaknesses in the traditional discretionary access controls built into the standards and products. SQL'89 is incomplete and omits revocation of privileges and control over creation of new relations and views. SQL'92 fixes some of these shortcomings. In the meantime such vendors as Oracle have developed RBAC; other vendors, such as Informix, have started delivering products incorporating mandatory access controls for multilevel security. There is a recognition that SQL needs to evolve to take some of these developments into consideration. If it does, stronger and better access controls can be expected in future products.

8

Centralized Authentication Services (RADIUS, TACACS, DIAMETER)

Bill Stackpole, CIS

Got the telecommuter, mobile workforce, VPN, **multi-platform, dial-in user authentication blues?** Need a centralized method for controlling and auditing external accesses to your network? Then RADIUS, TACACS, or DIAMETER may be just what you have been looking for. Flexible, inexpensive, and easy to implement, these centralized authentication servers improve remote access security and reduce the time and effort required to manage remote access server (RAS) clients.

RADIUS, TACACS, and DIAMETER are classified as authentication, authorization, and accounting (AAA) servers. The Internet Engineering Task Force (IETF) chartered an AAA Working Group in 1998 to develop the authentication, authorization, and accounting requirements for network access. The goal was to produce a base protocol that supported a number of different network access models, including traditional dial-in network access servers (NAS), Mobile-IP, and roaming operations (ROAMOPS). The group was to build upon the work of existing access providers such as Livingston Enterprises.

Livingston Enterprises originally developed RADIUS (Remote Authentication Dial-In User Service) for their line of network access servers (NAS) to assist timeshare and Internet service providers with billing information consolidation and connection configuration. Livingston based RADIUS on the IETF distributed security model and actively promoted it through the IETF Network Access Server Requirements Working Group in the early 1990s. The client/server design was created to be open and extensible so it could be easily adapted to work with other third-party products. At this writing, RADIUS version 2 was a proposed IETF standard managed by the RADIUS Working Group.

The origin of the Terminal Access Controller Access Control System (TACACS) daemon used in the early days of ARPANET is unknown. Cisco Systems adopted the protocol to support AAA services on its products in the early 1990s. Cisco extended the protocol to enhance security and support additional types of authentication requests and response codes. They named the new protocol TACACS+. The current version of the TACACS specification is a proposed IETF Standard (RFC 1492) managed by the Network Working Group. It was developed with the assistance of Cisco Systems.

Pat Calhoun (Sun Laboratories) and Allan Rubens (Ascend Communications) proposed the DIAMETER AAA framework as a draft standard to the IETF in 1998. The name DIAMETER is not an acronym but rather a play on the RADIUS name. DIAMETER was designed from the ground up to support roaming applications and to overcoming the extension limitations of the RADIUS and TACACS protocols. It provides the base protocols required to support any number of AAA extensions, including NAS, Mobile-IP, host, application, and Web-based requirements. At this writing, DIAMETER consisted of eight IETF draft proposals, authored by twelve different contributors from Sun, Microsoft, Cisco, Nortel, and others. Pat Calhoun continues to coordinate the DIAMETER effort.

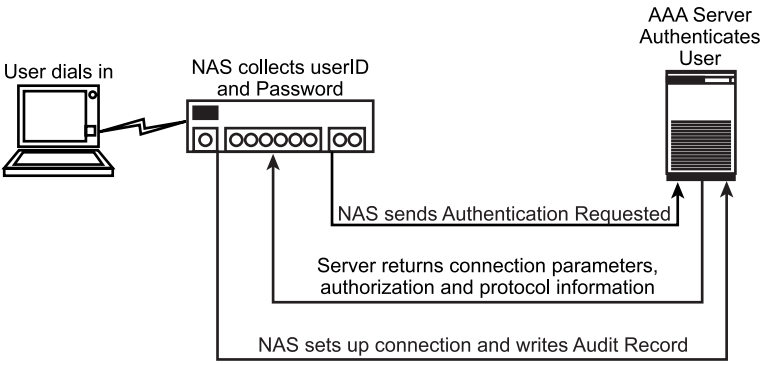


EXHIBIT 8.1 Key features of a centralized AAA service.

AAA 101: Key Features of an AAA Service

The key features of a centralized AAA service include (1) a distributed (client/server) security model, (2) authenticated transactions, (3) flexible authentication mechanisms, and (4) an extensible protocol. Distributed security separates the authentication process from the communications process, making it possible to consolidate user authentication information into a single centralized database. The network access devices (i.e., an NAS) are the clients. They pass user information to an AAA server and act upon the response(s) the server returns. The servers receive user connection requests, authenticate the user, and return to the client NAS the configuration information required to deliver services to the user. The returned information may include transport and protocol parameters, additional authentication requirements (i.e., callback, SecureID), authorization directives (i.e., services allowed, filters to apply), and accounting requirements (Exhibit 8.1).

Transmissions between the client and server are authenticated to ensure the integrity of the transactions. Sensitive information (e.g., passwords) is encrypted using a shared secret key to ensure confidentiality and prevent passwords and other authentication information from being monitored or captured during transmission. This is particularly important when the data travels across public carrier (e.g., WAN) links.

AAA servers can support a variety of authentication mechanisms. This flexibility is a key AAA feature. User access can be authenticated using PAP (Password Authentication Protocol), CHAP (Challenge Handshake Authentication Protocol), the standard UNIX login process, or the server can act as a proxy and forward the authentication to other mechanisms like a Microsoft domain controller, a Novell NDS server, or a SecureID ACE server. Some AAA server implementations use additional mechanisms such as calling number identification (caller ID) and callback to further secure connections.

Because technology changes so rapidly, AAA servers are designed with extensible protocols. RADIUS, DIAMETER, and TACACS use variable-length attribute values designed to support any number of new parameters without disturbing existing implementations of the protocol. DIAMETER's framework approach provides additional extensibility by standardizing a transport mechanism (framework) that can support any number of customized AAA modules.

From a management perspective, AAA servers provide some significant advantages, including:

- Reduced user setup and maintenance times because users are maintained on a single host
- Fewer configuration errors because formats are similar across multiple access devices
- Less security administrator training requirements because there is only one system syntax to learn
- Better auditing because all login and authentication requests come through a single system
- Reduced help desk calls because the user interface is consistent across all access methods
- Quicker proliferation of access information because information only needs to be replicated to a limited number of AAA servers
- Enhanced security support through the use of additional authentication mechanisms (i.e., SecureID)
- Extensible design makes it easy to add new devices without disturbing existing configurations

RADIUS: Remote Authentication Dial-In User Service

RADIUS is by far the most popular AAA service in use today. Its popularity can be attributed to Livingston's decision to open the distribution of the RADIUS source code. Users were quick to port the service across multiple platforms and add customized features, many of which Livingston incorporated as standard features in later releases. Today, versions of the RADIUS server are available for every major operating system from both freeware and commercial sources, and the RADIUS client comes standard on NAS products from every major vendor.

A basic RADIUS server implementation references two configuration files. The client configuration file contains the address of the client and the shared secret used to authenticate transactions. The user file contains the user identification and authentication information (e.g., userID and password) as well as connection and authorization parameters. Parameters are passed between the client and server using a simple five-field format encapsulated into a single UDP packet. The brevity of the format and the efficiency of the UDP protocol (no connection overhead) allow the server to handle large volumes of requests efficiently. However, the format and protocol also have a downside. They do not lend themselves well to some of today's diverse access requirements (i.e., ROAMOPS), and retransmissions are a problem in heavy load or failed node scenarios.

Putting the AA in RADIUS: Authentications and Authorizations

RADIUS has eight standard transaction types: access-request, access-accept, access-reject, accounting-request, accounting-response, access-challenge, status-server, and status-client. Authentication is accomplished by decrypting a NAS access-request packet, authenticating the NAS source, and validating the access-request parameters against the user file. The server then returns one of three authentication responses: access-accept, access-reject, or access-challenge. The latter is a request for additional authentication information such as a one-time password from a token or a callback identifier.

Authorization is not a separate function in the RADIUS protocol but simply part of an authentication reply. When a RADIUS server validates an access request, it returns to the NAS client all the connection attributes specified in the user file. These usually include the data link (i.e., PPP, SLIP) and network (i.e., TCP/IP, IPX) specifications, but may also include vendor-specific authorization parameters. One such mechanism automatically initiates a Telnet or rlogin session to a specified host. Other methods include forcing the port to a specific IP address with limited connectivity, or applying a routing filter to the access port.

The Third A: Well, Sometimes Anyway!

Accounting is a separate function in RADIUS and not all clients implement it. If the NAS client is configured to use RADIUS accounting, it will generate an Accounting-Start packet once the user has been authenticated, and an Accounting-Stop packet when the user disconnects. The Accounting-Start packet describes the type of service the NAS is delivering, the port being used, and user being serviced. The Accounting-Stop packet duplicates the Start packet information and adds session information such as elapsed time, bytes inputs and outputs, disconnect reason, etc.

Forward Thinking and Other Gee-Whiz Capabilities

A RADIUS server can act as a proxy for client requests, forwarding them to servers in other authentication domains. Forwarding can be based on a number of criteria, including a named or number domain. This is particularly useful when a single modem pool is shared across departments or organizations. Entities are not required to share authentication data; each can maintain its own RADIUS server and service proxied requests from the server at the modem pool. RADIUS can proxy both authentication and accounting requests. The relationship between proxies can be distributed (one-to-many) or hierarchical (many-to-one), and requests can be forwarded multiple times. For example, in Exhibit 8.2, it is perfectly permissible for the "master" server to forward a request to the user's regional server for processing.

Most RADIUS clients have the ability to query a secondary RADIUS server for redundancy purposes, although this is not required. The advantage is continued access when the primary server is offline. The disadvantage is the increase in administration required to synchronize data between the servers.

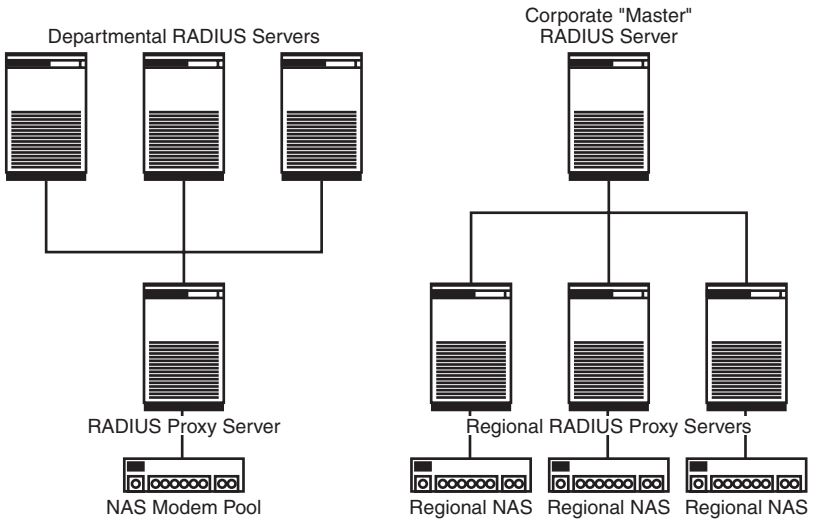


EXHIBIT 8.2 "Master" server forwards a request on to the user's regional server for processing.

Most RADIUS servers have a built-in database connectivity component. This allows accounting records to be written directly into a database for billing and reporting purposes. This is preferable to processing a flat text accounting "detail" file. Some server implementations also include database access for authentication purposes. Novell's implementation queries NDS, NT versions query the PDC, and several vendors are working on LDAP connectivity.

It Does Not Get Any Easier than This. Or Does It?

When implementing RADIUS, it is important to remember that the source code is both open and extensible. The way each AAA, proxy, and database function is implemented varies considerably from vendor to vendor. When planning a RADIUS implementation, it is best to define one's functional requirements first and then choose NAS components and server software that support them. Here are a few factors to consider:

- *What accesses need to be authenticated?* External accesses via modem pools and VPN servers are essential, but internal accesses to critical systems and security control devices (i.e., routers, firewalls) should also be considered.
- *What protocols need to be supported?* RADIUS can return configuration information at the data-link, network, and transport levels. Vendor documentation as well as the RADIUS RFCs and standard dictionary file are good sources of information for evaluating these parameters.
- *What services are required?* Some RADIUS implementations require support for services such as Telnet, rlogin, and third-party authentication (i.e., SecureID), which often require additional components and expertise to implement.
- *Is proxy or redundancy required?* When NAS devices are shared across management or security domains, proxy servers are usually required and it is necessary to determine the proxy relationships in advance. Redundancy for system reliability and accessibility is also an important consideration because not all clients implement this feature.

Other considerations might include:

- Authorization, accounting, and database access requirements
- Interfaces to authentication information in NDS, X.500, or PDC databases
- The RADIUS capabilities of existing clients
- Support for third-party Mobile-IP providers like iPass
- Secure connection support (i.e., L2TP, PPTP)

Client setup for RADIUS is straightforward. The client must be configured with the IP address of the server(s), the shared secret (encryption key), and the IP port numbers of the authentication and accounting services (the defaults are 1645 and 1646, respectively). Additional settings may be required by the vendor.

The RADIUS server setup consists of the server software installation and three configuration files:

1. The dictionary file is composed of a series of Attribute/Value pairs the server uses to parse requests and generate responses. The standard dictionary file supplied with most server software contains the attributes and values found in the RADIUS RFCs. One may need to add vendor-specific attributes, depending upon one's NAS selection. If any modifications are made, double-check that none of the attribute Names or Values are duplicated.
2. The client file is a flat text file containing the information the server requires to authenticate RADIUS clients. The format is the client name or IP address, followed by the shared secret. If names are used, the server must be configured for name resolution (i.e., DNS). Requirements for the length and format of the shared secret vary, but most UNIX implementations are eight characters or less. There is no limitation on the number of clients a server can support.
3. The user file is also a flat text file. It stores authentication and authorization information for all RADIUS users. To be authenticated, a user must have a profile consisting of three parts: the *username*, a list of authentication *check items,* and a list of *reply items.* A typical entry would look like the one displayed in Exhibit 8.3. The first line contains the user's name and a list of check items separated by commas. In this example, John is restricted to using one NAS device (the one at 10.100.1.1). The remaining lines contain reply items. Reply items are separated by commas at the end of each line. String values are put in quotes. The final line in this example contains an authorization parameter that applies a packet filter to this user's access.

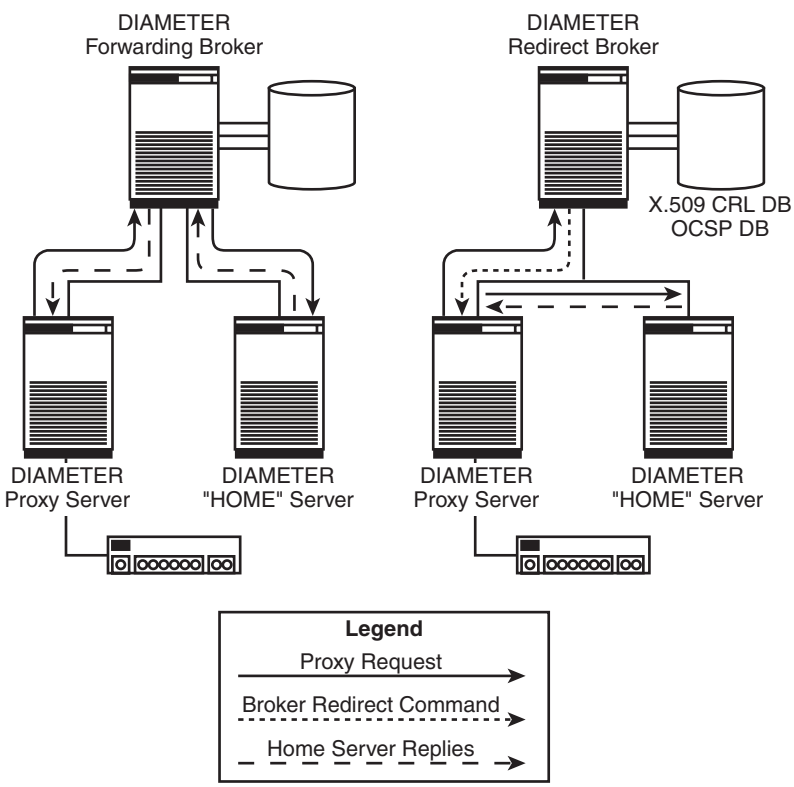


EXHIBIT 8.3 DIAMETER uses a broker proxy server.

The check and reply items contained in the user file are as diverse as the implementations, but a couple of conventions are fairly common. Username prefixes are commonly used for proxy requests. For example, usernames with the prefix CS/ would be forwarded to the computer science RADIUS server for authentication. Username suffixes are commonly used to designate different access types. For example, a user name with a %vpn suffix would indicate that this access was via a virtual private network (VPN). This makes it possible for a single RADIUS server to authenticate users for multiple NAS devices or provide different reply values for different types of accesses on the same NAS.

The DEFAULT user parameter is commonly used to pass authentication to another process. If the username is not found in the user file, the DEFAULT user parameters are used to transfer the validation to another mechanism. On UNIX, this is typically the `/etc/passwd` file. On NT, it can be the local user database or a domain controller. Using secondary authentication mechanisms has the advantage of expanding the check items RADIUS can use. For example, UNIX and NT groups can be checked as well as account activation and date and time restriction.

Implementations that use a common NAS type or one server for each NAS type have fairly uncomplicated user files, but user file contents can quickly become quite convoluted when NAS devices and access methods are mixed. This not only adds complexity to the management of the server, but also requires more sophistication on the part of users.

Stumbling Blocks, Complexities, and Other RADIUS Limitations

RADIUS works well for remote access authentication but is not suitable for host or application authentication. Web servers may be the first exception. Adding a RADIUS client to a Web server provides a secure method for authenticating users across open networks. RADIUS provides only basic accounting facilities with no facilities for monitoring nailed-up circuits or system events. User-based rather than device-based connection parameters are another major limitation of RADIUS. When a single RADIUS server manages several different types of NAS devices, user administration is considerably more complex. Standard RADIUS authentication does not provide facilities for checking a user's group membership, restricting access by date or time of day, or expiring a user's account on a given date. To provide these capabilities, the RADIUS server must be associated with a secondary authentication service.

Overall, RADIUS is an efficient, flexible, and well-supported AAA service that works best when associated with a secondary authentication service like NDS or NT where additional account restrictions can be applied. The adoption of RADIUS version 2 as an IETF standard will certainly ensure its continued success and importance as a good, general-purpose authentication, authorization, and accounting service.

TACACS: Terminal Access Controller Access Control System

What is commonly referred to today as TACACS actually represents two evolutions of the protocol. The original TACACS, developed in the early ARPANet days, had very limited functionality and used the UDP transport. In the early 1990s, the protocol was extended to include additional functionality and the transport changed to TCP. To maintain backward compatibility, the original functions were included as subsets of the extended functions. The new protocol was dubbed XTACACS (Extended TACACS). Virtually all current TACACS daemons are based on the extended protocol as described in RFC1492.

Cisco Systems adopted TACACS for its AAA architecture and further enhanced the product by separating the authentication, authorization, and accounting functions and adding encryption to all NAS-server transmissions. Cisco also improved the extensibility of TACACS by permitting arbitrary length and content parameters for authentication exchanges. Cisco called its version TACACS+ but, in reality, TACACS+ bares no resemblance to the original TACACS and packet formats are not backward compatible. Some server implementations support both formats for compatibility purposes. The remainder of this section is based on TACACS+ because it is the proposed IETF standard.

TACACS+ servers use a single configuration file to control server options, define users and attribute/value (AV) pairs, and control authentication and authorization actions. The options section specifies the settings of the service's operation parameters, the shared secret key, and the accounting file name. The remainder of the file is a series of user and group definitions used to control authentication and authorization actions. The format is "*user = username*" or "*group = groupname,*" followed by one or more AV pairs inside curly brackets.

The client initiates a TCP session and passes a series of AV pairs to the server using a standard header format followed by a variable length parameter field. The header contains the service request type (authentication, authorization, or accounting) and is sent in the clear. The entire parameter field is encrypted for confidentiality. TACACS' variable parameter field provides for extensibility and site-specific customization, while the TCP protocol ensures reliable delivery. However, the format and protocol also increase communications overhead, which can impact the server's performance under heavy load.

A 1: TACACS Authentication

TACACS authentication has three packet types: Start, Continue, and Reply. The client begins the authentication with a Start packet that describes the type of authentication to be performed. For simple authentication types such as PAP, the packet may also contain the userID and password. The server responds with a Reply. Additional information, if required, is passed with client Continue and server Reply packets. Transactions include login (by privilege level) and password change using various authentication protocols (i.e., CHAP, PAP, PPP, etc.). Like RADIUS, a successful TACACS authentication returns attribute-value (AV) pairs for connection configuration. These can include authorization parameters or they can be fetched separately.

A 2: TACACS Authorization

Authorization functions in TACACS consist of Request and Response AV pairs used to:

- Permit or deny certain commands, addresses, services or protocols
- Set user privilege level
- Invoke input and output packet filters
- Set access control lists (ACLs)
- Invoke callback actions
- Assign a specific network address

Functions can be returned as part of an authentication transaction or an authorization-specific request.

A 3: TACACS Accounting

TACACS accounting functions use a format similar to authorization functions. Accounting functions include Start, Stop, More, and Watchdog. The Watchdog function is used to validate TCP sessions when data is not sent for extended periods of time. In addition to the standard accounting data supported by RADIUS, TACACS has an event logging capability that can record system level changes in access rights or privilege. The reason for the event as well as the traffic totals associated with it can also be logged.

Take Another Look (and Other Cool Capabilities)

TACACS authentication and authorization processes are considerably enhanced by two special capabilities: recursive lookup and callout. Recursive lookup allows connection, authentication, and authorization information to be spread across multiple entries. AV pairs are first looked up in the user entry. Unresolved pairs are then looked up in the group entry (if the user is a member of a group) and finally assigned the default value (if one is specified). TACACS+ permits groups to be embedded in other groups, so recursive lookups can be configured to encompass any number of connection requirements. TACACS+ also supports a callout capability that permits the execution of user-supplied programs. Callout can be used to dynamically alter the authentication and authorization processes to accommodate any number of requirements — a considerably more versatile approach than RADIUS' static configurations. Callout can be used to interface TACACS+ with third-party authentication mechanisms (i.e., Kerberos and SecureID), pull parameters from a directory or database, or write audit and accounting records.

TACACS, like RADIUS, can be configured to use redundant servers and because TACACS uses a reliable transport (TCP); it also has the ability to detect failed nodes. Unlike RADIUS, TACACS cannot be configured to proxy NAS requests, which limits its usefulness in large-scale and cross-domain applications.

Cisco, Cisco, Cisco: Implementing TACACS

There are a number of TACACS server implementations available, including two freeware versions for UNIX, a Netware port, and two commercial versions for NT, but the client implementations are Cisco, Cisco, Cisco. Cisco freely distributes the TACACS and TACACS+ source code, so features and functionality vary considerably from one implementation to another. CiscoSecure is generally considered the most robust of the commercial implementations and even supports RADIUS functions. Once again, be sure to define functional requirements before selecting NAS components and server software. If your shop is Cisco-centric, TACACS is going to work well; if not, one might want to consider a server product with both RADIUS and TACACS+ capabilities.

Client setup for TACACS on Cisco devices requires an understanding of Cisco's AAA implementation. The AAA function must be enabled for any of the TACACS configuration commands to work. The client must be configured with the IP address of the server(s) and the shared secret encryption key. A typical configuration would look like this:

```
aaa new-model
tacacs-server key <your key here>
tacacs-server host <your primary TACACS server
IP address here >
tacacs-server host <your secondary TACACS server
IP address here >
```

followed by port-specific configurations. Different versions of Cisco IOS support different TACACS settings. Other NAS vendors support a limited subset of TACACS+ commands.

TACACS server setup consists of the server software installation and editing the options, authentication, and authorization entries in the configuration files. Comments may be placed anywhere in the file using a pound sign (#) to start the line. In the following example, Jane represents a dial-in support contractor, Bill a user with multiple access methods, and Dick an IT staff member with special NAS access.

```
# The default authentication method will use the
local UNIX
# password file, default authorization will be
permitted for
# users without explicit entries and accounting
records will be
# written to the /var/adm/tacacs file.
default authentication = file /etc/passwd
default authorization = permit
     accounting file = /var/adm/tacacs
# Contractors, vendors, etc.
user = jane {
name = "Jane Smith"
global = cleartext "Jane'sPassword"
expires = "May 10 2000"
service=ppp
protocol=ip {
  addr=10.200.10.64
  inacl=101
  outacl=102
}
}
# Employees with "special" requirements
user = bill {
```

```
name="Bill Jones"
arap = cleartext "Apple_ARAP_Password"
pap = cleartext "PC_PAP_Password"
default service = permit
     }
user = dick {
name="Dick Brown"
member = itstaff
# Use the service parameters from the default user
default service = permit
# Permit Dick to access the exec command using
connection access list 4
service = exec {
  acl = 4
}
# Permit Dick to use the telnet command
to everywhere but 10.101.10.1
cmd = telnet {
  deny 10\.101\.10\.1
  permit .*
}
     }
# Standard Employees use these entries
user = DEFAULT {
service = ppp {
  # Disconnect if idle for 5 minutes
  idletime = 5
  # Set maximum connect time to one hour
  timeout = 60
}
protocol = ip {
  addr-pool=hqnas
}
     }
# Group Entries
group = itstaff {
# Staff uses a special password file
login = file /etc/itstaff_passwds
     }
```

Jane's entry sets her password to "Jane'sPassword" for all authentication types, requires her to use PPP, forces her to a known IP, and applies both inbound and outbound extended IP access control lists (a.k.a. IP filters). It also contains an account expiration date so the account can be easily enabled and disabled. Bill's entry establishes different passwords for Apple and PAP logins, and assigns his connection the default service parameters. Dick's entry grants him access to the NAS executive commands, including Telnet, but restricts their use by applying a standard IP access control list and an explicit **deny** to the host at 10.101.10.1. Bill and Dick's entries also demonstrate TACACS' recursive lookup feature. The server first looks at user entry for a

password, then checks for a group entry. Bill is not a member of any group, so the default authentication method is applied. Dick, however, is a member of "itstaff," so the server validates the group name and looks for a password in the group entry. It finds the **login** entry and authenticates Dick using the /etc/itstaff_passwds file. The default user entry contains AV pairs specifying the use of PPP with an idle timeout of five minutes and a maximum session time of one hour.

In this example, the UNIX /etc/password and /etc/group files are used for authentication, but the use of other mechanisms is possible. Novell implementations use NDS, NT versions use the domain controller, and CiscoSecure support LDAP and several SQL-compatible databases.

Proxyless, Problems, and Pitfalls: TACACS Limitations

The principle limitation of TACACS+ may well be its lack of use. While TACACS+ is a versatile and robust protocol, it has few server implementations and even fewer NAS implementations. Outside of Cisco, this author was unable to find any custom extensions to the protocol or any vendor-specific AV pairs. Additionally, TACACS' scalability and performance are an issue. Unlike RADIUS' single-packet UDP design, TACACS uses multiple queries over TCP to establish connections, thus incurring overhead that can severely impact performance. TACACS+ servers have no ability to proxy requests so they cannot be configured in a hierarchy to support authentication across multiple domains. CiscoSecure scalability relies on regional servers and database replication to scale across multiple domains. While viable, the approach assumes a single management domain, which may not always be the case.

Overall, TACACS+ is a reliable and highly extensible protocol with existing support for Cisco's implementation of NAS-based VPNs. Its "outcalls" capability provides a fairly straightforward way to customize the AAA functions and add support for third-party products. Although TACACS+ supports more authentication parameters than RADIUS, it still works best when associated with a secondary authentication service like NDS or an NT domain. The adoption of TACACS+ as an IETF standard and its easy extensibility should improve its adoption by other NAS manufactures. Until then, TACACS+ remains a solid AAA solution for Cisco-centric environments.

DIAMETER: Twice RADIUS?

DIAMETER is a highly extensible AAA framework capable of supporting any number of authentication, authorization, or accounting schemes and connection types. The protocol is divided into two distinct parts: the Base Protocol and the Extensions. The DIAMETER Base Protocol defines the message format, transport, error reporting, and security services used by all DIAMETER extensions. DIAMETER Extensions are modules designed to conduct specific types of authentication, authorization, or accounting transactions (i.e., NAS, Mobile-IP, ROAMOPS, and EAP). The current IETF draft contains definitions for NAS requests, Mobile-IP, secure proxy, strong security, and accounting, but any number of other extensions are possible.

DIAMETER is built upon the RADIUS protocol but has been augmented to overcome inherent RADIUS limitations. Although the two protocols do not share a common data unit (PDU), there are sufficient similarities to make the migration from RADIUS to DIAMETER easier. DIAMETER, like RADIUS, uses a UDP transport but in a peer-to-peer rather than client/server configuration. This allows servers to initiate requests and handle transmission errors locally. DIAMETER uses reliable transport extensions to reduce retransmissions, improve failed node detection, and reduce node congestion. These enhancements reduce latency and significantly improve server performance in high-density NAS and hierarchical proxy configurations. Additional improvements include:

- Full support for roaming
- Cross-domain, broker-based authentication
- Full support for the Extensible Authentication Protocol (EAP)
- Vendor-defined attributes-value pairs (AVPs) and commands
- Enhanced security functionality with replay attack protections and confidentiality for individual AVPs

EXHIBIT 8.4 DIAMETER Base Protocol Packet Format

Type – Flags – Version	Message Length
Node Identifier	
Next Send	Next Received
AVPs . . .	

There Is Nothing Like a Good Foundation

The DIAMETER Base Protocol consists of a fixed-length (96 byte) header and two or more attribute-value pairs (AVPs). The header contains the message type, option flags, version number, and message length, followed by three transport reliability parameters (see Exhibit 8.4).

AVPs are the key to DIAMETER's extensibility. They carry all DIAMETER commands, connection parameters, and authentication, authorization, accounting, and security data. AVPs consist of a fixed-length header and a variable-length data field. A single DIAMETER message can carry any number of AVPs, up to the maximum UDP packet size of 8192 bytes. Two AVPs in each DIAMETER message are mandatory. They contain the message Command Code and the sender's IP address or host name. The message type or the Extension in use defines the remaining AVPs. DIAMETER reserves the first header byte and the first 256 AVPs for RADIUS backward compatibility.

A Is for the Way You Authenticate Me

The specifics of a DIAMETER authentication transaction are governed by the Extension in use, but they all follow a similar pattern. The client (i.e., a NAS) issues an authentication request to the server containing the AA-Request Command, a session-ID, and the client's address and host name followed by the user's name and password and a state value.

The session-ID uniquely identifies this connection and overcomes the problem in RADIUS with duplicate connection identifiers in high-density installations. Each connection has its own unique session with the server. The session is maintained for the duration of the connection and all transactions related to the connection use the same session-ID. The state AVP is used to track the state of multiple transaction authentication schemes such as CHAP or SecureID.

The server validates the user's credentials and returns an AA-Answer packet containing either a Failed-AVP or the accompanying Result-Code AVP or the authorized AVPs for the service being provided (i.e., PPP parameters, IP parameters, routing parameters, etc.). If the server is not the HOME server for this user, it will forward (proxy) the request.

Proxy on Steroids!

DIAMETER supports multiple proxy configurations, including the two RADIUS models and two additional Broker models. In the hierarchical model, the DIAMETER server forwards the request directly to the user's HOME server using a session-based connection. This approach provides several advantages over the standard RADIUS implementation. Because the proxy connection is managed separately from the client connection, failed node and packet retransmissions are handled more efficiently and the hop can be secured with enhanced security like IPSec. Under RADIUS the first server in the authentication chain must know the CHAP shared secret, but DIAMETER's proxy scheme permits the authentication to take place at the HOME server. As robust as DIAMETER's hierarchical model is, it still is not suitable for many roaming applications.

DIAMETER uses a Broker proxy server to support roaming across multiple management domains. Brokers are employed to reduce the amount of configuration information that needs to be shared between ISPs within a roaming consortium. The Broker provides a simple message routing function. In DIAMETER, two routing functions are provided: either the Broker forwards the message to the HOME server or provides the keys and certificates required for the proxy server to communicate directly with the HOME server (see Exhibit 8.5).

EXHIBIT 8.5 A Typical Entry

User Name	Attribute = Value
john	Password = "1secret9," NAS-IP-Address = 10.100.1.1 Service-Type = Framed-User Framed-Protocol = PPP, Framed-IP-Address = 10.200.10.1 Framed-IP-Netmask = 255.255.255.0 Filter-Id = "firewall"

A Two Brute: DIAMETER Authorization

Authorization transactions can be combined with authentication requests or conducted separately. The specifics of the transaction are governed by the Extension in use but follow the same pattern and use the same commands as authentications. Authorization requests must take place over an existing session; they cannot be used to initiate sessions but they can be forwarded using a DIAMETER proxy.

Accounting for Everything

DIAMETER significantly improves upon the accounting capabilities of RADIUS and TACACS+ by adding event monitoring, periodic reporting, real-time record transfer, and support for the ROAMOPS Accounting Data Interchange Format (ADIF). DIAMETER accounting is authorization-server directed. Instructions regarding how the client is to generate accounting records is passed to the client as part of the authorization process. Additionally, DIAMETER accounting servers can force a client to send current accounting data. This is particularly useful for connection troubleshooting or to capture accounting data when an accounting server experiences a crash. Client writes and server polls are fully supported by both DIAMETER proxy models.

For efficiency, records are normally batch transferred but for applications like ROAMOPS where credit limit checks or fraud detection are required, records can be generated in real-time. DIAMETER improves upon standard connect and disconnect accounting with a periodic reporting capability that is particularly useful for monitoring usage on nailed-up circuits. DIAMETER also has an event accounting capability like TACACS+ that is useful for recording service-related events like failed nodes and server reboots.

Security, Standards, and Other Sexy Stuff

Support for strong security is a standard part of the DIAMETER Base Protocol. Many applications, like ROAMOPS and Mobile-IP, require sensitive connection information to be transferred across multiple domains. Hop-by-hop security is inadequate for these applications because data is subject to exposure at each interim hop. DIAMETER's Strong Proxy Extension overcomes the problem by encrypting sensitive data in S/MIME objects and encapsulating them in standard AVPs.

Got the telecommuter, mobile workforce, VPN, multi-platform, dial-in user authentication blues? One does not need to! AAA server solutions like RADIUS, TACACS, and DIAMETER can chase those blues away. With a little careful planning and a few hours of configuration, one can increase security, reduce administration time, and consolidate one's remote access venues into a single, centralized, flexible, and scalable solution. That should put a smile on one's face.

Implementation of Access Controls

Stanley Kurzban

The decision of which access controls to implement is based on organizational policy and on two generally accepted standards of practice: separation of duties and least privilege. For controls to be accepted and, therefore, used effectively, they must not disrupt the usual work flow more than is necessary or place too many burdens on administrators, auditors, or authorized users.

To ensure that access controls adequately protect all of the organization's resources, it may be necessary to first categorize the resources. This chapter addresses this process and the various models of access controls. Methods of providing controls over unattended sessions are also discussed, and administration and implementation of access controls are examined.

CATEGORIZING RESOURCES

Policies establish levels of sensitivity (e.g., top secret, secret, confidential, and unclassified) for data and other resources. These levels should be used for guidance on the proper procedures for handling data — for example, instructions not to copy. They may be used as a basis for access control decisions as well. In this case, individuals are granted access to only those resources at or below a specific level of sensitivity. Labels are used to indicate the sensitivity level of electronically stored documents.

In addition, the access control policy may be based on compartmentalization of resources. For example, access controls may all relate to a particular project or to a particular field of endeavor (e.g., technical R&D or military intelligence). Implementation of the access controls may involve either single compartments or combinations of them. These units of involvement are called categories, though the term "compartment" and "category" are often used interchangeably. Neither term applies to restrictions on handling of data. Individuals may need authorization to all categories associated with a resource to be entitled access to it (as is the case in

the U.S. government's classification scheme) or to any one of the categories (as is more representative of how other organizations work).

The access control policy may distinguish among types of access as well. For example, only system maintenance personnel may be authorized to modify system libraries, but many if not all other users may be authorized to execute programs from those libraries. Billing personnel may be authorized to read credit files, but modification of such files may be restricted to those responsible for compiling credit data. Files with test data may be created only by testing personnel, but developers may be allowed to read and perhaps even modify such files.

One advantage of the use of sensitivity levels is that it allows security measures, which can be expensive, to be used selectively. For example, only for top-secret files might:

- The contents be zeroed after the file is deleted to prevent scavenging of a new file.
- Successful as well as unsuccessful requests for access be logged for later scrutiny, if necessary.
- Unsuccessful requests for access be reported on paper or in real-time to security personnel for action.

Although the use of sensitivity levels may be costly, it affords protection that is otherwise unavailable and may well be cost-justified in many organizations.

MANDATORY AND DISCRETIONARY ACCESS CONTROLS

Policy-based controls may be characterized as either mandatory or discretionary. With mandatory controls, only administrators and not owners of resources may make decisions that bear on or derive from policy. Only an administrator may change the category of a resource, and no one may grant a right of access that is explicitly forbidden in the access control policy.

Access controls that are not based on the policy are characterized as discretionary controls by the U.S. government and as need-to-know controls by other organizations. The latter term connotes least privilege — those who may read an item of data are precisely those whose tasks entail the need.

It is important to note that mandatory controls are prohibitive (i.e., all that is not expressly permitted is forbidden), not only permissive. Only within that context do discretionary controls operate, prohibiting still more access with the same exclusionary principle.

Discretionary access controls can extend beyond limiting which subjects can gain what type of access to which objects. Administrators can limit access to certain times of day or days of the week. Typically, the

period during which access would be permitted is 9 a.m. to 5 p.m. Monday through Friday. Such a limitation is designed to ensure that access takes place only when supervisory personnel are present, to discourage unauthorized use of data. Further, subjects' rights to access might be suspended when they are on vacation or leave of absence. When subjects leave an organization altogether, their rights must be terminated rather than merely suspended.

Supervision may be ensured by restricting access to certain sources of requests. For example, access to some resources might be granted only if the request comes from a job or session associated with a particular program, (e.g., the master PAYROLL program), a subsystem (e.g., CICS or IMS), ports, (e.g., the terminals in the area to which only bank tellers have physical access), type of port (e.g., hard-wired rather than dial-up lines), or telephone number. Restrictions based on telephone numbers help prevent access by unauthorized callers and involve callback mechanisms.

Restricting access on the basis of particular programs is a useful approach. To the extent that a given program incorporates the controls that administrators wish to exercise, undesired activity is absolutely prevented at whatever granularity the program can treat. An accounts-payable program, for example, can ensure that all the operations involved in the payment of a bill are performed consistently, with like amounts both debited and credited from the two accounts involved. If the program, which may be a higher-level entity, controls everything the user sees during a session through menus of choices, it may even be impossible for the user to try to perform any unauthorized act.

Program development provides an apt context for examination of the interplay of controls. Proprietary software under development may have a level of sensitivity that is higher than that of leased software that is being tailored for use by an organization. Mandatory policies should:

- Allow only the applications programmers involved to have access to application programs under development.
- Allow only systems programmers to have access to system programs under development.
- Allow only librarians to have write access to system and application libraries.
- Allow access to live data only through programs that are in application libraries.

Discretionary access control, on the other hand, should grant only planners access to the schedule data associated with various projects and should allow access to test cases for specific functions only to those whose work involves those functions.

When systems enforce mandatory access control policies, they must distinguish between these and the discretionary policies that offer flexibility. This must be ensured during object creation, classification downgrading, and labeling, as discussed in the following sections.

Object Creation

When a new object is created, there must be no doubt about who is permitted what type of access to it. The creating job or session may specify the information explicitly; however, because it acts on behalf of someone who may not be an administrator, it must not contravene the mandatory policies. Therefore, the newly created object must assume the sensitivity of the data it contains. If the data has been collected from sources with diverse characteristics, the exclusionary nature of the mandatory policy requires that the new object assume the characteristics of the most sensitive object from which its data derives.

Downgrading Data Classifications

Downgrading of data classifications must be effected by an administrator. Because a job or session may act on behalf of one who is not an administrator, it must not be able to downgrade data classifications. Ensuring that new objects assume the characteristics of the most sensitive object from which its data derives is one safeguard that serves this purpose. Another safeguard concerns the output of a job or session — the output must never be written into an object below the most sensitive level of the job or session being used. This is true even though the data involved may have a sensitivity well below the job or session's level of sensitivity, because tracking individual data is not always possible. This may seem like an impractically harsh precaution; however, even the best-intentioned users may be duped by a Trojan horse that acts with their authority.

Outside the Department of Defense's (DoD's) sphere, all those who may read data are routinely accorded the privilege of downgrading their classification by storing that data in a file of lower sensitivity. This is possible largely because aggregations of data may be more sensitive than the individual items of data among them. Where civil law applies, de facto upgrading, which is specifically sanctioned by DoD regulations, may be the more serious consideration. For example, courts may treat the theft of secret data lightly if notices of washroom repair are labeled secret. Nonetheless, no one has ever written of safeguards against de facto upgrading.

Labeling

When output from a job or session is physical rather than magnetic or electronic, it must bear a label that describes its sensitivity so that people can handle it in accordance with applicable policies. Although labels might

be voluminous and therefore annoying in a physical sense, even a single label can create serious problems if it is misplaced.

For example, a program written with no regard for labels may place data at any point on its output medium — for example, a printed page. A label arbitrarily placed on that page at a fixed position might overlay valuable data, causing more harm than the label could be expected to prevent. Placing the label in a free space of adequate size, even if there is one, does not serve the purpose because one may not know where to look for it and a false label may appear elsewhere on the page.

Because labeling each page of output poses such difficult problems, labeling entire print files is especially important. Although it is easy enough to precede and follow a print file with a page that describes it, protecting against counterfeiting of such a page requires more extensive measures. For example, a person may produce a page in the middle of an output file that appears to terminate that file. This person may then be able to simulate the appearance of a totally separate, misleadingly labeled file following the counterfeit page. If header and trailer pages contain a matching random number that is unpredictable and unavailable to jobs, this type of counterfeiting is impossible.

Discussions of labels usually focus on labels that reflect sensitivity to observation by unauthorized individuals, but labels can reflect sensitivity to physical loss as well. For example, ensuring that a particular file or document will always be available may be at least as important as ensuring that only authorized users can access that file or document. All the considerations discussed in this section in the context of confidentiality apply as well to availability.

ACCESS CONTROL MODELS

To permit rigorous study of access control policies, models of various policies have been developed. Early work was based on detailed definitions of policies in place in the U.S. government, but later models have addressed commercial concerns. The following sections contain the overviews of several models.

Lattice Models

In a lattice model, every resource and every user of a resource is associated with one of an ordered set of classes. The classes stemmed from the military designations top secret, secret, confidential, and unclassified. Resources associated with a particular class maybe used only by those whose associated class is as high as or higher than that of the resources. This scheme's applicability to governmentally classified data

is obvious; however, its application in commercial environments may also be appropriate.

The Bell-LaPadula Model

The lattice model took no account of the threat that might be posed by a Trojan horse lurking in a program used by people associated with a particular class that, unknown to them, copies information into a resource with a lower access level. In governmental terms, the Trojan horse would be said to effect de facto downgrading of classification. Despite the fact that there is no evidence that anyone has ever suffered a significant loss as a result of such an attack, such an attack would be very unattractive and several in the field are rightly concerned about it. Bell and LaPadula devised a model that took such an attack into account.

The Bell-LaPadula model prevents users and processes from reading above their security level, as does the lattice model (i.e., it asserts that processes with a given classification cannot read data associated with a higher classification). In addition, however, it prevents processes with any given classification from writing data associated with a lower classification. Although some might feel that the ability to write below the process's classification is a necessary function — placing data that is not sensitive, though contained in a sensitive document, into a less sensitive file so that it could be available to people who need to see it — DoD experts gave so much weight to the threat of de facto downgrading that it felt the model had to preclude it. All work sponsored by the National Computer Security Center (NCSC) has employed this model.

The term "higher", in this context, connotes more than a higher classification — it also connotes a superset of all resource categories. In asserting the Bell-LaPadula model's applicability to commercial data processing, Lipner omits mention of the fact that the requirement for a superset of categories may not be appropriate outside governmental circles.

Considerable nomenclature has arisen in the context of the Bell-LaPadula model. The read restriction is referred to as the simple security property. The write restriction is referred to as the star property, because the asterisk used as a place-holder until the property was given a more formal name was never replaced.

The Biba Model

In studying the two properties of the Bell-LaPadula model, Biba discovered a plausible notion of integrity, which he defined as prevention of unauthorized modification. The resulting Biba integrity model states that maintenance of integrity requires that data not flow from a receptacle of given integrity to a receptacle of higher integrity. For example, if a process

can write above its security level, trustworthy data could be contaminated by the addition of less trustworthy data.

The Take-Grant Model

Although auditors must be concerned with who is authorized to make what type of access to what data, they should also be concerned about what types of access to what data might become authorized without administrative intervention. This assumes that some people who are not administrators are authorized to grant authorization to others, as is the case when there are discretionary access controls. The take-grant model provides a mathematical framework for studying the results of revoking and granting authorization. As such, it is a useful analytical tool for auditors.

The Clark-Wilson Model

Wilson and Clark were among the many who had observed by 1987 that academic work on models for access control emphasized data's confidentiality rather than its integrity (i.e., the work exhibited greater concern for unauthorized observation than for unauthorized modification). Accordingly, they attempted to redress what they saw as a military view that differed markedly from a commercial one. In fact, however, what they considered a military view was not pervasive in the military.

The Clark-Wilson model consists of subject/program/object triples and rules about data, application programs, and triples. The following sections discuss the triples and rules in more detail.

Triples. All formal access control models that predate the Clark-Wilson model treat an ordered subject/object pair — that is, a user and an item or collection of data, with respect to a fixed relationship (e.g., read or write) between the two. Clark and Wilson recognized that the relationship can be implemented by an arbitrary program. Accordingly, they treat an ordered subject/program/object triple. They use the term "transformational procedure" for program to make it clear that the program has integrity-relevance because it modifies or transforms data according to a rule or procedure. Data that transformational procedures modify are called constrained data items because they are constrained in the sense that only transformational procedures may modify them and that integrity verification procedures exercise constraints on them to ensure that they have certain properties, of which consistency and conformance to the real world are two of the most significant. Unconstrained data items are all other data, chiefly the keyed input to transformational procedures.

Once subjects have been constrained so that they can gain access to objects only through specified transformational procedures, the transformational procedures can be embedded with whatever logic is needed to

effect limitation of privilege and separation of duties. The transformational procedures can themselves control access of subjects to objects at a level of granularity finer than that available to the system. What is more, they can exercise finer controls (e.g., reasonableness and consistency checks on unconstrained data items) for such purposes as double-entry bookkeeping, thus making sure that whatever is subtracted from one account is added to another so that assets are conserved in transactions.

Rules. To ensure that integrity is attained and preserved, Clark and Wilson assert, certain integrity-monitoring and integrity-preserving rules are needed. Integrity-monitoring rules are called certification rules, and integrity-preserving rules are called enforcement rules.

These certification rules address the following notions:

- Constrained data items are consistent.
- Transformational procedures act validly.
- Duties are separated.
- Accesses are logged.
- Unconstrained data items are validated.

The enforcement rules specify how the integrity of constrained data items and triples must be maintained and require that subjects' identities be authenticated, that triples be carefully managed, and that transformational procedures be executed serially and not in parallel.

Of all the models discussed, only Clark-Wilson contains elements that relate to the functions that characterize leading access control products. Unified access control generalizes notions of access rules and access types to permit description of a wide variety of access control policies.

UNATTENDED SESSIONS

Another type of access control deals with unattended sessions. Users cannot spend many hours continuously interacting with computers from the same port; everyone needs a break every so often. If resource-oriented passwords are not used, systems must associate all the acts of a session with the person who initiated it. If the session persists while its inhibitor takes a break, another person could come along and do something in that session with its initiator's authority. This would constitute a violation of security. Therefore, users must be discouraged from leaving their computers logged on when they are away from their workstations.

If administrators want users to attend their sessions, it is necessary to:

- Make it easy for people to interrupt and resume their work.
- Have the system try to detect absences and protect the session.

- Facilitate physical protection of the medium while it is unattended.
- Implement strictly human controls (e.g., training and surveillance of personnel to identify offenders).

There would be no unattended sessions if users logged off every time they left their ports. Most users do not do this because then they must log back on, and the log-on process of a typical system is neither simple nor fast. To compensate for this deficiency, some organizations use expedited log-on/log-off programs, also called suspend programs. Suspend programs do not sever any part of the physical or logical connection between a port and a host; rather, they sever the connection-maintaining resources of the host so that the port is put in a suspended state. The port can be released from suspended state only by the provision of a password or other identity-validation mechanism. Because this is more convenient for users, organizations hope that it will encourage employees to use it rather than leave their sessions unattended.

The lock function of UNIX is an example of a suspend program. Users can enter a password when suspending a session and resume it by simply reentering the same password. The password should not be the user's log-on password because an intruder could start a new session during the user's absence and run a program that would simulate the lock function, then read the user's resume password and store it in one of the intruder's own files before simulating a session-terminating failure.

Another way to prevent unattended sessions is to chain users to their sessions. For example, if a port is in an office that has a door that locks whenever it is released and only one person has a key to each door, it may not be necessary to have a system mechanism. If artifacts are used for verifying identities and the artifacts must be worn by their owners (e.g., similar to the identification badges in sensitive government buildings), extraction of the artifact can trigger automatic termination of a session. In more common environments, the best solution may be some variation of the following:

- If five minutes elapse with no signal from the port, a bell or other device sounds.
- If another half-minute elapses with no signal, automatic termination of the session, called time-out, occurs.

A system might automatically terminate a session if a user takes no action for a time interval specified by the administrator (e.g., five minutes). Such a measure is fraught with hazards, however. For example, users locked out (i.e., prevented from acting in any way the system can sense) by long-running processes will find their sessions needlessly terminated. In addition, users may circumvent the control by simulating an action, under program control, frequently enough to avoid session termination. If the system

issues no audible alarm a few seconds before termination, sessions may be terminated while users remain present. On the other hand, such an alarm may be annoying to some users. In any case, the control may greatly annoy users, doing more harm to the organization than good.

Physical protection is easier if users can simply turn a key, which they then carry with them on a break, to render an input medium and the user's session invulnerable. If that is impossible, an office's lockable door can serve the same purpose. Perhaps best for any situation is a door that always swings shut and locks when it is not being held open.

ADMINISTRATION OF CONTROLS

Administration of access controls involves the creation and maintenance of access control rules. It is a vital concern because if this type of administration is difficult, it is certain to be done poorly. The keys to effective administration are:

- Expressing rules as economically and as naturally as possible.
- Remaining ignorant of as many irrelevant distinctions as possible.
- Reducing the administrative scope to manageable jurisdictions (i.e., decentralization).

Rules can be economically expressed through use of grouping mechanisms. Administrator interfaces ensure that administrators do not have to deal with irrelevant distinctions and help reduce the administrative scope. The following sections discuss grouping and administrator interfaces.

Grouping Subjects and Objects

Reducing what must be said involves two aspects: grouping objects and grouping subjects. The resource categories represent one way of grouping objects. Another mechanism is naming. For example, all of a user's private objects may bear the user's own name within their identifiers. In that case, a single rule that states that a user may have all types of access to all of that user's own private objects may take the place of thousands or even millions of separate statements of access permission. Still another way that objects are grouped is by their types; in this case, administrators can categorize all volumes of magnetic tape or all CICS transactions. Still other methods of grouping objects are by device, directory, and library.

When subject groupings match categories, many permissions may be subsumed in a single rule that grants groups all or selected types of access to resources of specific categories. For various administrative purposes, however, groups may not represent categories; rather, they must represent organizational departments or other groupings (e.g., projects) that are not categories. Although subject grouping runs counter to the assignment-of-privilege standard, identity-based access control redresses the balance.

Whenever there are groups of subjects or objects, efficiency requires a way to make exceptions. For example, 10 individuals may have access to 10 resources. Without aggregation, an administrator must make 10 times 10 (or 100) statements to tell the system about each person's rights to access each object. With groups, only 21 statements are needed: one to identify each member of the group of subjects, one to identify each member of the group of objects, and one to specify the subjects' right of access to the objects. Suppose, however, that one subject lacks one right that the others have. If exceptions cannot be specified, either the subject or the object must be excluded from a group and nine more statements must be made. If an overriding exception can be made, it is all that must be added to the other 21 statements. Although exceptions complicate processing, only the computer need be aware of this complication.

Additional grouping mechanisms may be superimposed on the subject and object groupings. For example, sets of privileges may be associated with individuals who are grouped by being identified as, for example, auditors, security administrators, operators, or data base administrators.

Administrator Interfaces

To remain ignorant of irrelevant distinctions, administrators must have a coherent and consistent interface. What the interface is consistent with depends on the administrative context. If administrators deal with multiple subsystems, a single product can provide administrators with a single interface that hides the multiplicity of subsystems for which they supply administrative data. On the other hand, if administrators deal with single subsystems, the subsystem itself or a subsystem-specific product can provide administrators with an interface that makes administrative and other functions available to them.

The administrative burden can be kept within tolerable bounds if each administrator is responsible for only a reasonable number of individuals and functions. Functional distribution might focus on subsystems or types of resources (e.g., media or programs). When functional distribution is inadequate, decentralization is vital. With decentralized administration, each administrator may be responsible for one or more departments of an organization. In sum, effective control of access is the implementation of the policy's rules and implications to ensure that, within cost/benefit constraints, the principles of separation of duties and least privilege are upheld.

IMPLEMENTING CONTROLS

Every time a request for access to type of protected resource occurs in a job or session, an access control decision must be made. That decision must implement management's wishes, as recorded by administrators. The

program that makes the decisions has been called a reference monitor because the job or session is said to refer to a protected resource and the decision is seen as a monitoring of the references.

Although the reference monitor is defined by its function rather than by its embodiment, it is convenient to think of it as a single program. For each type of object, there is a program, called a resource manager, that must be involved in every access to each object of that type. The resource manager uses the reference monitor as an arbiter of whether to grant or deny each set of requests for access to any object of a type that it protects.

In a data base management system (DBMS) that is responding to a request for a single field, the DBMS's view-management routines act as a reference monitor. More conventional is the case of binding to a view, whereby the DBMS typically uses an external, multipurpose reference monitor to decide whether to grant or deny the job or session access to use the view.

Whatever the reference monitor's structure, it must collect, store, and use administrators' specifications of what access is to be granted. The information is essentially a simple function involving types of access permitted as defined on two fields of variables (i.e., subjects or people and objects or resources), efficient storage of the data, and the function's values. However, this function poses a complex problem.

Much of what administrators specify should be stated tersely, using an abbreviated version of many values of the function. Efficient storage of the information can mirror its statement. Indeed, this is true in the implementation of every general access control product. Simply mirroring the administrator-supplied rules is not enough, however. The stored version must be susceptible to efficient processing so that access control decisions can be made efficiently. This virtually requires that the rules be stored in a form that permits the subject's and object's names to be used as direct indexes to the rules that specify what access is permitted. Each product provides an instructive example of how this may be done.

Because rules take advantage of generalizations, however, they are inevitably less than optimum when generalizations are few. A rule that treats but one subject and one object would be an inefficient repository for a very small amount of information — the type of access permitted in this one case.

Access control information can be viewed as a matrix with rows representing the subjects, and columns representing the objects. The access that the subject is permitted to the object is shown in the body of the matrix. For example, in the matrix in Exhibit 1, the letter at an intersection of a row and a column indicates what type of access the subject may make to the object. Because least privilege is a primary goal of access control,

	OBJECTS / SUBJECTS	A	B	C	D	E	F	G	H	J	K	L
Group 1	Alex	W	W	W	R	R	R	R	R	R	R	R
	Brook	R	W	W	R							
	Chris	R	W	W	R	R						
	Denny	R	W	W	R	W	R					
Group 2	Eddie	R	R	R	W	W	W					
	Fran	R	R	R	R	W	W					
Group 3	Gabriel	R	R	R			R	W	W	R		
	Harry	R						W	W	R	R	R
	Jan							W	W	W		
Group 4	Kim	R									W	W
	Lee	R									W	W
	Meryl	R									W	W

Notes:
R Read
W Write and read

Exhibit 1. Access Control Matrix

most cells of the matrix will be empty, meaning that no access is allowed. When most of the cells are empty, the matrix is said to be sparse.

Storage of every cell's contents is not efficient if the matrix is sparse. Therefore, access control products store either the columns or the rows, as represented in Exhibits 2 and 3, which show storage of the matrix in Exhibit 1.

In Exhibit 2, a user called UACC, RACF's term for universal access, represents all users whose names do not explicitly appear in the access control lists represented in the matrix in Exhibit 1. The type of access associated with UACC is usually none, indicated by an N. In addition, groups are used to represent sets of users with the same access rights for the object in

Object	User	Access
A	UACC	R
	Alex	W
	Jan	N
B and C	UACC	N
	GP1	W
	GP2	R
	Gabriel	R
D	UACC	N
	GP1	R
	Eddie	W
	Fran	R
E	UACC	N
	Alex	R
	Chris	R
	GP2	W
F	UACC	N
	Alex	R
	Chris	N
	Denny	R
	GP2	W
F	UACC	N
	Alex	R
	Denny	R
	GP2	W
	Gabriel	R
G and H	UACC	N
	Alex	R
	GP3	W
J	UACC	N
	Alex	R
	Gabriel	R
	Harry	R
	Jan	W
K and L	UACC	N
	Alex	R
	Harry	R
	GP4	W

Notes:
GP Group
N None
R Read
W Write and read

Exhibit 2. List-Based Storage of Access Controls

question. For example, for objects B and C, GP1 (i.e., group 1) represents Alex, Brook, Chris, and Denny. Descriptions of the groups are stored separately. The grouping mechanisms reduce the amount of information that must be stored in the access control lists and the amount of keying a security administrator must do to specify all the permissions.

Exhibit 2 shows access control storage based on the columns (i.e., the lists of users whose authorized type of access to each object is recorded), called list-based storage. Unlisted users need not be denied all access. In many cases, most users are authorized some access — for example, execute

User	Object/Access
Alex	A/W, B/W, C/W, D/R, E/R, F/R, G/R, H/R, J/R, K/R, L/R
Brook	A/R, B/W, C/W, D/R
Chris	A/R, B/W, C/W, D/R, E/R
Denny	A/R, B/W, C/W, D/R, E/W, F/R
Eddie	A/R, B/R, C/R, D/W, E/W, F/W
Fran	A/R, B/R, C/R, D/R, E/W, F/W,
Gabriel	A/R, B/R, C/R, F/R, G/W, H/W, J/R
Harry	A/R, G/W, H/W, J/R, K/R, L/R
Jan	G/W, H/W, J/W
Kim	A/R, K/W, L/W
Lee	A/R, K/W, L/W
Meryl	A/R, K/W, L/W

Notes:
R Read
W Write and read

Exhibit 3. Ticket-Based Storage of Access Controls

or read access to the system's language processors — and only a few will be granted more or less authority — for example, either write or no access. An indicator in or with the list (e.g., UACC in RACF) may indicate the default type of access for the resource. List-based control is efficient because it contains only the exceptions.

Exhibit 3 shows access control storage based on the rows (i.e., the lists of objects to which the user is authorized to gain specified types of access), called ticket-based or capability-based storage. The latter term refers to rigorously defined constructs, called capabilities, that define both an object and one or more types of some access permitted to it. Capabilities may be defined by hardware or by software. The many implications of capabilities are beyond the scope of this chapter. Any pure ticket-based scheme has the disadvantage that it lacks the efficiency of a default access type per object. This problem can be alleviated, however, by grouping capabilities in shared catalogs and by grafting some list-based control onto a ticket-based scheme.

SUMMARY

Effective application security controls spring from such standards as least privilege and separation of duties. These controls must be precise and effective, but no more precise or granular than considerations of cost and value dictate. At the same time, they must place minimal burdens on administrators, auditors, and legitimate users of the system.

Controls must be built on a firm foundation of organizational policies. Although all organizations probably need the type of policy that predominates in the commercial environment, some require the more stringent type of policy that the U.S. government uses, which places additional controls on use of systems.

9

An Introduction to Secure Remote Access

Christina M. Bird, Ph.D, CISSP

In the past decade, the problem of establishing and controlling remote access to corporate networks has become one of the most difficult issues facing network administrators and information security professionals. As information-based businesses become a larger and larger fraction of the global economy, the nature of "business" itself changes. "Work" used to take place in a well-defined location — such as a factory, an office, or a store — at well-defined times, between relatively organized hierarchies of employees. But now, "work" happens everywhere: all over the world, around the clock, between employees, consultants, vendors, and customer representatives. An employee can be productive working with a personal computer and a modem in his living room, without an assembly line, a filing cabinet, or a manager in sight.

The Internet's broad acceptance as a communications tool in business and personal life has introduced the concept of remote access to a new group of computer users. They expect the speed and simplicity of Internet access to translate to their work environment as well. Traveling employees want their private network connectivity to work as seamlessly from their hotel room as if they were in their home office. This increases the demand for reliable and efficient corporate remote access systems, often within organizations for whom networking is tangential at best to the core business.

The explosion of computer users within a private network — now encompassing not only corporate employees in the office, but also telecommuters, consultants, business partners, and clients — makes the design and implementation of secure remote access even tougher. In the simplest local area networks (LANs), all users have unrestricted access to all resources on the network. Sometimes, granular access control is provided at the host computer level, by restricting log-in privileges. But in most real-world environments, access to different kinds of data — such as accounting, human resources, or research & development — must be restricted to limited groups of people. These restrictions may be provided by physically isolating resources on the network or through logical mechanisms (including router access control lists and stricter firewall technologies). Physical isolation, in particular, offers considerable protection to network resources, and sometimes develops without the result of a deliberate network security strategy.

Connections to remote employees, consultants, branch offices, and business partner networks make communications between and within a company extremely efficient; but they expose corporate networks and sensitive data to a wide, potentially untrusted population of users, and a new level of vulnerability. Allowing non-employees to use confidential information creates stringent requirements for data classification and access control. Managing a network infrastructure to enforce a corporate security policy for non-employees is a new challenge for most network administrators and security managers. Security policy must be tailored to facilitate the organization's reasonable business requirements for remote access. At the same time, policies and procedures help minimize the chances that improved connectivity will translate into compromise of data confidentiality, integrity, and availability on the corporate network.

Similarly, branch offices and customer support groups also demand cost-effective, robust, and secure network connections.

This chapter discusses general design goals for a corporate remote access architecture, common remote access implementations, and the use of the Internet to provide secure remote access through the use of virtual private networks (VPNs).

Security Goals for Remote Access

All remote access systems are designed to establish connectivity to privately maintained computer resources, subject to appropriate security policies, for legitimate users and sites located away from the main corporate campus. Many such systems exist, each with its own set of strengths and weaknesses. However, in a network environment in which the protection of confidentiality, data integrity, and availability is paramount, a secure remote access system possesses the following features:

- Reliable authentication of users and systems
- Easy-to-manage granular control of access to particular computer systems, files, and other network resources
- Protection of confidential data
- Logging and auditing of system utilization
- Transparent reproduction of the workplace environment
- Connectivity to a maximum number of remote users and locations
- Minimal costs for equipment, network connectivity, and support

Reliable Authentication of Remote Users/Hosts

It seems obvious, but it is worth emphasizing that the main difference between computer users in the office and remote users is that remote users are not there. Even in a small organization, with minimal security requirements, many informal authentication processes take place throughout the day. Co-workers recognize each other, and have an understanding about who is supposed to be using particular systems throughout the office. Similarly, they may provide a rudimentary access control mechanism if they pay attention to who is going in and out of the company's server room.

In corporations with higher security requirements, the physical presence of an employee or a computer provides many opportunities — technological and otherwise — for identification, authentication, and access control mechanisms to be employed throughout the campus. These include security guards, photographic employee ID cards, keyless entry to secured areas, among many other tools.

When users are not physically present, the problem of accurate identification and authentication becomes paramount. The identity of network users is the basis for assignment of all system access privileges that will be granted over a remote connection. When the network user is a traveling salesman 1500 miles away from corporate headquarters, accessing internal price lists and databases — a branch office housing a company's research and development organization — or a business partner with potential competitive interest in the company, reliable verification of identity allows a security administrator to grant access on a need-to-know basis within the network. If an attacker can present a seemingly legitimate identity, then that attacker can gain all of the access privileges that go along with it.

A secure remote access system supports a variety of strong authentication mechanisms for human users, and digital certificates to verify identities of machines and gateways for branch offices and business partners.

Granular Access Control

A good remote access system provides flexible control over the network systems and resources that may be accessed by an off-site user. Administrators must have fine-grain control to grant access for all appropriate business purposes while denying access for everything else. This allows management of a variety of access policies based on trust relationships with different types of users (employees, third-party contractors, etc.). The access control system must be flexible enough to support the organization's security requirements and easily modified when policies or personnel change. The remote access system should scale gracefully and enable the company to implement more complex policies as access requirements evolve.

Access control systems can be composed of a variety of mechanisms, including network-based access control lists, static routes, and host system- and application-based access filters. Administrative interfaces

can support templates and user groups, machines, and networks to help manage multiple access policies. These controls can be provided, to varying degrees, by firewalls, routers, remote access servers, and authentication servers. They can be deployed at the perimeter of a network as well as internally, if security policy so demands.

The introduction of the remote access system should not be disruptive to the security infrastructure already in place in the corporate network. If an organization has already implemented user- or directory-based security controls (e.g., based on Novell's Netware Directory Service or Windows NT domains), a remote access system that integrates with those controls will leverage the company's investment and experience.

Protection of Confidential Data

Remote access systems that use public or semi-private network infrastructure (including the Internet and the public telephone network) provide lots of opportunities for private data to fall into unexpected hands. The Internet is the most widely known public network, but it is hardly the only one. Even private Frame Relay connections and remote dial-up subscription services (offered by many telecommunications providers) transport data from a variety of locations and organizations on the same physical circuits. Frame Relay sniffers are commodity network devices that allow network administrators to examine traffic over private virtual circuits, and allow a surprising amount of eavesdropping between purportedly secure connections. Reports of packet leaks on these systems are relatively common on security mailing lists like *BUGTRAQ* and *Firewall-Wizards*.

Threats that are commonly acknowledged on the Internet also apply to other large networks and network services. Thus, even on nominally private remote access systems — modem banks and telephone lines, cable modem connections, Frame Relay circuits — security-conscious managers will use equipment that performs strong encryption and per-packet authentication.

Logging and Auditing of System Utilization

Strong authentication, encryption, and access control are important mechanisms for the protection of corporate data. But sooner or later, every network experiences accidental or deliberate disruptions, from system failures (either hardware or software), human error, or attack. Keeping detailed logs of system utilization helps to troubleshoot system failures.

If troubleshooting demonstrates that a network problem was deliberately caused, audit information is critical for tracking down the perpetrator. One's corporate security policy is only as good as one's ability to associate users with individual actions on the remote access system — if one cannot tell who did what, then one cannot tell who is breaking the rules.

Unfortunately, most remote access equipment performs rudimentary logging, at best. In most cases, call level auditing — storing username, start time, and duration of call — is recorded, but there is little information available about what the remote user is actually *doing*. If the corporate environment requires more stringent audit trails, one will probably have to design custom audit systems.

Transparent Reproduction of the Workplace Environment

For telecommuters and road warriors, remote access should provide the same level of connectivity and functionality that they would enjoy if they were physically in their office. Branch offices should have the same access to corporate headquarters networks as the central campus. If the internal network is freely accessible to employees at work, then remote employees will expect the same degree of access. If the internal network is subject to physical or logical security constraints, then the remote access system should enable those constraints to be enforced. If full functionality is not available to remote systems, priority must be given to the most business-critical resources and applications, or people will not use it.

Providing transparent connectivity can be more challenging than it sounds. Even within a small organization, personal work habits differ widely from employee to employee, and predicting how those differences might affect use of remote access is problematic. For example, consider access to data files stored on a UNIX file server. Employees with UNIX workstations use the Network File Service (NFS) protocol to access those files. NFS requires its own particular set of network connections, server configurations, and security settings in order to function properly. Employees with Windows-based workstations probably use the Server Message Bus (SMB) protocol to access the same files. SMB requires its own set of configuration files and security tuning. If the corporate remote access system fails to transport NFS

and SMB traffic as expected, or does not handle them at all, remote employees will be forced to change their day-to-day work processes.

Connectivity to Remote Users and Locations

A robust and cost-effective remote access system supports connections over a variety of mechanisms, including telephone lines, persistent private network connections, dial-on-demand network connections, and the Internet. This allows the remote access architecture to maintain its usefulness as network infrastructure evolves, whether or not all connectivity mechanisms are being used at any given time.

Support for multiple styles of connectivity builds a framework for access into the corporate network from a variety of locations: hotels, homes, branch offices, business partners, and client sites, domestic or international. This flexibility also simplifies the task of adding redundancy and performance tuning capabilities to the system.

The majority of currently deployed remote access systems, at least for employee and client-to-server remote connectivity, utilize TCP/IP as their network protocol. A smaller fraction continues to require support for IPX, NetBIOS/NetBEUI, and other LAN protocols; even fewer support SNA, DECNet, and older services. TCP/IP offers the advantage of support within most modern computer operating systems; most corporate applications either use TCP/IP as their network protocol, or allow their traffic to be encapsulated over TCP/IP networks. This chapter concentrates on TCP/IP-based remote access and its particular set of security concerns.

Minimize Costs

A good remote access solution will minimize the costs of hardware, network utilization, and support personnel. Note, of course, that the determination of appropriate expenditures for remote access, reasonable return on investment, and appropriate personnel budgets differs from organization to organization, and depends on factors including sensitivity to loss of resources, corporate expertise in network and security design, and possible regulatory issues depending on industry.

In any remote access implementation, the single highest contribution to overall cost is incurred through payments for persistent circuits, be they telephone capacity, private network connections, or access to the Internet. Business requirements will dictate the required combination of circuit types, typically based on the expected locations of remote users, the number of LAN-to-LAN connections required, and expectations for throughput and simultaneous connections. One-time charges for equipment, software, and installation are rarely primary differentiators between remote access architectures, especially in a high-security environment. However, to fairly judge between remote access options, as well as to plan for future growth, consider the following components in any cost estimates:

- One-time hardware and software costs
- Installation charges
- Maintenance and upgrade costs
- Network and telephone circuits
- Personnel required for installation and day-to-day administration

Not all remote access architectures will meet an organization's business requirements with a minimum of money and effort, so planning in the initial stages is critical.

At the time of this writing, Internet access for individuals is relatively inexpensive, especially compared to the cost of long-distance telephone charges. As long as home Internet access cost is based on a monthly flat fee rather than per-use calculations, use of the Internet to provide individual remote access, especially for traveling employees, will remain economically compelling. Depending on an organization's overall Internet strategy, replacing private network connections between branch offices and headquarters with secured Internet connections may result in savings of one third to one half over the course of a couple of years. This huge drop in cost for remote access is often the primary motivation for the evaluation of secure virtual private networks as a corporate remote access infrastructure. But note that if an organization does not already have technical staff experienced in the deployment of Internet networks and security systems, the perceived savings in terms of ongoing circuit costs can easily be lost in the attempt to hire and train administrative personnel.

It is the security architect's responsibility to evaluate remote access infrastructures in light of these requirements. Remote access equipment and service providers will provide information on the performance of their

equipment, expected administrative and maintenance requirements, and pricing. Review pricing on telephone and network connectivity regularly; the telecommunications market changes rapidly and access costs are extremely sensitive to a variety of factors, including geography, volume of voice/data communications, and the likelihood of corporate mergers.

A good remote access system is scalable, cost-effective, and easy to support. Scalability issues include increasing capacity on the remote access servers (the gateways into the private network), through hardware and software enhancements; increasing network bandwidth (data or telephone lines) into the private network; and maintaining staff to support the infrastructure and the remote users. If the system will be used to provide mission-critical connectivity, then it needs to be designed with reliable, measurable throughput and redundancy from the earliest stages of deployment. Backup methods of remote access will be required from *every* location at which mission-critical connections will originate.

Remember that not every remote access system necessarily possesses (or requires) each of these attributes. Within any given corporate environment, security decisions are based on preexisting policies, perceived threat, potential losses, and regulatory requirements — and remote access decisions, like all else, will be specific to a particular organization and its networking requirements. An organization supporting a team of 30 to 40 traveling sales staff, with a relatively constant employee population, has minimal requirements for flexibility and scalability — especially since the remote users are all trusted employees and only one security policy applies. A large organization with multiple locations, five or six business partners, and a sizable population of consultants probably requires different levels of remote access. Employee turnover and changing business conditions also demand increased manageability from the remote access servers, which will probably need to enforce multiple security policies and access control requirements simultaneously.

Remote Access Mechanisms

Remote access architectures fall into three general categories: (1) remote user access via analog modems and the public telephone network; (2) access via dedicated network connections, persistent or on-demand; and (3) access via public network infrastructures such as the Internet.

Telephones

Telephones and analog modems have been providing remote access to computer resources for the past two decades. A user, typically at home or in a hotel room, connects her computer to a standard telephone outlet and establishes a point-to-point connection to a network access server (NAS) at the corporate location. The NAS is responsible for performing user authentication, access control, and accounting, as well as maintaining connectivity while the phone connection is live. This model benefits from low end-user cost (phone charges are typically very low for local calls, and usually covered by the employer for long-distance tolls) and familiarity. Modems are generally easy to use, at least in locations with pervasive access to phone lines. Modem-based connectivity is more limiting if remote access is required from business locations, which may not be willing to allow essentially unrestricted outbound access from their facilities.

But disadvantages are plentiful. Not all telephone systems are created equal. In areas with older phone networks, electrical interference or loss of signal may prevent the remote computer from establishing a reliable connection to the NAS. Even after a connection is established, some network applications (particularly time-sensitive services such as multimedia packages and applications that are sensitive to network latency) may fail if the rate of data throughput is low. These issues are nearly impossible to resolve or control from corporate headquarters.

Modem technology changes rapidly, requiring frequent and potentially expensive maintenance of equipment. And network access servers are popular targets for hostile action because they provide a single point of entrance to the private network — a gateway that is frequently poorly protected.

Dedicated Network Connections

Branch office connectivity — network connections for remote corporate locations — and business partner connections are frequently met using dedicated private network circuits. Dedicated network connections are offered by most of the major telecommunications providers. They are generally deemed to be the safest way of connecting multiple locations because the only network traffic they carry "belongs" to the same organization.

Private network connections fall into two categories: dedicated circuits and Frame Relay circuits. Dedicated circuits are the most private, as they provide an isolated physical circuit for their subscribers (hence, the name).

The only data on a dedicated link belongs to the subscribing organization. An attacker can subvert a dedicated circuit infrastructure only by attacking the telecommunications provider itself. This offers substantial protection. But remember that telco attacks are the oldest in the hacker lexicon — most mechanisms that facilitate access to voice lines work on data circuits as well because the physical infrastructure is the same. For high-security environments, such as financial institutions, strong authentication and encryption are required even over private network connections.

Frame Relay connections provide private bandwidth over a shared physical infrastructure by encapsulating traffic in frames. The frame header contains addressing information to get the traffic to its destination reliably. But the use of shared physical circuitry reduces the security of Frame Relay connections relative to dedicated circuits. Packet leak between frame circuits is well-documented, and devices that eavesdrop on Frame Relay circuits are expensive but readily available. To mitigate these risks, many vendors provide Frame Relay-specific hardware that encrypts packet payload, protecting it against leaks and sniffing but leaving the frame headers alone.

The security of private network connections comes at a price, of course — subscription rates for private connections are typically two to five times higher than connections to the Internet, although discounts for high-volume use can be significant. Deployment in isolated areas is challenging if telecommunications providers fail to provide the required equipment in those areas.

Internet-Based Remote Access

The most cost-effective way to provide access into a corporate network is to take advantage of shared network infrastructure whenever feasible. The Internet provides ubiquitous, easy-to-use, inexpensive connectivity. However, important network reliability and security issues must be addressed.

Internet-based remote user connectivity and wide area networks are much less expensive than in-house modem banks and dedicated network circuits, both in terms of direct charges and in equipment maintenance and ongoing support. Most importantly, ISPs manage modems and dial-in servers, reducing the support load and upgrade costs on the corporate network/telecommunications group.

Of course, securing private network communications over the Internet is a paramount consideration. Most TCP/IP protocols are designed to carry data in cleartext, making communications vulnerable to eavesdropping attacks. Lack of IP authentication mechanisms facilitates session hijacking and unauthorized data modification (while data is in transit). A corporate presence on the Internet may open private computer resources to denial-of-service attacks, thereby reducing system availability. Ongoing development of next-generation Internet protocols, especially IPSec, will address many of these issues. IPSec adds per-packet authentication, payload verification, and encryption mechanisms to traditional IP. Until it becomes broadly implemented, private security systems must explicitly protect sensitive traffic against these attacks.

Internet connectivity may be significantly less reliable than dedicated network links. Troubleshooting Internet problems can be frustrating, especially if an organization has typically managed its wide area network connections in-house. The lack of any centralized authority on the Internet means that resolving service issues, including packet loss, higher than expected latency, and loss of packet exchange between backbone Internet providers, can be time-consuming. Recognizing this concern, many of the national Internet service providers are beginning to offer "business class" Internet connectivity, which provides service level agreements and improved monitoring tools (at a greater cost) for business-critical connections.

Given mechanisms to ensure some minimum level of connectivity and throughput, depending on business requirements, VPN technology can be used to improve the security of Internet-based remote access. For the purposes of this discussion, a VPN is a group of two or more privately owned and managed computer systems that communicates "securely" over a public network (see Exhibit 9.1).

Security features differ from implementation to implementation, but most security experts agree that VPNs include encryption of data, strong authentication of remote users and hosts, and mechanisms for hiding or masking information about the private network topology from potential attackers on the public network. Data in transmission is encrypted between the remote node and the corporate server, preserving data confidentiality and integrity. Digital signatures verify that data has not been modified. Remote users and hosts are subject to strong authentication and authorization mechanisms, including one-time password generators and digital certificates. These help to guarantee that only appropriate personnel can access and modify corporate data. VPNs can prevent private network addresses from being propagated over the public network, thus hiding potential target machines from attackers attempting to disrupt service.

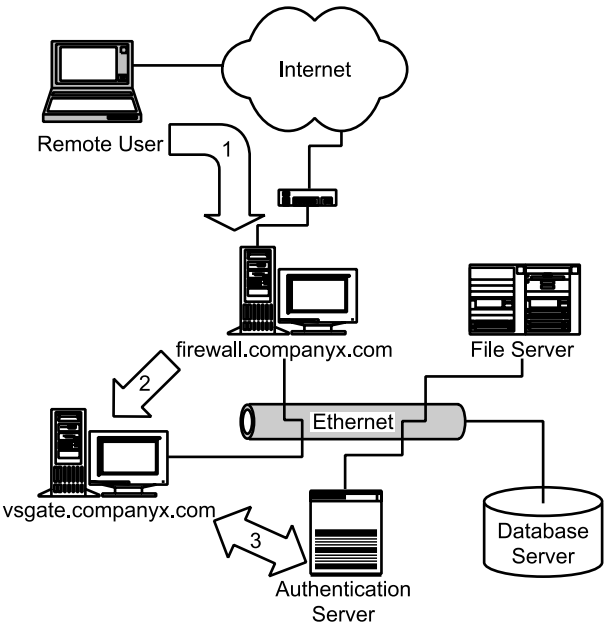


EXHIBIT 9.1 Remote user VPN.

In most cases, VPN technology is deployed over the Internet (see Exhibit 9.2), but there are other situations in which VPNs can greatly enhance the security of remote access. An organization may have employees working at a business partner location or a client site, with a dedicated private network circuit back to the home campus. The organization may choose to employ a VPN application to connect its own employees back into their home network — protecting sensitive data from potential eavesdropping on the business partner network. In general, whenever a connection is built between a private network and an entity over which the organization has no administrative or managerial control, VPN technology provides valuable protection against data compromise and loss of system integrity.

When properly implemented, VPNs provide granular access control, accountability, predictability, and robustness at least equal to that provided by modem-based access or Frame Relay circuits. In many cases, because network security has been a consideration throughout the design of VPN products, they provide a higher level of control, auditing capability, and flexibility than any other remote access technology.

Virtual Private Networks

The term "virtual private network" is used to mean many different things. Many different products are marketed as VPNs, but offer widely varying functionality. In the most general sense, a VPN allows remote sites to communicate as if their networks were directly connected. VPNs also enable multiple independent networks to operate over a common infrastructure. The VPN is implemented as part of the system's networking. That is, ordinary programs like Web servers and e-mail clients see no difference between connections across a physical network and connections across a VPN.

VPN technologies fall into a variety of categories, each designed to address distinct sets of concerns. VPNs designed for secure remote access implement cryptographic technology to ensure the confidentiality, authenticity, and integrity of traffic carried on the VPN. These are sometimes referred to as secure VPNs or crypto VPNs. In this context, private suggests confidentiality and has specific security implications: namely, that the data will be encoded so as to be unreadable, and unmodified, by unauthorized parties.

Some VPN products are aimed at network service providers. These service providers — including AT&T, UUNET, and MCI/Sprint, to name only a few — built and maintain large telecommunications networks, using infrastructure technologies like Frame Relay and ATM. The telecom providers manage large IP networks based

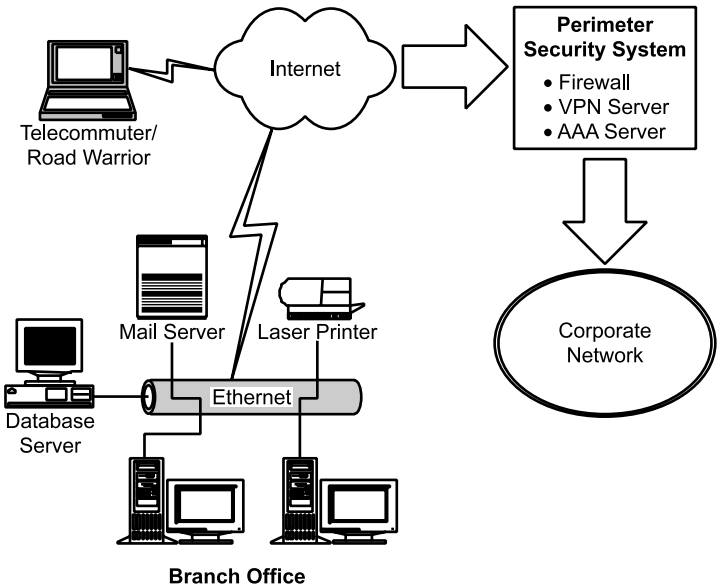


EXHIBIT 9.2 Intranet WAN over VPN.

on this private infrastructure. For them, the ability to manage multiple IP networks using a single infrastructure might be called a VPN. Some network equipment vendors offer products for this purpose and call them VPNs.

When a network service provider offers this kind of service to an enterprise customer, it is marketed as equivalent to a private, leased-line network in terms of security and performance. The fact that it is implemented over an ATM or Frame Relay infrastructure does not matter to the customer, and is rarely made apparent. These so-called VPN products are designed for maintenance of telecom infrastructure, not for encapsulating private traffic over public networks like the Internet, and are therefore addressing a different problem. In this context, the private aspect of a VPN refers only to network routing and traffic management. It does not imply the use of security mechanisms such as encryption or strong authentication.

Adding further confusion to the plethora of definitions, many telecommunications providers offer subscription dial-up services to corporate customers. These services are billed as "private network access" to the enterprise computer network. They are less expensive for the organization to manage and maintain than in-house access servers because the telecom provider owns the telephone circuits and network access equipment.

But let the buyer beware. Although the providers tout the security and privacy of the subscription services, the technological mechanisms provided to help guarantee privacy are often minimal. The private network points-of-presence in metropolitan areas that provide local telephone access to the corporate network are typically co-located with the provider's Internet access equipment, sometimes running over the same physical infrastructure. Thus, the security risks are often equivalent to using a bare-bones Internet connection for corporate access, often without much ability for customers to monitor security configurations and network utilization. Two years ago, the services did not encrypt private traffic. After much criticism, service providers are beginning to deploy cryptographic equipment to remedy this weakness.

Prospective customers are well-advised to question providers on the security and accounting within their service. The security considerations that apply to applications and hardware employed within an organization apply to network service providers as well, and are often far more difficult to evaluate. Only someone familiar with a company's security environment and expectations can determine whether or not they are supported by a particular service provider's capabilities.

Selecting A Remote Access System

For organizations with small, relatively stable groups of remote users (whether employees or branch offices), the cost benefits of VPN deployment are probably minimal relative to the traditional remote access methods.

However, for dynamic user populations, complex security policies, and expanding business partnerships, VPN technology can simplify management and reduce expenses:

- VPNs enable traveling employees to access the corporate network over the Internet. By using remote sites' existing Internet connections where available, and by dialing into a local ISP for individual access, expensive long-distance charges can be avoided.
- VPNs allow employees working at customer sites, business partners, hotels, and other untrusted locations to access a corporate network safely over dedicated, private connections.
- VPNs allow an organization to provide customer support to clients using the Internet, while minimizing risks to the client's computer networks.

For complex security environments requiring the simultaneous support of multiple levels of access to corporate servers, VPNs are ideal. Most VPN systems interoperate with a variety of perimeter security devices, such as firewalls. VPNs can utilize many different central authentication and auditing servers, simplifying management of the remote user population. Authentication, authorization, and accounting (AAA) servers can also provide granular assignment of access to internal systems. Of course, all this flexibility requires careful design and testing — but the benefits of the initial learning curve and implementation effort are enormous.

Despite the flexibility and cost advantages of using VPNs, they may not be appropriate in some situations; for example:

1. VPNs reduce costs by leveraging existing Internet connections. If remote users, branch offices, or business partners lack adequate access to the Internet, then this advantage is lost.
2. If the required applications rely on non-IP traffic, such as SNA or IPX, then the VPNs are more complex. Either the VPN clients and servers must support the non-IP protocols, or IP gateways (translation devices) must be included in the design. The cost and complexity of maintaining gateways in one's network must be weighed against alternatives like dedicated Frame Relay circuits, which can support a variety of non-IP communications.
3. In some industries and within some organizations, the use of the Internet for transmission of private data is forbidden. For example, the federal Health Care Finance Administration does not allow the Internet to be used for transmission of patient-identifiable Medicare data (at the time of this writing). However, even within a private network, highly sensitive data in transmission may be best protected through the use of cryptographic VPN technology, especially bulk encryption of data and strong authentication/digital certificates.

Remote Access Policy

A formal security policy sets the goals and ground rules for all of the technical, financial, and logistical decisions involved in solving the remote access problem (and in the day-to-day management of all IT resources). Computer security policies generally form only a subset of an organization's overall security framework; other areas include employee identification mechanisms, access to sensitive corporate locations and resources, hiring and termination procedures, etc.

Few information security managers or auditors believe that their organizations have well-documented policy. Configurations, resources, and executive philosophy change so regularly that maintaining up-to-date documentation can be prohibitive. But the most effective security policies define expectations for the use of computing resources within the company, and for the behavior of users, operations staff, and managers on those computer systems. They are built on the consensus of system administrators, executives, and legal and regulatory authorities within the organization. Most importantly, they have clear management support and are enforced fairly and evenly throughout the employee population.

Although the anatomy of a security policy varies from company to company, it typically includes several components.

- A concisely stated *purpose* defines the security issue under discussion and introduces the rest of the document.
- The *scope* states the intended audience for the policy, as well as the chain of oversight and authority for enforcement.

- The *introduction* provides background information for the policy, and its cultural, technical, and economic motivators.
- *Usage expectations* include the responsibilities and privileges with regard to the resource under discussion. This section should include an explicit statement of the corporate ownership of the resource.
- The final component covers *system auditing and violation of policy*: an explicit statement of an employee's right to privacy on corporate systems, appropriate use of ongoing system monitoring, and disciplinary action should a violation be detected.

Within the context of remote access, the scope needs to address which employees qualify for remote access to the corporate network. It may be tempting to give access to everyone who is a "trusted" user of the local network. However, need ought to be justified on a case-by-case basis, to help minimize the risk of inappropriate access.

A sample remote access policy is included in Exhibit 9.3.

Another important issue related to security policy and enforcement is ongoing, end-user education. Remote users require specific training, dealing with the appropriate use of remote connectivity; awareness of computer security risks in homes, hotels, and customer locations, especially related to unauthorized use and disclosure of confidential information; and the consequences of security breaches within the remote access system.

EXHIBIT 9.3 Sample Remote Access Policy

Purpose of Policy: To define expectations for use of the corporate remote access server (including access via the modem bank and access via the Internet); to establish policies for accounting and auditing of remote access use; and to determine the chain of responsibility for misuse of the remote access privilege.

Intended Audience: This document is provided as a guideline to all employees requesting access to corporate network computing resources from non-corporate locations.

Introduction: Company X provides access to its corporate computing environment for telecommuters and traveling employees. This remote connectivity provides convenient access into the business network and facilitates long-distance work. But it also introduces risk to corporate systems: risk of inappropriate access, unauthorized data modification, and loss of confidentiality if security is compromised. For this reason, Company X provides the following standards for use of the remote access system.

All use of the Company X remote access system implies knowledge of and compliance with this policy.

Requirements for Remote Access: An employee requesting remote access to the Company X computer network must complete the *Remote Access Agreement*, available on the internal Web server or from the Human Resources group. The form includes the following information: employee's name and log-in ID; job title, organizational unit, and direct manager; justification for the remote access; and a copy of remote user responsibilities. After completing the form, and acknowledging acceptance of the usage policy, the employee must obtain the manager's signature and send the form to the Help Desk.

EXHIBIT 9.3 Sample Remote Access Policy (continued)

NO access will be granted unless all fields are complete.

The Human Resources group will be responsible for annually reviewing ongoing remote access for employees. This review verifies that the person is still employed by Company X and that their role still qualifies them for use of the remote access system. Human Resources is also responsible for informing the IT/Operations group of employee terminations within one working day of the effective date of termination.

IT/Operations is responsible for maintaining the modem-based and Internet-based remote access systems; maintaining the user authentication and authorization servers; and auditing use of the remote access system (recording start and end times of access and user IDs for chargeback accounting to the appropriate organizational units).

Remote access users are held ultimately responsible for the use of their system accounts. The user must protect the integrity of Company X resources by safeguarding modem telephone numbers, log-in processes and start-up scripts; by maintaining their strong authentication tokens in their own possession at all times; and by NOT connecting their remote computers to other private networks at the same time that the Company X connection is active. [This provision does not include private networks maintained solely by the employee within their own home, so long as the home network does not contain independent connections to the Internet or other private (corporate) environments.] Use of another employee's authentication token, or loan of a personal token to another individual, is strictly forbidden.

Unspecified actions that may compromise the security of Company X computer resources are also forbidden. IT/Operations will maintain ongoing network monitoring to verify that the remote access system is being used appropriately. Any employee who suspects that the remote access system is being misused is required to report the misuse to the Help Desk immediately.

Violation of this policy will result in disciplinary action, up to and including termination of employment or criminal prosecution.

10

Hacker Tools and Techniques

Ed Skoudis, CISSP

Recent headlines demonstrate that the latest crop of hacker tools and techniques can be highly damaging to an organization's sensitive information and reputation. With the rise of powerful, easy-to-use, and widely distributed hacker tools, many in the security industry have observed that today is the golden age of hacking. The purpose of this chapter is to describe the tools in widespread use today for compromising computer and network security. Additionally, for each tool and technique described, the chapter presents practical advice on defending against each type of attack.

The terminology applied to these tools and their users has caused some controversy, particularly in the computer underground. Traditionally, and particularly in the computer underground, the term "hacker" is a benign word, referring to an individual who is focused on determining how things work and devising innovative approaches to addressing computer problems. To differentiate these noble individuals from a nasty attacker, this school of thought labels malicious attackers as "crackers." While hackers are out to make the world a better place, crackers want to cause damage and mayhem. To avoid the confusion often associated with these terms, in this chapter, the terms "system and security administrator" and "security practitioner" will be used to indicate an individual who has a legitimate and authorized purpose for running these tools. The term "attacker" will be used for those individuals who seek to cause damage to systems or who are not authorized to run such tools.

Many of the tools described in this chapter have dual personalities; they can be used for good or evil. When used by malicious individuals, the tools allow a motivated attacker to gain access to a network, mask the fact that a compromise occurred, or even bring down service, thereby impacting large masses of users. When used by security practitioners with proper authorization, some tools can be used to measure the security stance of their own organizations, by conducting "ethical hacking" tests to find vulnerabilities before attackers do.

Caveat

The purpose of this chapter is to explain the various computer underground tools in use today, and to discuss defensive techniques for addressing each type of tool. This chapter is *not* designed to encourage attacks. Furthermore, the tools described below are for illustration purposes only, and mention in this chapter is *not* an endorsement. If readers feel compelled to experiment with these tools, they should do so at their own risk, realizing that such tools frequently have viruses or other undocumented features that could damage networks and information systems. Curious readers who want to use these tools should conduct a through review of the source code, or at least install the tools on a separate, air-gapped network to protect sensitive production systems.

General Trends in the Computer Underground

The Smart Get Smarter, and the Rise of the Script Kiddie

The best and brightest minds in the computer underground are conducting probing research and finding new vulnerabilities and powerful, novel attacks on a daily basis. The ideas and deep research done by super-smart attackers and security practitioners are being implemented in software programs and scripts. Months of research into how a particular operating system implements its password scheme is being rendered in code, so even a clueless attacker (often called a "script kiddie") can conduct a highly sophisticated attack with just a point-and-click. Although the script kiddie may not understand the tools' true function and nuances, most of the attack is automated.

In this environment, security practitioners must be careful not to underestimate their adversaries' capabilities. Often, security and system administrators think of their potential attackers as mere teenage kids cruising the Internet looking for easy prey. While this assessment is sometimes accurate, it masks two major concerns. First, some of these teenage kids are amazingly intelligent, and can wreak havoc on a network. Second, attackers may not be just kids; organized crime, terrorists, and even foreign governments have taken to sponsoring cyberattacks.

Wide Distribution of High-Quality Tools

Another trend in the computing underground involves the widespread distribution of tools. In the past (a decade ago), powerful attack tools were limited to a core group of elites in the computer underground. Today, hundreds of Web sites are devoted to the sharing of tools for every attacker (and security practitioner) on the planet. FAQs abound describing how to penetrate any type of operating system. These overall trends converge in a world where smart attackers have detailed knowledge of undermining our systems, while the not-so-smart attackers grow more and more plentiful. To address this increasing threat, system administrators and security practitioners must understand these tools and how to defend against them. The remainder of this chapter describes many of these very powerful tools in widespread use today, together with practical defensive tips for protecting one's network from each type of attack.

Network Mapping and Port Scanning

When launching an attack across a TCP/IP network (such as the Internet or a corporate intranet), an attacker needs to know what addresses are active, how the network topology is constructed, and which services are available. A network mapper identifies systems that are connected to the target network. Given a network address range, the network mapper will send packets to each possible address to determine which addresses have machines.

By sending a simple Internet Control Message Protocol (ICMP) packet to a server (a "ping"), the mapping tool can discover if a server is connected to the network. For those networks that block incoming pings, many of the mapping tools available today can send a single SYN packet to attempt to open a connection to a server. If a server is listening, the SYN packet will trigger an ACK if the port is open, and potentially a "Port Unreachable" message if the port is closed. Regardless of whether the port is open or closed, the response indicates that the address has a machine listening. With this list of addresses, an attacker can refine the attack and focus on these listening systems.

A port scanner identifies open ports on a system. There are 65,535 TCP ports and 65,535 UDP ports, some of which are open on a system, but most of which are closed. Common services are associated with certain ports. For example, TCP Port 80 is most often used by Web servers, TCP Port 23 is used by Telnet daemons, and TCP Port 25 is used for server-to-server mail exchange across the Internet. By conducting a port scan, an attacker will send packets to each and every port. Essentially, ports are rather like doors on a machine. At any one of the thousands of doors available, common services will be listening. A port scanning tool allows an attacker to knock on every one of those doors to see who answers.

Some scanning tools include TCP fingerprinting capabilities. While the Internet Engineering Task Force (IETF) has carefully specified TCP and IP in various Requests for Comments (RFCs), not all packet options have standards associated with them. Without standards for how systems should respond to illegal packet formats, different vendors' TCP/IP stacks respond differently to illegal packets. By sending various combina-

tions of illegal packet options (such as initiating a connection with an RST packet, or combining other odd and illegal TCP code bits), an attacker can determine what type of operating system is running on the target machine. For example, by conducting a TCP fingerprinting scan, an attacker can determine if a machine is running Cisco IOS, Sun Solaris, or Microsoft Windows 2000. In some cases, even the particular version or service pack level can be determined using this technique.

After utilizing network mapping tools and port scanners, an attacker will know which addresses on the target network have listening machines, which ports are open on those machines (and therefore which services are running), and which operating system platforms are in use. This treasure trove of information is useful to the attacker in refining the attack. With this data, the attacker can search for vulnerabilities on the particular services and systems to attempt to gain access.

Nmap, written by Fyodor, is one of the most full-featured mapping and scanning tools available today. Nmap, which supports network mapping, port scanning, and TCP fingerprinting, can be found at http://www.insecure. org/nmap.

Network Mapping and Port Scanning Defenses

To defend against network mapping and port scans, the administrator should remove all unnecessary systems and close all unused ports. To accomplish this, the administrator must disable and remove unneeded services from the machine. Only those services that have an absolute, defined business need should be running. A security administrator should also periodically scan the systems to determine if any unneeded ports are open. When discovered, these unneeded ports must be disabled.

Vulnerability Scanning

Once the target systems are identified with a port scanner and network mapper, an attacker will search to determine if any vulnerabilities are present on the victim machines. Thousands of vulnerabilities have been discovered, allowing a remote attacker to gain a toehold on a machine or to take complete administrative control. An attacker could try each of these vulnerabilities on each system by entering individual commands to test for every vulnerability, but conducting an exhaustive search could take years. To speed up the process, attackers use automated scanning tools to quickly search for vulnerabilities on the target.

These automated vulnerability scanning tools are essentially databases of well-known vulnerabilities with an engine that can read the database, connect to a machine, and check to see if it is vulnerable to the exploit. The effectiveness of the tool in discovering vulnerabilities depends on the quality and thoroughness of its vulnerability database. For this reason, the best vulnerability scanners support the rapid release and update of the vulnerability database and the ability to create new checks using a scripting language.

High-quality commercial vulnerability scanning tools are widely available, and are often used by security practitioners and attackers to search for vulnerabilities. On the freeware front, SATAN (the Security Administrator Tool for Analyzing Network) was one of the first widely distributed automated vulnerability scanners, introduced in 1995. More recently, Nessus has been introduced as a free, open-source vulnerability scanner available at http://www.nessus.org. The Nessus project, which is led by Renaud Deraison, provides a full-featured scanner for identifying vulnerabilities on remote systems. It includes source code and a scripting language for writing new vulnerability checks, allowing it to be highly customized by security practitioners and attackers alike.

While Nessus is a general-purpose vulnerability scanner, looking for holes in numerous types of systems and platforms, some vulnerability scanners are much more focused on particular types of systems. For example, Whisker is a full-feature vulnerability scanning tool focusing on Web server CGI scripts. Written by Rain Forest Puppy, Whisker can be found at http://www.wiretrip.net/rfp.

Vulnerability Scanning Defenses

As described above, the administrator must close unused ports. Additionally, to eliminate the vast majority of system vulnerabilities, system patches must be applied in a timely fashion. All organizations using computers should have a defined change control procedure that specifies when and how system patches will be kept up-to-date.

Security practitioners should also conduct periodic vulnerability scans of their own networks to find vulnerabilities before attackers do. These scans should be conducted on a regular basis (such as quarterly or even monthly for sensitive networks), or when major network changes are implemented. The discovered vulnerabilities must be addressed in a timely fashion by updating system configurations or applying patches.

Wardialing

A cousin of the network mapper and scanner, a wardialing tool is used to discover target systems across a telephone network. Organizations often spend large amounts of money in securing their network from a full, frontal assault over the Internet by implementing a firewall, intrusion detection system, and secure DMZ. Unfortunately, many attackers avoid this route and instead look for other ways into the network. Modems left on users' desktops or old, forgotten machines often provide the simplest way into a target network.

Wardialers, also known as "demon dialers," dial a series of telephone numbers, attempting to locate modems on the victim network. An attacker will determine the telephone extensions associated with the target organization. This information is often gleaned from a Web site listing telephone contacts, employee newsgroup postings with telephone contact information in the signature line, or even general employee e-mail. Armed with one or a series of telephone numbers, the attacker will enter into the wardialing tool ranges of numbers associated with the original number (for example, if an employee's telephone number in a newsgroup posting is listed as 555-1212, the attacker will dial 555-XXXX). The wardialer will automatically dial each number, listen for the familiar wail of a modem carrier tone, and make a list of all telephone numbers with modems listening.

With the list of modems generated by the wardialer, the attacker will dial each discovered modem using a terminal program or other client. Upon connecting to the modem, the attacker will attempt to identify the system based on its banner information and see if a password is required. Often, no password is required, because the modem was put in place by a clueless user requiring after-hours access and not wanting to bother using approved methods. If a password is required, the attacker will attempt to guess passwords commonly associated with the platform or company.

Some wardialing tools also support the capability of locating a repeat dial-tone, in addition to the ability to detect modems. The repeat dial-tone is a great find for the attacker, as it could allow for unrestricted dialing from a victim's PBX system to anywhere in the world. If an attacker finds a line on PBX supporting repeat dial-tone in the same local dialing exchange, the attacker can conduct international wardialing, with all phone bills paid for by the victim with the misconfigured PBX.

The most fully functional wardialing tool available today is distributed by The Hacker's Choice (THC) group. Known as THC-Scan, the tool was written by Van Hauser and can be found at http://inferno.tusculum.edu/thc. THC-Scan 2.0 supports many advanced features, including sequential or randomized dialing, dialing through a network out-dial, modem carrier and repeat dial-tone detection, and rudimentary detection avoidance capabilities.

Wardialing Defenses

The best defense against wardialing attacks is a strong modem policy that prohibits the use of modems and incoming lines without a defined business need. The policy should also require the registration of all modems with a business need in a centralized database only accessible by a security or system administrator.

Additionally, security personnel should conduct periodic wardialing exercises of their own networks to find the modems before the attackers do. When a phone number with an unregistered modem is discovered, the physical device must be located and deactivated. While finding such devices can be difficult, network defenses depend on finding these renegade modems before an attacker does.

Network Exploits: Sniffing, Spoofing, and Session Hijacking

TCP/IP, the underlying protocol suite that makes up the Internet, was not originally designed to provide security services. Likewise, the most common data-link type used with TCP/IP, Ethernet, is fundamentally unsecure. A whole series of attacks are possible given these vulnerabilities of the underlying protocols. The

most widely used and potentially damaging attacks based on these network vulnerabilities are sniffing, spoofing, and session hijacking.

Sniffing

Sniffers are extremely useful tools for an attacker and are therefore a fundamental element of an attacker's toolchest. Sniffers allow an attacker to monitor data passing across a network. Given their capability to monitor network traffic, sniffers are also useful for security practitioners and network administrators in troubleshooting networks and conducting investigations. Sniffers exploit characteristics of several data-link technologies, including Token Ring and especially Ethernet.

Ethernet, the most common LAN technology, is essentially a broadcast technology. When Ethernet LANs are constructed using hubs, all machines connected to the LAN can monitor all data on the LAN segment. If userIDs, passwords, or other sensitive information are sent from one machine (e.g., a client) to another machine (e.g., a server or router) on the same LAN, all other systems connected to the LAN could monitor the data. A sniffer is a hardware or software tool that gathers all data on a LAN segment. When a sniffer is running on a machine gathering all network traffic that passes by the system, the Ethernet interface and the machine itself are said to be in "promiscuous mode."

Many commonly used applications, such as Telnet, FTP, POP (the Post Office Protocol used for e-mail), and even some Web applications, transmit their passwords and sensitive data without any encryption. Any attacker on a broadcast Ethernet segment can use a sniffer to gather these passwords and data.

Attackers who take over a system often install a software sniffer on the compromised machine. This sniffer acts as a sentinel for the attacker, gathering sensitive data that moves by the compromised system. The sniffer gathers this data, including passwords, and stores it in a local file or transmits it to the attacker. The attacker then uses this information to compromise more and more systems. The attack methodology of installing a sniffer on one compromised machine, gathering data passing that machine, and using the sniffed information to take over other systems is referred to as an island-hopping attack.

Numerous sniffing tools are available across the Internet. The most fully functional sniffing tools include sniffit (by Brecht Claerhout, available at http://reptile.rug.ac.be/~coder/sniffit/sniffit.html) and Snort (by Martin Roesch, available at http://www.clark.net/~roesch/security.html). Some operating systems ship with their own sniffers installed by default, notably Solaris (with the snoop tool) and some varieties of Linux (which ship with tcpdump). Other commercial sniffers are also available from a variety of vendors.

Sniffing Defenses

The best defense against sniffing attacks is to encrypt the data in transit. Instead of sending passwords or other sensitive data in cleartext, the application or network should encrypt the data (SSH, secure Telnet, etc.).

Another defense against sniffers is to eliminate the broadcast nature of Ethernet. By utilizing a switch instead of a hub to create a LAN, the damage that can be done with a sniffer is limited. A switch can be configured so that only the required source and destination ports on the switch carry the traffic. Although they are on the same LAN, all other ports on the switch (and the machines connected to those ports) do not see this data. Therefore, if one system is compromised on a LAN, a sniffer installed on this machine will not be capable of seeing data exchanged between other machines on the LAN. Switches are therefore useful in improving security by minimizing the data a sniffer can gather, and also help to improve network performance.

IP Spoofing

Another network-based attack involves altering the source address of a computer to disguise the attacker and exploit weak authentication methods. IP address spoofing allows an attacker to use the IP address of another machine to conduct an attack. If the target machines rely on the IP address to authenticate, IP spoofing can give an attacker access to the systems. Additionally, IP spoofing can make it very difficult to apprehend an attacker, because logs will contain decoy addresses and not the real source of the attack. Many of the tools described in other sections of this chapter rely on IP spoofing to hide the true origin of the attack.

Spoofing Defenses

Systems should not use IP addresses for authentication. Any functions or applications that rely solely on IP address for authentication should be disabled or replaced. In UNIX, the "**r**-commands" (**rlogin**, **rsh**, **rexec**,

and **rcp**) are notoriously subject to IP spoofing attacks. UNIX trust relationships allow an administrator to manage systems using the **r**-commands without providing a password. Instead of a password, the IP address of the system is used for authentication. This major weakness should be avoided by replacing the **r**-commands with administration tools that utilize strong authentication. One such tool, secure shell (ssh), uses strong cryptography to replace the weak authentication of the **r**-commands. Similarly, all other applications that rely on IP addresses for critical security and administration functions should be replaced.

Additionally, an organization should deploy anti-spoof filters on its perimeter networks that connect the organization to the Internet and business partners. Anti-spoof filters drop all traffic coming from outside the network claiming to come from the inside. With this capability, such filters can prevent some types of spoofing attacks, and should be implemented on all perimeter network routers.

Session Hijacking

While sniffing allows an attacker to view data associated with network connections, a session hijack tool allows an attacker to take over network connections, kicking off the legitimate user or sharing a login. Session hijacking tools are used against services with persistent login sessions, such as Telnet, rlogin, or FTP. For any of these services, an attacker can hijack a session and cause a great deal of damage.

A common scenario illustrating session hijacking involves a machine, Alice, with a user logged in to remotely administer another system, Bob, using Telnet. Eve, the attacker, sits on a network segment between Alice and Bob (either Alice's LAN, Bob's LAN, or between any of the routers between Alice's and Bob's LANs). Exhibit 10.1 illustrates this scenario in more detail.

Using a session hijacking tool, Eve can do any of the following:

- *Monitor Alice's session.* Most session hijacking tools allow attackers to monitor all connections available on the network and select which connections they want to hijack.
- *Insert commands into the session.* An attacker may just need to add one or two commands into the stream to reconfigure Bob. In this type of hijack, the attacker never takes full control of the session. Instead, Alice's login session to Bob has a small number of commands inserted, which will be executed on Bob as though Alice had typed them.
- *Steal the session.* This feature of most session hijacking tools allows an attacker to grab the session from Alice, and directly control it. Essentially, the Telnet client control is shifted from Alice to Eve, without Bob's knowing.
- *Give the session back.* Some session hijacking tools allow the attacker to steal a session, interact with the server, and then smoothly give the session back to the user. While the session is stolen, Alice is put on hold while Eve controls the session. With Alice on hold, all commands typed by Alice are displayed on Eve's screen, but not transmitted to Bob. When Eve is finished making modifications on Bob, Eve transfers control back to Alice.

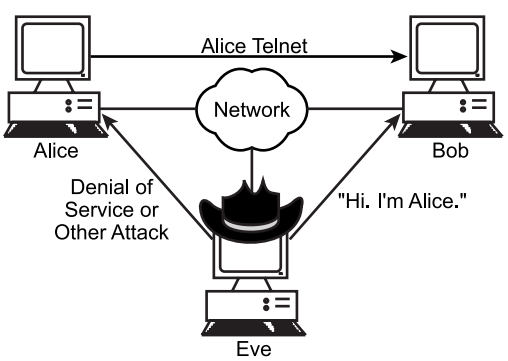


EXHIBIT 10.1 Eve hijacks the session between Alice and Bob.

For a successful hijack to occur, the attacker must be on a LAN segment between Alice and Bob. A session hijacking tool monitors the connection using an integrate sniffer, observing the TCP sequence numbers of the packets going each direction. Each packet sent from Alice to Bob has a unique TCP sequence number used by Bob to verify that all packets are received and put in proper order. Likewise, all packets going back from Bob to Alice have sequence numbers. A session hijacking tool sniffs the packets to determine these sequence numbers. When a session is hijacked (through command insertion or session stealing), the hijacking tool automatically uses the appropriate sequence numbers and spoofs Alice's address, taking over the conversation with Bob where Alice left off.

One of the most fully functional session hijacking tool available today is Hunt, written by Kra and available at http://www.cri.cz/kra/index.html. Hunt allows an attacker to monitor and steal sessions, insert single commands, and even give a session back to the user.

Session Hijacking Defenses

The best defense against session hijacking is to avoid the use of insecure protocols and applications for sensitive sessions. Instead of using the easy-to-hijack (and easy-to-sniff) Telnet application, a more secure, encrypted session tool should be used. Because the attacker does not have the session encryption keys, an encrypted session cannot be hijacked. The attacker will simply see encrypted gibberish using Hunt, and will only be able to reset the connection, not take it over or insert commands.

Secure shell (ssh) offers strong authentication and encrypted sessions, providing a highly secure alternative to Telnet and rlogin. Furthermore, ssh includes a secure file transfer capability (scp) to replace traditional FTP. Other alternatives are available, including secure, encrypted Telnet or a virtual private network (VPN) established between the source and destination.

Denial-of-Service Attacks

Denial-of-service attacks are among the most common exploits available today. As their name implies, a denial-of-service attack prevents legitimate users from being able to access a system. With E-commerce applications constituting the lifeblood of many organizations and a growing piece of the world economy, a well-timed denial-of-service attack can cause a great deal of damage. By bringing down servers that control sensitive machinery or other functions, these attacks could also present a real physical threat to life and limb. An attacker could cause the service denial by flooding a system with bogus traffic, or even purposely causing the server to crash. Countless denial-of-service attacks are in widespread use today, and can be found at http://packetstorm.securify.com/exploits/DoS. The most often used network-based denial-of-service attacks fall into two categories: malformed packet attacks and packet floods.

Malformed Packet Attacks

This type of attack usually involves one or two packets that are formatted in an unexpected way. Many vendor product implementations do not take into account all variations of user entries or packet types. If the software handles such errors poorly, the system may crash when it receives such packets. A classic example of this type of attack involves sending IP fragments to a system that overlap with each other (the fragment offset values are incorrectly set). Some unpatched Windows and Linux systems will crash when they encounter such packets. The teardrop attack is an example of a tool that exploits this IP fragmentation handling vulnerability. Other malformed packet attacks that exploit other weaknesses in TCP/IP implementations include the colorfully named WinNuke, Land, LaTierra, NewTear, Bonk, Boink, etc.

Packet Flood Attacks

Packet flood denial-of-service tools send a deluge of traffic to a system on the network, overwhelming its capability to respond to legitimate users. Attackers have devised numerous techniques for creating such floods, with the most popular being SYN floods, directed broadcast attacks, and distributed denial-of-service tools.

SYN flood tools initiate a large number of half-open connections with a system by sending a series of SYN packets. When any TCP connection is established, a three-way handshake occurs. The initiating system (usually

the client) sends a SYN packet to the destination to establish a sequence number for all packets going from source to destination in that session. The destination responds with a SYN-ACK packet, which acknowledges the sequence number for packets going from source to destination, and establishes an initial sequence number for packets going the opposite direction. The source completes the three-way handshake by sending an ACK to the destination. The three-way handshake is completed, and communication (actual data transfer) can occur.

SYN floods take advantage of a weakness in TCP's three-way handshake. By sending only spoofed SYN packets and never responding to the SYN-ACK, an attacker can exhaust a server's ability to maintain state of all the initiated sessions. With a huge number of so-called half-open connections, a server cannot handle any new, legitimate traffic. Rather than filling up all of the pipe bandwidth to a server, only the server's capacity to handle session initiations needs to be overwhelmed (in most network configurations, a server's ability to handle SYNs is lower than the total bandwidth to the site). For this reason, SYN flooding is the most popular packet flood attack. Other tools are also available that flood systems with ICMP and UDP packets, but they merely consume bandwidth, so an attacker would require a bigger connection than the victim to cut off all service.

Another type of packet flood that allows attackers to amplify their bandwidth is the directed broadcast attack. Often called a smurf attack, named after the first tool to exploit this technique, directed broadcast attacks utilize a third-party's network as an amplifier for the packet flood. In a smurf attack, the attacker locates a network on the Internet that will respond to a broadcast ICMP message (essentially a ping to the network's broadcast address). If the network is configured to allow broadcast requests and responses, all machines on the network will send a response to the ping. By spoofing the ICMP request, the attacker can have all machines on the third-party network send responses to the victim. For example, if an organization has 30 hosts on a single DMZ network connected to the Internet, an attacker can send a spoofed network broadcast ping to the DMZ. All 30 hosts will send a response to the spoofed address, which would be the ultimate victim. By sending repeated messages to the broadcast network, the attacker has amplified bandwidth by a factor of 30. Even an attacker with only a 56-kbps dial-up line could fill up a T1 line (1.54 Mbps) with that level of amplification. Other directed broadcast attack tools include Fraggle and Papasmurf.

A final type of denial-of-service that has received considerable press is the distributed denial-of-service attack. Essentially based on standard packet flood concepts, distributed denial-of-service attacks were used to cripple many major Internet sites in February 2000. Tools such as Trin00, Tribe Flood Network 2000 (TFN2K), and Stacheldraht all support this type of attack. To conduct a distributed denial-of-service attack, an attacker must find numerous vulnerable systems on the Internet. Usually, a remote buffer overflow attack (described below) is used to take over a dozen, a hundred, or even thousands of machines. Simple daemon processes, called zombies, are installed on these machines taken over by the attacker. The attacker communicates with this network of zombies using a control program. The control program is used to send commands to the hundreds or thousands of zombies, requesting them to take uniform action simultaneously.

The most common action to be taken is to simultaneously launch a packet flood against a target. While a traditional SYN flood would deluge a target with packets from one host, a distributed denial-of-service attack would send packets from large numbers of zombies, rapidly exhausting the capacity of even very high-bandwidth, well-designed sites. Many distributed denial-of-service attack tools support SYN, UDP, and ICMP flooding, smurf attacks, as well as some malformed packet attacks. Any one or all of these options can be selected by the attacker using the control program.

Denial-of-Service Attack Defenses

To defend against malformed packet attacks, system patches and security fixes must be regularly applied. Vendors frequently update their systems with patches to handle a new flavor of denial-of-service attack. An organization must have a program for monitoring vendor and industry security bulletins for security fixes, and a controlled method for implementing these fixes soon after they are announced and tested.

For packet flood attacks, critical systems should have underlying network architectures with multiple, redundant paths, eliminating a single point of failure. Furthermore, adequate bandwidth is a must. Also, some routers and firewalls support traffic flow control to help ease the burden of a SYN flood.

Finally, by configuring an Internet-accessible network appropriately, an organization can minimize the possibility that it will be used as a jumping-off point for smurf and distributed denial-of-service attacks. To

prevent the possibility of being used as a smurf amplifier, the external router or firewall should be configured to drop all directed broadcast requests from the Internet. To lower the chance of being used in a distributed denial-of-service attack, an organization should implement anti-spoof filters on external routers and firewalls to make sure that all outgoing traffic has a source IP address of the site. This egress filtering prevents an attacker from sending spoofed packets from a zombie or other denial-of-service tool located on the network. Antispoof ingress filters, which drop all packets from the Internet claiming to come from one's internal network, are also useful in preventing some denial-of-service attacks.

Stack-Based Buffer Overflows

Stack-based buffer overflow attacks are commonly used by an attacker to take over a system remotely across a network. Additionally, buffer overflows can be employed by local malicious users to elevate their privileges and gain superuser access to a system. Stack-based buffer overflow attacks exploit the way many operating systems handle their stack, an internal data structure used by running programs to store data temporarily. When a function call is made, the current state of the executing program and variables to be passed to the function are pushed on the stack. New local variables used by the function are also allocated space on the stack. Additionally, the stack stores the return address of the code calling the function. This return address will be accessed from the stack once the function call is complete. The system uses this address to resume execution of the calling program at the appropriate place. Exhibit 10.2 shows how a stack is constructed.

Most UNIX and all Windows systems have a stack that can hold data and executable code. Because local variables are stored on the stack when a function is called, poor code can be exploited to overrun the boundaries of these variables on the stack. If user input length is not examined by the code, a particular variable on the stack may exceed the memory allocated to it on the stack, overwriting all variables and even the return address for where execution should resume after the function is complete. This operation, called "smashing" the stack, allows an attacker to overflow the local variables to insert executable code and another return address on the stack. Exhibit 10.2 also shows a stack that has been smashed with a buffer overflow.

The attacker will overflow the buffer on the stack with machine-specific bytecodes that consist of executable commands (usually a shell routine), and a return pointer to begin execution of these inserted commands. Therefore, with very carefully constructed binary code, the attacker can actually enter information as a user into a program that consists of executable code and a new return address. The buggy program will not analyze the length of this input, but will place it on the stack, and actually begin to execute the attacker's code. Such vulnerabilities allow an attacker to break out of the application code, and access any system components with

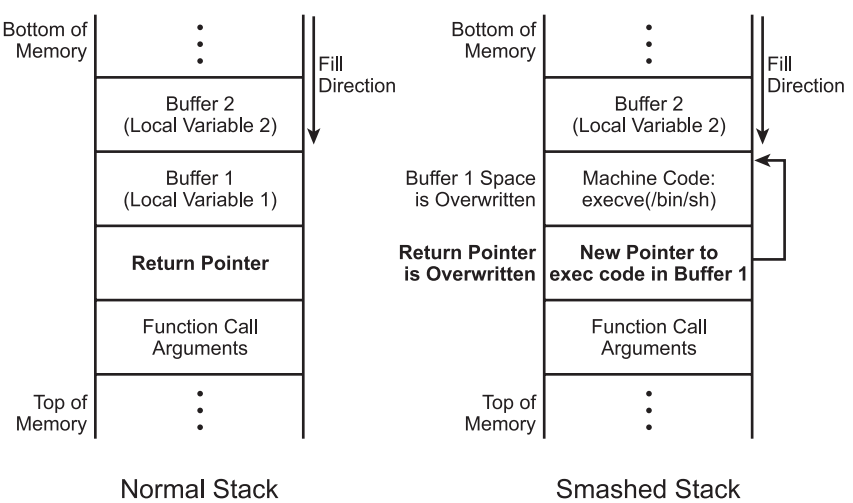


EXHIBIT 10.2 A normal stack and a stack with a buffer overflow.

the permissions of the broken program. If the broken program is running with superuser privileges (e.g., SUID root on a UNIX system), the attacker has taken over the machine with a buffer overflow.

Stack-Based Buffer Overflow Defenses

The most thorough defenses against buffer overflow attacks is to properly code software so that it cannot be used to smash the stack. All programs should validate all input from users and other programs, ensuring that it fits into allocated memory structures. Each variable should be checked (including user input, variables from other functions, input from other programs, and even environment variables) to ensure that allocated buffers are adequate to hold the data. Unfortunately, this ultimate solution is only available to individuals who write the programs and those with source code.

Additionally, security practitioners and system administrators should carefully control and minimize the number of SUID programs on a system that users can run and have permissions of other users (such as root). Only SUID programs with an explicit business need should be installed on sensitive systems.

Finally, many stack-based buffer overflow attacks can be avoided by configuring the systems to not execute code from the stack. Notably, Solaris and Linux offer this option. For example, to secure a Solaris system against stack-based buffer overflows, the following lines should be added:

```
/etc/system:

        set noexec_user_stack=1
        set noexec_user_stack_log=1
```

The first line will prevent execution on a stack, and the second line will log any attempt to do so. Unfortunately, some programs legitimately try to run code off the stack. Such programs will crash if this option is implemented. Generally, if the system is single purpose and needs to be secure (e.g., a Web server), this option should be used to prevent stack-based buffer overflow.

The Art and Science of Password Cracking

The vast majority of systems today authenticate users with a static password. When a user logs in, the password is transmitted to the system, which checks the password to make the decision whether to let the user log in. To make this decision, the system must have a mechanism to compare the user's input with the actual password. Of course, the system could just store all of the passwords locally and compare from this file. Such a file of cleartext passwords, however, would provide a very juicy target for an attacker. To make the target less useful for attackers, most modern operating systems use a one-way hash or encryption mechanism to protect the stored passswords. When a user types in a password, the system hashes the user's entry and compares it to the stored hash. If the two hashes match, the password is correct and the user can login.

Password cracking tools are used to attack this method of password protection. An attacker will use some exploit (often a buffer overflow) to gather the encrypted or hashed password file from a system (on a UNIX system without password shadowing, any user can read the hashed password file). After downloading the hashed password file, the attacker uses a password cracking tool to determine users' passwords. The cracking tool operates using a loop: it guesses a password, hashes or encrypts the password, and compares it to the hashed password from the stolen file. If the hashes match, the attacker has the password. If the hashes do not match, the loop begins again with another password guess.

Password cracking tools base their password guesses on a dictionary or a complete brute-force attack, attempting every possible password. Dozens of dictionaries are available online, in a multitude of languages, including English, French, German, Klingon, etc.

Numerous password-cracking tools are available. The most popular and full-functional password crackers include:

- John-the-Ripper, by Solar Designer, focuses on cracking UNIX passwords, and is available at http://www.openwall.com/john/.
- L0phtCrack, used to crack Windows NT passwords, is available at http://www.l0pht.com.

Password Cracking Defenses

The first defense against password cracking is to minimize the exposure of the encrypted/hashed password file. On UNIX systems, shadow password files should be used, which allow only the superuser to read the password file. On Windows NT systems, the SYSKEY feature available in NT 4.0 SP 3 and later should be installed and enabled. Furthermore, all backups and system recovery disks should be stored in physically secured locations and possibly even encrypted.

A strong password policy is a crucial element in ensuring a secure network. A password policy should require password lengths greater than eight characters, require the use of alphanumeric *and* special characters in every password, and force users to have passwords with mixed-case letters. Users must be aware of the issue of weak passwords and be trained in creating memorable, yet difficult-to-guess passwords.

To ensure that passwords are secure and to identify weak passwords, security practitioners should check system passwords on a periodic basis using password cracking tools. When weak passwords are discovered, the security group should have a defined procedure for interacting with users whose passwords can be easily guessed.

Finally, several software packages are available that prevent users from setting their passwords to easily guessed values. When a user establishes a new password, these filtering programs check the password to make sure that it is sufficiently complex and is not just a variation of the user name or a dictionary word. With this kind of tool, users are simply unable to create passwords that are easily guessed, eliminating a significant security issue. For filtering software to be effective, it must be installed on all servers where users establish passwords, including UNIX servers, Windows NT Primary and Back-up Domain Controllers, and Novell servers.

Backdoors

Backdoors are programs that bypass traditional security checks on a system, allowing an attacker to gain access to a machine without providing a system password and getting logged. Attackers install backdoors on a machine (or dupe a user into installing one for them) to ensure they will be able to gain access to the system at a later time. Once installed, most backdoors listen on special ports for incoming connections from the attacker across the network. When the attacker connects to the backdoor listener, the traditional userID and password or other forms of authentication are bypassed. Instead, the attacker can gain access to the system without providing a password, or by using a special password used only to enter the backdoor.

Netcat is an incredibly flexible tool written for UNIX by Hobbit and for Windows NT by Weld Pond (both versions are available at http://www. l0pht.com/~weld/netcat/). Among its numerous other uses, Netcat can be used to create a backdoor listener with a superuser-level shell on any TCP or UDP port. For Windows systems, an enormous number of backdoor applications are available, including Back Orifice 2000 (called BO2K for short, and available at http://www.bo2k.com) and hack-a-tack (available at http://www.hack-a-tack.com).

Backdoor Defenses

The best defense against backdoor programs is for system and security administrators to know what is running on their machines, particularly sensitive systems storing critical information or processing high-value transactions. If a process suddenly appears running as the superuser listening on a port, the administrator needs to investigate. Backdoors listening on various ports can be discovered using the **netstat –na** command on UNIX and Windows NT systems.

Additionally, many backdoor programs (such as BO2K) can be discovered by an anti-virus program, which should be installed on all users' desktops, as well as on servers throughout an organization.

Trojan Horses and RootKits

Another fundamental element of an attacker's toolchest is the Trojan horse program. Like the Trojan horse of ancient Greece, these new Trojan horses appear to have some useful function, but in reality are just disguising some malicious activity. For example, a user may receive an executable birthday card program in electronic mail. When the unsuspecting user activates the birthday card program and watches birthday cakes dance across

the screen, the program secretly installs a backdoor or perhaps deletes the users' hard drive. As illustrated in this example, Trojan horses rely on deception — they trick a user or system administrator into running them for their (apparent) usefulness, but their true purpose is to attack the user's machine.

Traditional Trojan Horses

A traditional Trojan horse is simply an independent program that can be run by a user or administrator. Numerous traditional Trojan horse programs have been devised, including:

- The familiar birthday card or holiday greeting e-mail attachment described above.
- A software program that claims to be able to turn CD-ROM readers into CD writing devices. Although this feat is impossible to accomplish in software, many users have been duped into downloading this "tool," which promptly deletes their hard drives upon activation.
- A security vulnerability scanner, WinSATAN. This tool claims to provide a convenient security vulnerability scan for system and security administrators using a Windows NT system. Unfortunately, an unsuspecting user running this program will also have a deleted hard drive.

Countless other examples exist. While conceptually unglamorous, traditional Trojan horses can be a major problem if users are not careful and run untrusted programs on their machines.

RootKits

A RootKit takes the concept of a Trojan horse to a much more powerful level. Although the name implies otherwise, RootKits do not allow an attacker to gain "root" (superuser) access to a system. Instead, RootKits allow an attacker who already has superuser access to keep that access by foiling all attempts of an administrator to detect the invasion. RootKits consist of an entire suite of Trojan horse programs that replace or patch critical system programs. The various tools used by administrators to detect attackers on their machines are routinely undermined with RootKits.

Most RootKits include a Trojan horse backdoor program (in UNIX, the */bin/login* routine). The attacker will install a new Trojan horse version of */bin/login,* overwriting the previous version. The RootKit */bin/login* routine includes a special backdoor userID and password so that the attacker can access the system at later times.

Additionally, RootKits include a sniffer and a program to hide the sniffer. An administrator can detect a sniffer on a system by running the **ifconfig** command. If a sniffer is running, the **ifconfig** output will contain the PROMISC flag, an indication that the Ethernet card is in promiscuous mode and therefore is sniffing. RootKit contains a Trojan horse version of **ifconfig** that does not display the PROMISC flag, allowing an attacker to avoid detection.

UNIX-based RootKits also replace other critical system executables, including **ps** and **du**. The **ps** command, emloyed by users and administrators to determine which processes are running, is modified so that an attacker can hide processes. The **du** command, which shows disk utilization, is altered so that the file space taken up by RootKit and the attacker's other programs can be masked.

By replacing programs like /bin/login, ifconfig, ps, du, and numerous others, these RootKit tools become part of the operating system itself. Therefore, RootKits are used to cover the eyes and ears of an administrator. They create a virtual world on the computer that appears benign to the system administrator, when in actuality, an attacker can log in and move around the system with impunity. RootKits have been developed for most major UNIX systems and Windows NT. A whole variety of UNIX RootKits can be found at http://packetstorm.securify.com/UNIX/penetration/rootkits, while an NT RootKit is available at http://www.rootkit.com.

A recent development in this arena is the release of kernel-level RootKits. These RootKits act at the most fundamental levels of an operating system. Rather than replacing application programs such as /bin/login and ifconfig, kernel-level RootKits actually patch the kernel to provide very low-level access to the system. These tools rely on the loadable kernel modules that many new UNIX variants support, including Linux and Solaris. Loadable kernel modules let an administrator add functionality to the kernel on-the-fly, without even rebooting the system. An attacker with superuser access can install a kernel-level RootKit that will allow for the remapping of execution of programs.

When an administrator tries to run a program, the Trojanized kernel will remap the execution request to the attacker's program, which could be a backdoor offering access or other Trojan horse. Because the kernel does the remapping of execution requests, this type of activity is very difficult to detect. If the administrator

attempts to look at the remapped file or check its integrity, the program will appear unaltered, because the program's image *is* unaltered. However, when executed, the unaltered program is skipped, and a malicious program is substituted by the kernel. Knark, written by Creed, is a kernel-level RootKit that can be found at http://packetstorm.securify.com/UNIX/penetration/rootkits.

Trojan Horses and RootKit Defenses

To protect against traditional Trojan horses, user awareness is key. Users must understand the risks associated with downloading untrusted programs and running them. They must also be made aware of the problems of running executable attachments in e-mail from untrusted sources.

Additionally, some traditional Trojan horses can be detected and eliminated by anti-virus programs. Every end-user computer system (and even servers) should have an effective and up-to-date anti-virus program installed.

To defend against RootKits, system and security administrators must use integrity checking programs for critical system files. Numerous tools are available, including the venerable Tripwire, that generate a hash of the executables commonly altered when a RootKit is installed. The administrator should store these hashes on a protected medium (such as a write-protected floppy disk) and periodically check the veracity of the programs on the machine with the protected hashes. Commonly, this type of check is done at least weekly, depending on the sensitivity of the machine. The administrator must reconcile any changes discovered in these critical system files with recent patches. If system files have been altered, and no patches were installed by the administrator, a malicious user or outside attacker may have installed a RootKit. If a RootKit is detected, the safest way to ensure its complete removal is to rebuild the entire operating system and even critical applications.

Unfortunately, kernel-level RootKits cannot be detected with integrity check programs because the integrity checker relies on the underlying kernel to do its work. If the kernel lies to the integrity checker, the results will not show the RootKit installation. The best defense against the kernel-level RootKit is a monolithic kernel that does not support loadable kernel modules. On critical systems (such as firewalls, Internet Web servers, DNS servers, mail servers, etc.), administrators should build the systems with complete kernels without support for loadable kernel modules. With this configuration, the system will prevent an attacker from gaining root-level access and patching the kernel in real-time.

Overall Defenses: Intrusion Detection and Incident Response Procedures

Each of the defensive strategies described in this chapter deals with particular tools and attacks. In addition to employing each of those strategies, organizations must also be capable of detecting and responding to an attack. These capabilities are realized through the deployment of intrusion detection systems (IDSs) and the implementation of incident response procedures.

IDSs act as burglar alarms on the network. With a database of known attack signatures, IDSs can determine when an attack is underway and alert security and system administration personnel. Acting as early warning systems, IDSs allow an organization to detect an attack in its early stages and minimize the damage that may be caused.

Perhaps even more important than IDSs, documented incident response procedures are among the most critical elements of an effective security program. Unfortunately, even with industry-best defenses, a sufficiently motivated attacker can penetrate the network. To address this possibility, an organization must have procedures defined in advance describing how the organization will react to the attack. These incident response procedures should specify the roles of individuals in the organization during an attack. The chain of command and escalation procedures should be spelled out in advance. Creating these items during a crisis will lead to costly mistakes.

Truly effective incident response procedures should also be multidisciplinary, not focusing only on information technology. Instead, the roles, responsibilities, and communication channels for the Legal, Human Resources, Media Relations, Information Technology, and Security organizations should all be documented and communicated. Specific members of these organizations should be identified as the core of a Security Incident Response Team (SIRT), to be called together to address an incident when one occurs. Additionally,

the SIRT should conduct periodic exercises of the incident response capability to ensure that team members are effective in their roles.

Additionally, with a large number of organizations outsourcing their information technology infrastructure by utilizing Web hosting, desktop management, e-mail, data storage, and other services, the extension of the incident response procedures to these outside organizations can be critical. The contract established with the outsourcing company should carefully state the obligations of the service provider in intrusion detection, incident notification, and participation in incident response. A specific service-level agreement for handling security incidents and the time needed to pull together members of the service company's staff in a SIRT should also be agreed upon.

Conclusions

While the number and power of these attack tools continues to escalate, system administrators and security personnel should not give up the fight. All of the defensive strategies discussed throughout this chapter boil down to doing a thorough and professional job of administering systems: know what is running on the system, keep it patched, ensure appropriate bandwidth is available, utilize IDSs, and prepare a Security Incident Response Team. Although these activities are not easy and can involve a great deal of effort, through diligence, an organization can keep its systems secured and minimize the chance of an attack. By employing intrusion detection systems and sound incident response procedures, even those highly sophisticated attacks that do get through can be discovered and contained, minimizing the impact on the organization. By creating an effective security program with sound defensive strategies, critical systems and information can be protected.

11

A New Breed of Hacker Tools and Defenses

Ed Skoudis, CISSP

The state-of-the-art in computer attack tools and techniques is rapidly advancing. Yes, we still face the tried-and-true, decades-old arsenal of traditional computer attack tools, including denial-of-service attacks, password crackers, port scanners, sniffers, and RootKits. However, many of these basic tools and techniques have seen a renaissance in the past couple of years, with new features and underlying architectures that make them more powerful than ever. Attackers are delving deep into widely used protocols and the very hearts of our operating systems. In addition to their growing capabilities, computer attack tools are becoming increasingly easy to use. Just when you think you have seen it all, a new and easy-to-use attack tool is publicly released with a feature that blows your socks off. With this constant increase in the sophistication and ease of use in attack tools, as well as the widespread deployment of weak targets on the Internet, we now live in the golden age of hacking.

The purpose of this chapter is to describe recent events in this evolution of computer attack tools. To create the best defenses for our computers, one must understand the capabilities and tactics of one's adversaries. To achieve this goal, this chapter describes several areas of advance among attack tools, including distributed attacks, active sniffing, and kernel-level RootKits, along with defensive techniques for each type of attack.

Distributed Attacks

One of the primary trends in the evolution of computer attack tools is the movement toward distributed attack architectures. Essentially, attackers are harnessing the distributed power of the Internet itself to improve their attack capabilities. The strategy here is pretty straightforward, perhaps deceptively so given the power of some of these distributed attack tools. The attacker takes a conventional computer attack and splits the work among many systems. With more and more systems collaborating in the attack, the attacker's chances for success increase. These distributed attacks offer several advantages to attackers, including:

- They may be more difficult to detect.
- They usually make things more difficult to trace back to the attacker.
- They may speed up the attack, lowering the time necessary to achieve a given result.
- They allow an attacker to consume more resources on a target.

So, where does an attacker get all of the machines to launch a distributed attack? Unfortunately, enormous numbers of very weak machines are readily available on the Internet. The administrators and owners of such systems do not apply security patches from the vendors, nor do they configure their machines securely, often just using the default configuration right out of the box. Poorly secured computers at universities, companies of all sizes, government institutions, homes with always-on Internet connectivity, and elsewhere are easy prey for an attacker. Even lowly skilled attackers can take over hundreds or thousands of systems around the globe with ease. These attackers use automated vulnerability scanning tools, including homegrown scripts and freeware tools such as the Nessus vulnerability scanner (http://www.nessus.org), among many others, to scan large swaths of the Internet. They scan indiscriminately, day in and day out, looking to take over vulnerable

systems. After taking over a suitable number of systems, the attackers will use these victim machines as part of the distributed attack against another target.

Attackers have adapted many classic computer attack tools to a distributed paradigm. This chapter explores many of the most popular distributed attack tools, including distributed denial-of-service attacks, distributed password cracking, distributed port scanning, and relay attacks.

Distributed Denial-of-Service Attacks

One of the most popular and widely used distributed attack techniques is the distributed denial-of-service (DDoS) attack. In a DDoS attack, the attacker takes over a large number of systems and installs a remotely controlled program called a zombie on each system. The zombies silently run in the background awaiting commands. An attacker controls these zombie systems using a specialized client program running on one machine. The attacker uses one client machine to send commands to the multitude of zombies, telling them to simultaneously conduct some action. In a DDoS attack, the most common action is to flood a victim with packets. When all the zombies are simultaneously launching packet floods, the victim machine will be suddenly awash in bogus traffic. Once all capacity of the victim's communication link is exhausted, no legitimate user traffic will be able to reach the system, resulting in a denial of service.

The DDoS attack methodology was in the spotlight in February 2000 when several high-profile Internet sites were hit with the attack. DDoS tools have continued to evolve, with new features that make them even nastier. The latest generation of DDoS attacks includes extensive spoofing capabilities, so that all traffic from the client to the zombies and from the zombies to the target has a decoy source address. Therefore, when a flood begins, the investigators must trace the onslaught back, router hop by router hop, from the victim to the zombies. After rounding up some of the zombies, the investigators must still trace from the zombies to the client, across numerous hops and multiple Internet service providers (ISPs). Furthermore, DDoS tools are employing encryption to mask the location of the zombies. In early generations of DDoS tools, most of the client software included a file with a list of network addresses for the zombies. By discovering such a client, an investigation team could quickly locate and eradicate the zombies. With the latest generation of DDoS tools, the list of network addresses at the client is strongly encrypted so that the client does not give away the location of the zombies.

Defenses against Distributed Denial-of-Service Attacks

To defend against any packet flood, including DDoS attacks, one must ensure that critical network connections have sufficient bandwidth and redundancy to eliminate simple attacks. If a network connection is mission critical, one should have at least a redundant T1 connection because all lower connection speeds can easily be flooded by an attacker.

While this baseline of bandwidth eliminates the lowest levels of attackers, one must face the fact that one will not be able to buy enough bandwidth to keep up with attackers who have installed zombies on a hundred or thousand systems and pointed them at your system as a target. If one's system's availability on the Internet is critical to the business, one must employ additional techniques for handling DDoS attacks. From a technological perspective, one may want to consider traffic shaping tools, which can help manage the number of incoming sessions so that one's servers are not overwhelmed. Of course, a large enough cadre of zombies flooding one's connection could even overwhelm traffic shapers. Therefore, one should employ intrusion detection systems (IDSs) to determine when an attack is underway. These IDSs act as network burglar alarms, listening to the network for traffic that matches common attack signatures stored in the IDS database. From a procedural perspective, one should have an incident response team on stand-by for such alarms from the IDS. For mission-critical Internet connections, one must have the cell phone and pager numbers for one's ISP's own incident response team. When a DDoS attack begins, one's incident response team must be able to quickly and efficiently marshal the forces of the ISP's incident response team. Once alerted, the ISP can deploy filters in their network to block an active DDoS attack upstream.

Distributed Password Cracking

Password cracking is another technique that has been around for many years and is now being leveraged in distributed attacks. The technique is based on the fact that most modern computing systems (such as UNIX and Windows NT) have a database containing encrypted passwords used for authentication. In Windows NT,

the passwords are stored in the SAM database. On UNIX systems, the passwords are located in the /etc/passwd or /etc/shadow files. When a user logs on to the system, the machine asks the user for a password, encrypts the value entered by the user, and compares the encrypted version of what the user typed with the stored encrypted password. If they match, the user is allowed to log in.

The idea behind password cracking is simple: steal an encrypted password file, guess a password, encrypt the guess, and compare the result to the value in the stolen encrypted password file. If the encrypted guess matches the encrypted password, the attacker has determined the password. If the two values do not match, the attacker makes another guess. Because user passwords are often predictable combinations of user IDs, dictionary words, and other characters, this technique is often very successful in determining passwords.

Traditional password cracking tools automate the guess-encrypt-compare loop to help determine passwords quickly and efficiently. These tools use variations of the user ID, dictionary terms, and brute-force guessing of all possible character combinations to create their guesses for passwords. The better password-cracking tools can conduct hybrid attacks, appending and prepending characters in a brute-force fashion to standard dictionary words. Because most passwords are simply a dictionary term with a few special characters tacked on at the beginning or end, the hybrid technique is extremely useful. Some of the best traditional password-cracking tools are L0phtCrack for Windows NT passwords (available at http://www.l0pht.com) and John the Ripper for a variety of password types, including UNIX and Windows NT (available at http://www.openwall.com).

When cracking passwords, speed rules. Tools that can create and check more password guesses in less time will result in more passwords recovered by the attacker. Traditional password cracking tools address this speed issue by optimizing the implementation of the encryption algorithm used to encrypt the guesses. Attackers can gain even more speed by distributing the password-cracking load across numerous computers. To more rapidly crack passwords, attackers will simultaneously harness hundreds or thousands of systems located all over the Internet to churn through an encrypted password file.

To implement distributed password cracking, an attacker can use a traditional password-cracking tool in a distributed fashion by simply dividing up the work manually. For example, consider a scenario in which an attacker wants to crack a password file with ten encrypted passwords. The attacker could break the file into ten parts, each part containing one encrypted password, and then distribute each part to one of ten machines. Each machine runs a traditional password-cracking tool to crack the one encrypted password assigned to that system. Alternatively, the attacker could load all ten encrypted passwords on each of the machines and configure each traditional password-cracking tool to guess a different set of passwords, focusing on a different part of a dictionary or certain characters in a brute-force attack.

Beyond manually splitting up the work and using a traditional password-cracking tool, several native distributed password-cracking tools have been released. These tools help to automate the spreading of the workload across several machines and coordinate the computing resources as the attack progresses. Two of the most popular distributed password-cracking tools are Mio-Star and Saltine Cracker, both available at http://packetstorm.securify.com/distributed

Defenses against Distributed Password Cracking

The defenses against distributed password cracking are really the same as those employed for traditional password cracking: eliminate weak passwords from your systems. Because distributed password cracking speeds up the cracking process, passwords need to be even more difficult to guess than in the days when nondistributed password cracking ruled. One must start with a policy that mandates users to establish passwords that are greater than a minimum length (such as greater than nine characters) and include numbers, letters, and special characters in each password. Users must be aware of the policy; thus, an awareness program emphasizing the importance of difficult-to-guess passwords is key. Furthermore, to help enforce a password policy, one may want to deploy password-filtering tools on one's authentication servers. When a user establishes a new password, these tools check the password to make sure it conforms to the password policy. If the password is too short, or does not include numbers, letters, and special characters, the user will be asked to select another password. The passfilt.dll program included in the Windows NT Resource Kit and the passwd+ program on UNIX systems implement this type of feature, as do several third-party add-on authentication products. One also may want to consider the elimination of standard passwords from very sensitive environments, using token-based access technologies.

Finally, security personnel should periodically run a password-cracking tool against one's own users' passwords to identify the weak ones before an attacker does. When weak passwords are found, there should be a defined and approved process for informing users that that they should select a better password. Be sure to

get appropriate permissions before conducting in-house password-cracking projects to ensure that management understands and supports this important security program. Not getting management approval could negatively impact one's career.

Distributed Port Scanning

Another attack technique that lends itself well to a distributed approach is the port scan. A port is an important concept in the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP), two protocols used by the vast majority of Internet services. Every server that receives TCP or UDP traffic from a network listens on one or more ports. These ports are like little virtual doors on a machine, where packets can go in or come out. The port numbers serve as addresses on a system where the packets should be directed. While an administrator can configure a network service to listen on any port, the most common services listen on well-known ports, so that client software knows where to send the packets. Web servers usually listen on TCP port 80, while Internet mail servers listen on TCP port 25. Domain Name Servers listen for queries on UDP port 53. Hundreds of other ports are assigned to various services in RFC 1700, a document available at http://www.ietf.org/rfc.html.

Port scanning is the process of sending packets to various ports on a target system to determine which ports have listening services. It is similar to knocking on the doors of the target system to see which ones are open. By knowing which ports are open on the target system, the attacker has a good idea of the services running on the machine. The attacker can then focus an attack on the services associated with these open ports. Furthermore, each open port on a target system indicates a possible entry point for an attacker. The attacker can scan the machine and determine that TCP port 25 and UDP port 53 are open. This result tells the attacker that the machine is likely a mail server and a DNS server. While there are a large number of traditional port-scanning tools available, one of the most powerful (by far) is the Nmap tool, available at http://www.insecure.org.

Because a port scan is often the precursor to a more in-depth attack, security personnel often use IDS tools to detect port scans as an early-warning indicator. Most IDSs include specific capabilities to recognize port scans. If a packet arrives from a given source going to one port, followed by another packet from the same source going to another port, followed by yet another packet for another port, the IDS can quickly correlate these packets to detect the scan. This traffic pattern is shown on the left-hand side of Exhibit 11.1, where port numbers are plotted against source network address. IDSs can easily spot such a scan, and ring bells and whistles (or send an e-mail to an administrator).

Now consider what happens when an attacker uses a distributed approach for conducting the scan. Instead of a barrage of packets coming from a single address, the attacker will configure many systems to participate in the scan. Each scanning machine will send only one or two packets and receive the results. By working together, the scanning machines can check all of the interesting ports on the target system and send their result to be correlated by the attacker. An IDS looking for the familiar pattern of the traditional port scan will not detect the attack. Instead, the pattern of incoming packets will appear more random, as shown on the right side of Exhibit 11.1. In this way, distributed scanning makes detection of attacks more difficult.

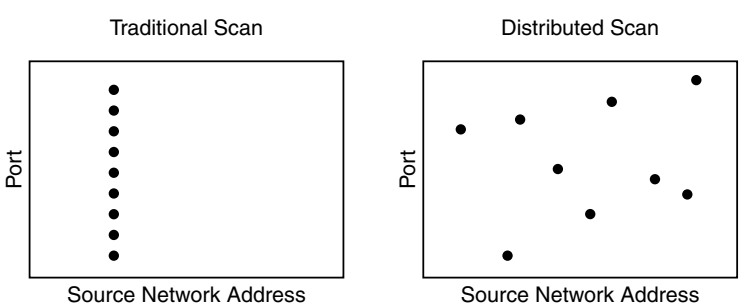


EXHIBIT 11.1 Traditional scans versus distributed scans.

Of course, an IDS system can still detect the distributed port scan by focusing on the destination address (i.e., the place where the packets are going) rather than the source address. If a number of systems suddenly sends packets to several ports on a single machine, an IDS can deduce that a port scan is underway. But the attacker has raised the bar for detection by conducting a distributed scan. If the distributed scan is conducted over a longer period of time (e.g., a week or a month), the chances of evading an IDS are quite good for an attacker. Distributed port scans are also much more difficult to trace back to an attacker because the scan comes from so many different systems, none of which are owned by the attacker.

Several distributed port-scanning tools are available. An attacker can use the descriptively named Phpdistributedportscanner, which is a small script that can be placed on Web servers to conduct a scan. Whenever attackers take over a PHP-enabled Web server, they can place the script on the server and use it to scan other systems. The attacker interacts with the individual scanning scripts running on the various Web servers using HTTP requests. Because everything is Web based, distributed port scans are quite simple to run. This scanning tool is available at http://www.digitaloffense.net:8000/phpDistributedPortS*canner/*. Other distributed port scanners tend to be based on a client/server architecture, such as Dscan (available at http://packetstorm.securify.com/distributed) and SIDEN (available at http://siden.sourceforge.net).

Defenses against Distributed Scanning

The best defense against distributed port scanning is to shut off all unneeded services on one's systems. If a machine's only purpose is to run a Web server that communicates via HTTP and HTTPS, the system should have only TCP port 80 and TCP port 443 open. If one does not need a mail server running on the same machine as the Web server, one should configure the system so that the mail server is deactivated. If the X Window system is not needed on the machine, turn it off. All other services should be shut off, which would close all other ports. One should develop a secure configuration document that provides a step-by-step process for all system administrators in an organization for building secure servers.

Additionally, one must ensure that IDS probes are kept up-to-date. Most IDS vendors distribute new attack signatures on a regular basis — usually once a month. When a new set of attack signatures is available, one should quickly test it and deploy it on the IDS probes so they can detect the latest batch of attacks.

Relay Attacks

A final distributed attack technique involves relaying information from machine to machine across the Internet to obscure the true source of the attack. As one can expect, most attackers do not want to get caught. By setting up extra layers of indirection between an attacker and the target, the attacker can avoid being apprehended. Suppose an attacker takes over half a dozen Internet-accessible machines located all over the world and wants to attack a new system. The attacker can set up packet redirector programs on the six systems. The first machine will forward any packets received on a given port to the second system. The second system would then forward them to the third system, and so on, until the new target is reached. Each system acts as a link in a relay chain for the attacker's traffic. If and when the attack is detected, the investigation team will have to trace the attack back through each relay point before finding the attacker.

Attackers often set up relay chains consisting of numerous systems around the globe. Additionally, to further foil investigators, attackers often try to make sure there is a great change in human language and geopolitical relations between the countries where the links of the relay chain reside. For example, the first relay may be in the United States, while the second may be in China. The third could be in India, while the fourth is in Pakistan. Finally, the chain ends in Iran for an attack against a machine back in the United States. At each stage of the relay chain, the investigators would have to contend with dramatic shifts in human language, less-than-friendly relations between countries, and huge law enforcement jurisdictional issues.

Relay attacks are often implemented using a very flexible tool called Netcat, which is available for UNIX at http://www.l0pht.com/users/10pht/ nc110.tgz, and for Windows NT at http://www.l0pht.com/~weld/netcat/. Another popular tool for creating relays is Redir, located at http:// oh.verio.com/~sammy/hacks.

Defenses against Relay Attacks

Because most of the action in a relay attack occurs outside an organization's own network, there is little one can do to prevent such attacks. One cannot really stop attackers from bouncing their packets through a bunch of machines before being attacked. One's best bet is to make sure that systems are secure by applying security

patches and shutting down all unneeded services. Additionally, it is important to cooperate with law enforcement officials in their investigations of such attacks.

Active Sniffing

Sniffing is another, older technique that is being rapidly expanded with new capabilities. Traditional sniffers are simple tools that gather traffic from a network. The user installs a sniffer program on a computer that captures all data passing by the computer's network interface, whether it is destined for that machine or another system. When used by network administrators, sniffers can capture errant packets to help troubleshoot the network. When used by attackers, sniffers can grab sensitive data from the network, such as passwords, files, e-mail, or anything else transmitted across the network.

Traditional Sniffing

Traditional sniffing tools are passive; they wait patiently for traffic to pass by on the network and gather the data when it arrives. This passive technique works well for some network types. Traditional Ethernet, a popular technology used to create a large number of local area networks (LANs), is a broadcast medium. Ethernet hubs are devices used to create traditional Ethernet LANs. All traffic sent to any one system on the LAN is broadcast to all machines on the LAN. A traditional sniffer can therefore snag any data going between other systems on the same LAN. In a traditional sniffing attack, the attacker takes over one system on the LAN, installs a sniffer, and gathers traffic destined for other machines on the same LAN. Some of the best traditional sniffers include Snort (available at http://www.snort.org) and Sniffit (available at http://reptile.rug.ac.be/~coder/sniffit/sniffit.html).

One of the commonly used defenses against traditional sniffers is a switched LAN. Contrary to an Ethernet hub, which acts as a broadcast medium, an Ethernet switch only sends data to its intended destination on the LAN. No other system on the LAN is able to see the data because the Ethernet switch sends the data to its appropriate destination and nowhere else. Another commonly employed technique to foil traditional sniffers is to encrypt data in transit. If the attackers do not have the encryption keys, they will not be able to determine the contents of the data sniffed from the network. Two of the most popular encryption protocols are the Secure Socket Layer (SSL), which is most often used to secure Web traffic, and Secure Shell (SSH), which is most often used to protect command-line shell access to systems.

Raising the Ante with Active Sniffing

While the defenses against passive sniffers are effective and useful to deploy, attackers have developed a variety of techniques for foiling them. These techniques, collectively known as active sniffing, involve injecting traffic into the network to allow an attacker to grab data that should otherwise be unsniffable. One of the most capable active sniffing programs available is Dsniff, available at http://www.monkey.org/~dugsong/dsniff/. One can explore Dsniff's various methods for sniffing by injecting traffic into a network, including MAC address flooding, spurious ARP traffic, fake DNS responses, and person-in-the-middle attacks against SSL.

MAC Addresses Flooding

An Ethernet switch determines where to send traffic on a LAN based on its media access control (MAC) address. The MAC address is a unique 48-bit number assigned to each Ethernet card in the world. The MAC address indicates the unique network interface hardware for each system connected to the LAN. An Ethernet switch monitors the traffic on a LAN to learn which plugs on the switch are associated with which MAC addresses. For example, the switch will see traffic arriving from MAC address AA:BB:CC:DD:EE:FF on plug number one. The switch will remember this information and send data destined for this MAC address only to the first plug on the switch. Likewise, the switch will autodetect the MAC addresses associated with the other network interfaces on the LAN and send the appropriate data to them.

One of the simplest, active sniffing techniques involves flooding the LAN with traffic that has bogus MAC addresses. The attacker uses a program installed on a machine on the LAN to generate packets with random MAC addresses and feed them into the switch. The switch will attempt to remember all of the MAC addresses as they arrive. Eventually, the switch's memory capacity will be exhausted with bogus MAC addresses. When their memory fills up, some switches fail into a mode where traffic is sent to all machines connected to the LAN. By using MAC flooding, therefore, an attacker can bombard a switch so that the switch will send all

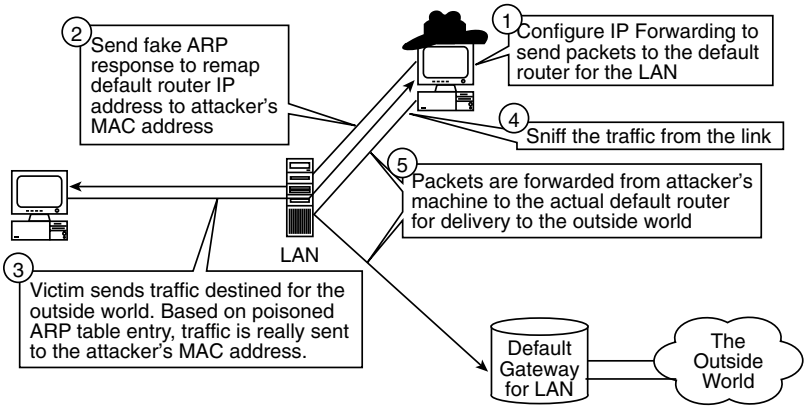


EXHIBIT 11.2 Active sniffing in a switched environment using gratuitous ARP messages. (Reprinted with permission. *CounterHack: A Step by Step Guide to Computer Attacks and Effective Defenses.* Copyright 2002, Prentice Hall PTR.)

traffic to all machines on the LAN. The attacker can then utilize a traditional sniffer to grab the data from the LAN.

Spurious ARP Traffic

While some switches fail under a MAC flood in a mode where they send all traffic to all systems on the LAN, other switches do not. During a flood, these switches remember the initial set of MAC addresses that were autodetected on the LAN, and utilize those addresses throughout the duration of the flood. The attacker cannot launch a MAC flood to overwhelm the switch. However, an attacker can still undermine such a LAN by injecting another type of traffic based on the Address Resolution Protocol (ARP).

ARP is used to map Internet Protocol (IP) addresses into MAC addresses on a LAN. When one machine has data to send to another system on the LAN, it formulates a packet for the destination's IP address; however, the IP address is just a configuration setting on the destination machine. How does the sending machine with the packet to deliver determine which hardware device on the LAN to send the packet to? ARP is the answer. Suppose a machine on the LAN has a packet that is destined for IP address 10.1.2.3. The machine with the packet will send an ARP request on the LAN, asking which network interface is associated with IP address 10.1.2.3. The machine with this IP address will transmit an ARP response, saying, in essence, "IP Address 10.1.2.3 is associated with MAC address AA:BB:CC:DD:EE:FF." When a system receives an ARP response, it stores the mapping of IP address to MAC address in a local table, called the ARP table, for future reference. The packet will then be delivered to the network interface with this MAC address. In this way, ARP is used to convert IP addresses into MAC addresses so that packets can be delivered to the appropriate network interface on the LAN. The results are stored in a system's ARP table to minimize the need for additional ARP traffic on the LAN.

ARP includes support for a capability called the "gratuitous ARP." With a gratuitous ARP, a machine can send an ARP response although no machine sent an ARP request. Most systems are thirsty for ARP entries in their ARP tables, to help improve performance on the LAN. In another form of active sniffing, an attacker utilizes faked gratuitous ARP messages to redirect traffic for sniffing a switched LAN, as shown in Exhibit 11.2. For the exhibit, the attacker's machine on the LAN is indicated by a black hat.

The steps of this attack, shown in Exhibit 11.2, are:

1. The attacker activates IP forwarding on the attacker's machine on the LAN. Any packets directed by the switch to the black-hat machine will be redirected to the default router for the LAN.
2. The attacker sends a gratuitous ARP message to the target machine. The attacker wants to sniff traffic sent from this machine to the outside world. The gratuitous ARP message will map the IP address of the default router for the LAN to the MAC address of the attacker's own machine. The target machine accepts this bogus ARP message and enters it into its ARP table. The target's ARP table is now poisoned with the false entry.

3. The target machine sends traffic destined for the outside world. It consults its ARP table to determine the MAC address associated with the default router for the LAN. The MAC address it finds in the ARP table is the attacker's address. All data for the outside world is sent to the attacker's machine.
4. The attacker sniffs the traffic from the line.
5. The IP forwarding activated in Step 1 redirects all traffic from the attacker's machine to the default router for the LAN. The default router forwards the traffic to the outside world. In this way, the victim will be able to send traffic to the outside world, but it will pass through the attacker's machine to be sniffed on its way out.

This sequence of steps allows the attacker to view all traffic to the outside world from the target system. Note that, for this technique, the attacker does not modify the switch at all. The attacker is able to sniff the switched LAN by manipulating the ARP table of the victim. Because ARP traffic and the associated MAC address information are only transmitted across a LAN, this technique only works if the attacker controls a machine on the same LAN as the target system.

Fake DNS Responses

A technique for injecting packets into a network to sniff traffic beyond a LAN involves manipulating the Domain Name System (DNS). While ARP is used on a LAN to map IP addresses to MAC addresses on a LAN, DNS is used across a network to map domain names into IP addresses. When a user types a domain name into some client software, such as entering www.skoudisstuff.com into a Web browser, the user's system sends out a query to a DNS server. The DNS server is usually located across the network on a different LAN. Upon receiving the query, the DNS server looks up the appropriate information in its configuration files and sends a DNS response to the user's machine that includes an IP address, such as 10.22.12.41. The DNS server maps the domain name to IP address for the user.

Attackers can redirect traffic by sending spurious DNS responses to a client. While there is no such thing as a gratuitous DNS response, an attacker that sits on any network between the target system and the DNS server can sniff DNS queries from the line. Upon seeing a DNS query from a client, the attacker can send a fake DNS response to the client, containing an IP address of the attacker's machine. The client software on the users' machine will send packets to this IP address, thinking that it is communicating with the desired server. Instead, the information is sent to the attacker's machine. The attacker can view the information using a traditional sniffer, and relay the traffic to its intended destination.

Person-in-the-Middle Attacks against SSL

Injecting fake DNS responses into a network is a particularly powerful technique when it is used to set up a person-in-the-middle attack against cryptographic protocols such as SSL, which is commonly used for secure Web access. Essentially, the attacker sends a fake DNS response to the target so that a new SSL session is established through the attacker's machine. As highlighted in Exhibit 11.3, the attacker uses a specialized relay tool to set up two cryptographic sessions: one between the client and the attacker, and the other between the attacker and the server. While the data moves between these sessions, the attacker can view it in cleartext.

The steps shown in Exhibit 11.3 include:

1. The attacker activates Dsniff's dnsspoof program, a tool that sends fake DNS responses. Additionally, the attacker activates another Dsniff tool called "webmitm," an abbreviation for Web Monkey-in-the-Middle. This tool implements a specialized SSL relay.
2. The attacker observes a DNS query from the victim machine and sends a fake DNS response. The fake DNS response contains the IP address of the attacker's machine.
3. The victim receives the DNS response and establishes an SSL session with the IP address included in the response.
4. The webmitm tool running on the attacker's machine established an SSL session with the victim machine, and another SSL session with the actual Web server that the client wants to access.
5. The victim sends data across the SSL connection. The webmitm tool decrypts the traffic from the SSL connection with the victim, displays it for the attacker, and encrypts the traffic for transit to the external Web server. The external Web server receives the traffic, not realizing that a person-in-the-middle attack is occurring.

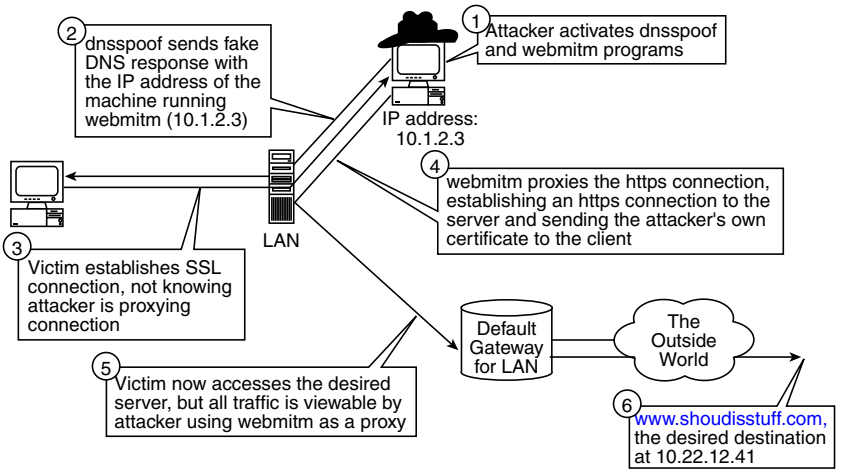


EXHIBIT 11.3 Injecting DNS responses to redirect and capture SSL traffic. (Reprinted with permission. *CounterHack: A Step by Step Guide to Computer Attacks and Effective Defenses.* Copyright 2002, Prentice Hall PTR.)

While this technique is quite effective, it does have one limitation from the attacker's point of view. When establishing the SSL connection between the victim and the attacker's machine, the attacker must send the victim an SSL digital certificate that belongs to the attacker. To decrypt all data sent from the target, the attacker must use his or her own digital certificate, and not the certificate from the actual destination Web server. When the victim's Web browser receives the bogus certificate from the attacker, it will display a warning message to the user. The browser will indicate that the certificate it was presented by the server was signed by a certificate authority that is not trusted by the browser. The browser then gives the user the option of establishing the connection by simply clicking on a button labeled "OK" or "Connect." Most users do not understand the warning messages from their browsers and will continue the connection without a second thought. The browser will be satisfied that it has established a secure connection because the user told it to accept the attacker's certificate. After continuing the connection, the attacker will be able to gather all traffic from the SSL session. In essence, the attacker relies on the fact that trust decisions about SSL certificates are left in the hands of the user.

The same basic technique works against the Secure Shell (SSH) protocol used for remote command-shell access. Dsniff includes a tool called sshmitm that can be used to set up a person-in-the-middle attack against SSH. Similar to the SSL attack, Dsniff establishes two SSH connections: one between the victim and the attacker, and another between the attacker and the destination server. Also, just as the Web browser complained about the modified SSL certificate, the SSH client will complain that it does not recognize the public key used by the SSH server. The SSH client will still allow the user, however, to override the warning and establish the SSH session so the attacker can view all traffic.

Defenses against Active Sniffing Techniques

Having seen how an attacker can grab all kinds of useful information from a network using sniffing tools, how can one defend against these attacks? First, whenever possible, encrypt data that gets transmitted across the network. Use secure protocols such as SSL for Web traffic, SSH for encrypted log-in sessions and file transfer, S/MIME for encrypted e-mail, and IPSec for network-layer encryption. Users must be equipped to apply these tools to protect sensitive information, both from a technology and an awareness perspective.

It is especially important that system administrators, network managers, and security personnel understand and use secure protocols to conduct their job activities. Never telnet to firewall, routers, sensitive servers, or public key infrastructure (PKI) systems! It is just too easy for an attacker to intercept one's password, which telnet transmits in cleartext. Additionally, pay attention to those warning messages from the browser and SSH client. Do not send any sensitive information across the network using an SSL session created with an untrusted certificate. If the SSH client warns that the server public key mysteriously changed, there is need to investigate.

Additionally, one really should consider getting rid of hubs because they are just too easy to sniff through. Although the cost may be higher than hubs, switches not only improve security, but also improve performance. If a complete migration to a switched network is impossible, at least consider using switched Ethernet on critical network segments, particularly the DMZ.

Finally, for networks containing very sensitive systems and data, enable port-level security on your switches by configuring each switch port with the specific MAC address of the machine using that port to prevent MAC flooding problems and fake ARP messages. Furthermore, for extremely sensitive networks, such as Internet DMZs, use static ARP tables on the end machines, hard coding the MAC addresses for all systems on the LAN. Port security on a switch and hard-coded ARP tables can be very difficult to manage because swapping components or even Ethernet cards requires updating the MAC addresses stored in several systems. For very sensitive networks such as Internet DMZs, this level of security is required and should be implemented.

The Proliferation of Kernel-Level RootKits

Just as attackers are targeting key protocols such as ARP and DNS at a very fundamental level, so too are they exploiting the heart of our operating systems. In particular, a great deal of development is underway on kernel-level RootKits. To gain a better understanding of kernel-level RootKits, one should first analyze their evolutionary ancestors, traditional RootKits.

Traditional RootKits

A traditional RootKit is a suite of tools that allows an attacker to maintain superuser access on a system. Once an attacker gets root-level control on a machine, the RootKit lets the attacker maintain that access. Traditional RootKits usually include a backdoor so the attacker can access the system, bypassing normal security controls. They also include various programs to let the attacker hide on the system. Some of the most fully functional traditional RootKits include Linux RootKit 5 (lrk5) and T0rnkit, which runs on Solaris and Linux. Both of these RootKits, as well as many others, are located at http://packetstorm.securify.com/UNIX/penetration/rootkits.

Traditional RootKits implement backdoors and hiding mechanisms by replacing critical executable programs included in the operating system. For example, most traditional RootKits include a replacement for the /bin/login program, which is used to authenticate users logging into a UNIX system. A RootKit version of /bin/login usually includes a backdoor password, known by the attacker, that can be used for root-level access of the machine. The attacker will write the new version of /bin/login over the earlier version, and modify the timestamps and file size to match the previous version.

Just as the /bin/login program is replaced to implement a backdoor, most RootKits include Trojan horse replacement programs for other UNIX tools used by system administrators to analyze the system. Many traditional RootKits include Trojan horse replacements for the ls command (which normally shows the contents of a directory). Modified versions of ls will hide the attacker's tools, never displaying their presence. Similarly, the attackers will replace netstat, a tool that shows which TCP and UDP ports are in use, with a modified version that lies about the ports used by an attacker. Likewise, many other system programs will be replaced, including ifconfig, du, and ps. All of these programs act like the eyes and ears of a system administrator. The attacker utilizes a traditional RootKit to replace these eyes and ears with new versions that lie about the attacker's presence on the system.

To detect traditional RootKits, many system administrators employ file system integrity checking tools, such as the venerable Tripwire program available at http://www.tripwire.com. These tools calculate cryptographically strong hashes of critical system files (such as /bin/login, ls, netstat, ifconfig, du, and ps) and store these digital fingerprints on a safe medium such as a write-protected floppy disk. Then, on a periodic basis (usually daily or weekly), the integrity-checking tool recalculates the hashes of the executables on the system and compares them with the stored values. If there is a change, the program has been altered, and the system administrator is alerted.

Kernel-Level RootKits

While traditional RootKits replace critical system executables, attackers have gone even further by implementing kernel-level RootKits. The kernel is the heart of most operating systems, controlling access to all resources,

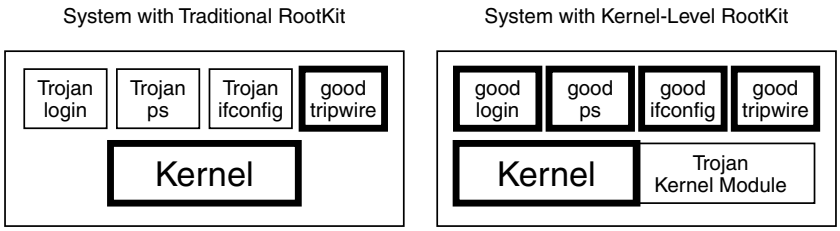


EXHIBIT 11.4 Traditional and kernel-level RootKits.

such as the disk, system processor, and memory. Kernel-level RootKits modify the kernel itself, rather than manipulating application-level programs like traditional RootKits. As shown on the left side of Exhibit 11.4, a traditional RootKit can be detected because a file system integrity tool such as Tripwire can rely on the kernel to let it check the integrity of application programs. When the application programs are modified, the good Tripwire program utilizes the good kernel to detect the Trojan horse replacement programs.

A kernel-level RootKit is shown on the right-hand side of Exhibit 11.4. While all of the application programs are intact, the kernel itself is rotten, facilitating backdoor access by the attacker and lying to the administrator about the attacker's presence on the system. Some of the most powerful kernel-level RootKits include Knark for Linux available at http://packetstorm. securify.com/UNIX/penetration/rootkits, Plasmoid's Solaris kernel-level RootKit available at http://www.infowar.co.uk/thc/slkm-1.0.html, and a Windows NT kernel-level RootKit available at http://www.rootkit.com.

While a large number of kernel-level RootKits have been released with a variety of features, the most popular capabilities of these tools include:

- *Execution redirection*. This capability intercepts a call to run a certain application and maps that call to run another application of the attacker's choosing. Consider a scenario involving the UNIX /bin/login routine. The attacker will install a kernel-level RootKit and leave the /bin/login file unaltered. All execution requests for /bin/login (which occur when anyone logs in to the system) will be mapped to the hidden file /bin/backdoorlog in. When a user tries to login, the /bin/backdoorlogin program will be executed, containing a backdoor password allowing for root-level access. However, when the system administrator runs a file integrity checker such as Tripwire, the standard /bin/login routine is analyzed. Only execution is redirected; one can look at the original file /bin/login and verify its integrity. This original routine is unaltered, so the Tripwire hash will remain the same.
- *File hiding*. Many kernel-level RootKits let an attacker hide any file in the file system. If any user or application looks for the file, the kernel will lie and say that the file is not present on the machine. Of course, the file is still on the system, and the attacker can access it when required.
- *Process hiding*. In addition to hiding files, the attacker can use the kernel-level RootKit to hide a running process on the machine.

Each of these capabilities is quite powerful by itself. Taken together, they offer an attacker the ability to completely transform the machine at the attacker's whim. The system administrator will have a view of the system created by the attacker, with everything looking intact. But in actuality, the system will be rotten to the core, quite literally. Furthermore, detection of kernel-level RootKits is often rather difficult because all access to the system relies on the attacker-modified kernel.

Kernel-Level RootKit Defenses

To stop attackers from installing kernel-level RootKits (or traditional RootKits, for that matter), one must prevent the attackers from gaining superuser access on one's systems in the first place. Without superuser access, an attacker cannot install a kernel-level RootKit. One must configure systems securely, disabling all unneeded services and applying all relevant security patches. Hardening systems and keeping them patched are the best preventative means for dealing with kernel-level RootKits.

Another defense involves deploying kernels that do not support loadable kernel modules (LKMs), a feature of some operating systems that allows the kernel to be dynamically modified. LKMs are often used to implement kernel-level RootKits. Linux kernels can be built without support for kernel modules. Unfortunately, Solaris systems up through and including Solaris 8 do not have the ability to disable kernel modules. For critical Linux

systems, such as Internet-accessible Web, mail, DNS, and FTP servers, one should build the kernels of such systems without the ability to accept LKMs. One will have eliminated the vast majority of these types of attacks by creating nonmodular kernels.

Conclusions

The arms race between computer defenders and computer attackers continues to accelerate. As attackers devise methods for widely distributed attacks and burrow deeper into our protocols and operating systems, we must work even more diligently to secure our systems. Do not lose heart, however. Sure, the defensive techniques covered in this chapter can be a lot of work. However, by carefully designing and maintaining systems, one can maintain a secure infrastructure.

12

Social Engineering: The Forgotten Risk

John Berti, CISSP and Marcus Rogers, Ph.D., CISSP

Information security practitioners are keenly aware of the major goals of information technology: availability, integrity, and confidentiality (the AIC triad). However, none of these goals is attainable if there is a weak link in the defense or security "chain." It has often been said that with information security, one is only as strong as one's weakest link. When we think of information and information technology security, we tend to focus collective attention on certain technical areas of this security chain. There are numerous reference sources available to information security practitioners that describe the latest operating system, application, or hardware vulnerabilities. Many companies have built their business plans and are able to survive based on being the first to discover these vulnerabilities and then provide solutions to the public and to the vendors themselves. It is quite obvious that the focus of the security industry has been primarily on the hardware, software, firmware, and the technical aspects of information security.

The security industry seems to have forgotten that computers and technology are merely tools, and that it is the human who is using, configuring, installing, implementing, and abusing these tools. Information security is more than just implementing a variety of technologically complex controls. It also encompasses dealing with the behavior or, more appropriately, the misbehavior of people. To be effective, information security must also address vulnerabilities within the "wetware," a term used to describe "people." One can spend all the money and effort one wants on technical controls and producing better, more secure code, but all of this is moot if our people give away the "keys to the kingdom." Recent research on network attacks clearly indicates that this is exactly what people are doing — albeit unintentionally. We seem to have done a good job instilling the notions of teamwork and cooperation in our workplace. So much so that in our eagerness to help out, we are falling prey to unscrupulous people who gain unauthorized access into systems through attacks categorized as "social engineering."

This chapter attempts to shed some light on social engineering by examining how this attack works, what are the common methods used, and how we can mitigate the risk of social engineering by proper education, awareness training, and other controls. This is not intended to be a "how-to" chapter, but rather a discussion of some of the details of this type of attack and how to prevent becoming a victim of social engineering. None of this information is secret; it is already well-known to certain sectors of society. Therefore, it is also important for information security professionals to be aware of social engineering and the security controls to mitigate the risk.

Defining Social Engineering

To understand what social engineering is, it is first important to clearly define what is being discussed. The term "social engineering" is not a new term. It comes from the field of social control. Social engineering can refer to the process of redefining a society — or more correctly, an engineering society — to achieve some desired outcome. The term can also refer to the process of attempting to change people's behavior in a predictable manner, usually in order to have them comply with some new system. It is the latter social

psychological definition of social engineering that is germane to this discussion. For our purposes, social engineering will refer to:

> Successful or unsuccessful attempts to influence a person(s) into either revealing information or acting in a manner that would result in unauthorized access, unauthorized use, or unauthorized disclosure, to an information system, network or data.

From definition, social engineering is somewhat synonymous with conning or deceiving someone. Using deception or conning a person is nothing new in the field of criminal activity; and despite its longevity, this kind of behavior is still surprisingly effective.

It would be very interesting at this point to include some information on the apparent size of the social engineering problem. Unfortunately, there is very little data to use for this purpose. Despite the frequent references to social engineering in the information security field, there has not been much direct discussion of this type of attack. The reasons for this vary; some within the field have suggested that social engineering attacks the intelligence of the victim and, as such, there is a reluctance to admit that it has occurred. Despite this reluctance, some of the most infamous computer criminals have relied more on social engineering to perpetrate their crimes than on any real technical ability. Why spend time researching and scanning systems looking for vulnerabilities and risk being detected when one can simply ask someone for a password to gain access? Most computer criminals, or any criminal for that matter, are opportunists. They look for the easy way into a system, and what could be easier than asking someone to let them in.

Why Does Social Engineering Work?

The success of social engineering attacks is primarily due to two factors: basic human nature and the business environment.

Human Nature

Falling victim to a social engineering attack has nothing to do with intelligence, and everything to do with being human, being somewhat naïve, and not having the proper mind set and training to deal with this type of attack. People, for the most part, are trusting and cooperative by nature. The field of social psychology has studied human interactions, both in groups and individually. These studies have concluded that almost anyone who is put in the right situation and who is dealing with a skilled person can be influenced to behave in a specific manner or divulge information he or she usually would not in other circumstances. These studies have also found that people who are in authority, or have the air of being in authority, easily intimidate other people.

For the most part, social engineering deals with individual dynamics as opposed to group dynamics, as the primary targets are help desks and administrative or technical support people, and the interactions are usually one-on-one but not necessarily face-to-face (i.e., the relationship is usually virtual in nature, either by phone or online). As discussed in this chapter, attackers tend to seek out individuals who display signs of being susceptible to this psychological attack.

Business Environment

Combined with human nature, the current business trend of mergers and acquisitions, rapid advances in technology, and the proliferation of wide area networking has made the business environment conducive to social engineering. In today's business world it is not uncommon to have never met the people one deals with on a regular basis, including those from one's own organization, let alone suppliers, vendors, and customers. Face-to-face human interaction is becoming even more rare with the widespread adoption of telecommuting technologies for employees. In today's marketplace, one can work for an organization and, apart from a few exceptions, rarely set foot in the office. Despite this layer of abstraction we have with people in our working environment, our basic trust in people, including those we have never actually met, has pretty much remained intact.

Businesses and organizations today have also become more service oriented than ever before. Employees are often rated on how well they contribute to a "team" environment, and on the level of service they provide to customers and other departments. It is rare to see a category on an evaluation that measures the degree to which someone used common sense, or whether an employee is conscious of security when performing his

or her duties. This is a paradigm that needs to change in order to deal effectively with the threat of social engineering.

Social Engineering Attacks

Social engineering attacks tend to follow a phased approach and, in most cases, the attacks are very similar to how intelligence agencies infiltrate their targets.

For the purpose of simplicity, the phases can be categorized as:

- Intelligence gathering
- Target selection
- The attack

Intelligence Gathering

One of the keys to a successful social engineering attack is information. It is surprisingly easy to gather sufficient information on an organization and its staff in order to sound like an employee of the company, a vendor representative, or in some cases a member of a regulatory or law enforcement body. Organizations tend to put far too much information on their Web sites as part of their marketing strategies. This information often describes or gives clues as to the vendors they may be dealing with, lists phone and e-mail directories, and indicates whether there are branch offices and, if so, where they are located. Some organizations even go as far as listing their entire organizational charts on their Web pages. All this information may be nice for potential investors, but it can also be used to lay the foundation for a social engineering attack.

Poorly thought-out Web sites are not the only sources of open intelligence. What organizations throw away can also be a source of important information. Going through an organization's garbage (also known as dumpster diving) can reveal invoices, correspondence, manuals, etc. that can assist an attacker in gaining important information. Several convicted computer criminals confessed to dumpster diving to gather information on their targets.

The attacker's goal at this phase is to learn as much information as possible in order to sound like he or she is a legitimate employee, contractor, vendor, strategic partner, or, in some cases, a law enforcement official.

Target Selection

Once the appropriate amount of information is collected, the attacker looks for noticeable weaknesses in the organization's personnel. The most common target is help desk personnel, as these professionals are trained to give assistance and can usually change passwords, create accounts, re-activate accounts, etc. In some organizations, the help desk function is contracted out to a third party with no real connection to the actual organization. This increases the chances of success, as the contracted third party would usually not know any of the organization's employees. The goal of most attackers is to either gather sensitive information or to get a foothold into a system. Attackers realize that once they have access, even at a guest level, it is relatively easy to increase their privileges, launch more destructive attacks, and hide their tracks.

Administrative assistants are the next most common victims. This is largely due to the fact that these individuals are privy to a large amount of sensitive information that normally flows between members of senior management. Administrative assistants can be used as either an attack point or to gather additional information regarding names of influential people in the organization. Knowing the names of the "movers and shakers" in an organization is valuable if there is a need to "name drop." It is also amazing how many administrative assistants know their executive managers' passwords. A number of these assistants routinely perform tasks for their managers that require their manager's account privileges (e.g., updating a spreadsheet, booking appointments in electronic calendars, etc.).

The Attack

The actual attack is usually based on what we would most commonly call a "con." These are broken down into three categories: (1) attacks that appeal to the vanity or ego of the victim, (2) attacks that take advantage of feelings of sympathy or empathy, and (3) attacks that are based on intimidation.

Ego Attacks

In the first type of attack — ego or vanity attacks — the attacker appeals to some of the most basic human characteristics. We all like to be told how intelligent we are and that we really know what we are doing or how to "fix" the company. Attackers will use this to extract information from their victims, as the attacker is a receptive audience for victims to display how much knowledge they have. The attacker usually picks a victim who feels under-appreciated and is working in a position that is beneath his or her talents. The attacker can usually sense this after only a brief conversation with the individual. Often, attackers using this type of an attack will call several different employees until they find the right one. Unfortunately, in most cases, the victim has no idea that he or she has done anything wrong.

Sympathy Attacks

In the second category of attacks, the attacker usually pretends to be a fellow employee (usually a new hire), a contractor, or a new employee of a vendor or strategic partner who just happens to be in a real jam and needs assistance to get some tasks done immediately. The importance of the intelligence phase becomes obvious here because attackers will have to create some level of trust with the victim that they are who they say they are. This is done by name dropping, using the appropriate jargon, or displaying knowledge of the organization. The attacker pretends that he or she is in a rush and must complete some task that requires access but cannot remember the account name or password, was inadvertently locked out, etc. A sense of urgency is usually part of the scenario because this provides an excuse for circumventing the required procedures that may be in place to regain access if the attacker was truly the individual he or she was pretending to be. It is human nature to sympathize or empathize with who the attacker is pretending to be; thus, in the majority of cases, the requests are granted. If the attacker fails to get the access or the information from one employee, he or she will just keep trying until a sympathetic ear is found, or until he or she realizes that the organization is getting suspicious.

Intimidation Attacks

In the third category, attackers pretend to be authority figures, either an influential person in the organization or, in some documented cases, law enforcement. Attackers will target a victim several levels within the organization below the level of the individual they are pretending to be. The attacker creates a plausible reason for making some type of request for a password reset, account change, access to systems, or sensitive information (in cases where the attacker is pretending to be a law enforcement official, the scenario usually revolves around some "hush-hush" investigation or national security issue, and the employee is not to discuss the incident). Again, the attackers will have done their homework and pretend to be someone with just enough power to intimidate the victim, but not enough to be either well-known to the victim or implausible for the scenario.[1] Attackers use scenarios in which time is of the essence and that they need to circumvent whatever the standard procedure is. If faced with resistance, attackers will try to intimidate their victims into cooperation by threatening sanctions against them.

Mitigating the Risk

Regardless of the type of social engineering attack, the success rate is alarmingly high. Many convicted computer criminals joke about the ease with which they were able to fool their victims into letting them literally "walk" into systems. The risk and impact of social engineering attacks are high. These attacks are often difficult to trace and, in some cases, difficult to identify. If the attacker has gained access via a legitimate account, in most cases the controls and alarms will never be activated because they have done nothing wrong as far as the system is concerned.

If social engineering is so easy to do, then how do organizations protect themselves against the risks of these attacks? The answer to this question is relatively simple but it entails a change in thinking on behalf of the entire organization. To mitigate the risk of social engineering, organizations need to effectively educate and train their staff on information security threats and how to recognize potential attacks. The control for these attacks can be found in education, awareness, training, and other controls, the discussion of which follows.

Social engineering concentrates on the weakest link in the information security chain — people. The fact that someone could persuade an employee to provide sensitive information means that the most secure systems

become vulnerable. The human part of any information security solution is the most essential. In fact, almost all information security solutions rely on the human element to a large degree. This means that this weakness — the human element — is universal, independent of hardware, software, platform, network, age of equipment, etc.

Many companies spend hundreds of thousands of dollars to ensure effective information security. This security is used to protect what the company regards as its most important assets, including information. Unfortunately, even the best security mechanisms can be bypassed when social engineering techniques are used. Social engineering uses very low-cost and low-technology means to overcome impediments posed by information security measures.

Protection against Social Engineering

To protect ourselves from the threat of social engineering, there must be a basic understanding of information security. In simple terms, information security can be defined as the protection of information against unauthorized disclosure, transfer, modification, or destruction, whether accidental or intentional. In general terms, information security denotes a state that a company reaches when its data and information, systems and services, are adequately protected against any type of threat. Information security protects information from a wide range of threats to ensure business continuity, minimize business damage, and maximize return on investment and business opportunities. Information security is about safeguarding a business money, image, and reputation — and perhaps its very existence.

Protection mechanisms usually fall into three categories, and it is important to note that to adequately protect an organization's information security assets, regardless of the type of threat, and including social engineering attacks, a combination of all three is required; that is:

1. Physical security
2. Logical (technical) security
3. Administrative security

Information security practitioners have long understood that a balanced approach to information security is required. That "balance" differs from company to company and is based on the system's vulnerabilities, threats, and information sensitivity, but in most instances will require a combination of all three elements mentioned above. Information security initiatives must be customized to meet the unique needs of the business. That is why it is very important to have an information security program that understands the needs of the corporation and can relate its information security needs to the goals and missions of the organization. Achieving the correct balance means implementing a variety of information security measures that fit into the three categories above, but implementing the correct balance so as to meet the organization's security requirements as efficiently and cost effectively as possible. Effective information security is the result of a process of identifying an organization's valued information assets; considering the range of potential risks to those assets; implementing effective policies to those specific conditions; and ensuring that those policies are properly developed, implemented, and communicated.

Physical Security

The physical security components are the easiest to understand and, arguably, the easiest to implement. Most people will think of keys, locks, alarms, and guards when they think of physical security. While these are by no means the only security precautions that need to be considered when securing information, they are a logical place to begin. Physical security, along with the other two (logical and administrative), are vital components and fundamental to most information security solutions. Physical security refers to the protection of assets from theft, vandalism, catastrophes, natural disasters, deliberate or accidental damage, and unstable environmental conditions such as electrical, temperature, humidity, and other such related problems. Good physical security requires efficient building and facility construction, emergency preparedness, reliable electrical power supplies, reliable and adequate climate control, and effective protection from both internal and external intruders.

Logical (Technical) Security

Logical security measures are those that employ a technical solution to protect the information asset. Examples include firewall systems, access control systems, password systems, and intrusion detection systems. These controls can be very effective, but usually rely on human element or interaction to work successfully. As mentioned, it is this human element that can be exploited rather easily.

Administrative Security

Administrative security controls are those that usually involve policies, procedures, guidelines, etc. Administrative security examples include information security policies, awareness programs, and background checks for new employees. These examples are administrative in nature, do not require a logical or technical solution to implement, but they all address the issue of information security.

Coverage

To be effective, information security must include the entire organization — from the top to the bottom, from the managers to the end users. Most importantly, the highest level of management present in any organization must endorse and support the idea and principles of information security. Everyone from top to bottom must understand the security principles involved and act accordingly. This means that high-level management must define, support, and issue the information security policy of the organization, which every person in the organization must then abide by. It also means that upper management must provide appropriate support, in the way of funding and resourcing, for information security. To summarize, a successful information security policy requires the leadership, commitment, and active participation of top-level management.

Critical information security strategies primarily rely on the appropriate and expected conduct on the part of personnel, and secondly on the use of technological solutions. This is why it is critical for all information security programs to address the threat of social engineering.

Securing against Social Engineering Attacks

Policies, Awareness, and Education

Social engineering attacks are very difficult to counter. The problem with countering social engineering attacks is that most logical security controls are ineffective as protection mechanisms. Because social engineering attacks target the human element, protective measures need to concentrate on the administrative portion of information security. An effective countermeasure is to have very good, established information security policies that are communicated across the entire organization. Policies are instrumental in forming a "rules of behavior" for employees. The second effective countermeasure is an effective user awareness program. When one combines these two administrative information security countermeasure controls effectively, the result is an integrated security program that everyone understands and believes is part of his or her own required job duties. From a corporate perspective, it is critical to convey this message to all employees, from top to bottom. The result will be an organization that is more vigilant at all levels, and an organization comprised of individuals who believe they are "contributing" to the well-being of the overall corporation. This is an important perception that greatly contributes to the employee satisfaction level. It also protects from the threat of disgruntled employees, another major concern of information security programs. It may be these disgruntled employees who willingly give sensitive information to unauthorized users, regardless of the social engineering methods.

Most people learn best from first-hand experience. Once it has been demonstrated that each individual is susceptible to social engineering attacks, these individuals tend to be more wary and aware. It is possible to make an organization more immune to social engineering attacks by providing a forum for discussions of other organizations' experiences.

Continued awareness is also very important. Awareness programs need to be repeated on a regular basis in order to re-affirm policies regarding social engineering. With today's technology, it is very easy to set up effective ways to communicate with one's employees on a regular basis. A good way to provide this type of forum is to

use an intranet Web site that will contain not only the organization's policies, but also safety tips and information regarding amusing social engineering stories. Amusing stories tend to get the point across better, especially if one takes into account that people love to hear about other people's misfortunes.

Recognition of "Good Catches"

Sometimes, the positive approach to recognition is the most effective one. If an employee has done the appropriate thing when it comes to an information security incident, acknowledge the good action and reward him or her appropriately. But do not stop there; let everyone else in the organization know. And as a result, the entire organization's preparedness will be improved.

Preparedness of Incident Response Teams

All companies should have the capability to deal effectively with what they may consider an incident. An incident can be defined as any event that threatens the company's livelihood. From an information security perspective, dealing with any outside threat (including social engineering) would be considered an incident. The goals of a well-prepared incident response team are to detect potential information security breaches and provide an effective and efficient means of dealing with the situation in a manner that reduces the potential impact to the corporation. A secondary but also very important goal would be to provide management with sufficient information to decide on an appropriate course of action. Having a team in place, comprised of knowledgeable individuals from key areas of the corporation who would be educated and prepared to respond to social engineering attacks, is a key aspect of an effective information security program.

Testing Readiness

Penetration testing is a method of examining the security controls of an organization from an outsider's point of view. To be effective, it involves testing all controls that prevent, track, and warn of internal and external intrusions. Companies that want to test their readiness against social engineering attacks can use this approach to reveal their weaknesses that may not have been previously evident. One must remember, however, that although penetration testing is one of the best ways to evaluate an organization's controls, it is only as effective as the efforts of the individuals who are performing the test.

Immediate Notification to Targeted Groups

If someone reports or discovers a social engineering attempt, one must notify personnel in similar areas. It is very important at this point to have a standard process and a quick procedure to do this. This is where a well-prepared incident response team can help. Assuming that a procedure is already in place, the incident response team can quickly deal with the problem and effectively remove it before any damage is done.

Apply Technology Where Possible

Other than making employees aware of the threat and providing guidance on how to handle both co-workers and others asking for information, there are no true solid methods for protecting information and employees from social engineering. However, a few options to consider may be the following:

- *Trace calls if possible.* Tracing calls may be an option, but only if one has the capability and is prepared for it. What one does not want in the midst of an attack is to ask oneself, "how do we trace a call?" Again, be prepared. Have some incident response procedures in place that will allow you to react accordingly in a very efficient manner.
- *Ensure good physical security.* As mentioned, good physical security is a must in order to provide efficient protection. There are many ways to effectively protect one's resources using the latest technology. This may mean using methods that employ biometrics or smart cards.
- *Mark sensitive documents according to data classification scheme.* If there is a well-established information classification scheme in place, it may protect one from revealing sensitive information in the event of a social engineering attack. For example, if someone is falling for an attack, and he or she pulls out a document that is marked "confidential," it may prevent him or her from releasing that information.

Similarly, if a file is electronically marked according to one's classification schemes, the same would apply.

Conclusion

Social engineering methods, when employed by an attacker, pose a serious threat to the security of information in any organization. There are far too many real-life examples of the success of this type of attack. However, following some of the basic principles of information systems security can mitigate the risk of social engineering. Policies need to be created in order to provide guidelines for the correct handling and release of information considered critical and sensitive within an organization. Information security awareness also plays a critical role. People need to be aware of the threats; and more importantly, they need to know exactly how to react in such an event. Explaining to employees the importance of information security and that there are people who are prepared to try and manipulate them to gain access to sensitive information is a wise first step in any defense plan. Simply forewarning people of possible attacks is often enough to make them alert to be able to spot them and react accordingly. The old saying that "knowledge is power" is true; or in this case, it increases security.

It is far easier to hack people than to hack some technically sound security device such as a firewall system. However, it is also takes much less effort to educate and prepare employees so that they can prevent and detect attempts at social engineering than it takes to properly secure that same firewall system. Organizations can no longer afford to have people as the weakest link in the information security chain.

Notes

1. CEOs are usually relatively well-known to employees, either from the media or from annual general meetings. Also, most CEOs would not be calling after-hours regarding a forgotten password. On the other hand, their assistant might.

13

Breaking News: The Latest Hacker Attacks and Defenses

Ed Skoudis, CISSP

Computer attackers continue to hone their techniques, getting ever better at undermining our systems and networks. As the computer technologies we use advance, these attackers find new and nastier ways to achieve their goals — unauthorized system access, theft of sensitive data, and alteration of information. This chapter explores some of the recent trends in computer attacks and presents tips for securing your systems. To create effective defenses, we need to understand the latest tools and techniques our adversaries are throwing at our networks. With that in mind, we will analyze four areas of computer attack that have received significant attention in the past year or so: wireless LAN attacks, active and passive operating system fingerprinting, worms, and sniffing backdoors.

Wireless LAN Attacks (War Driving)

In the past year, a very large number of companies have deployed wireless LANs, using technology based on the IEEE 802.11b protocol, informally known as *Wi-Fi*. Wireless LANs offer tremendous benefits from a usability and productivity perspective: a user can access the network from a conference room, while sitting in an associate's cubicle, or while wandering the halls. Unfortunately, wireless LANs are often one of the least secure methods of accessing an organization's network. The technology is becoming very inexpensive, with a decent access point costing less than U.S.$200 and wireless cards for a laptop or PC costing below U.S.$100. In addition to affordability, setting up an access point is remarkably simple (if security is ignored, that is). Most access points can be plugged into the corporate network and configured in a minute by a completely inexperienced user. Because of their low cost and ease of (insecure) use, wireless LANs are in rapid deployment in most networks today, whether upper management or even IT personnel realize or admit it. These wireless LANs are usually completely unsecure because the inexperienced employees setting them up have no idea of or interest in activating security features of their wireless LANs.

In our consulting services, we often meet with CIOs or Information Security Officers to discuss issues associated with information security. Given the widespread use of wireless LANs, we usually ask these upper-level managers what their organization is doing to secure its wireless infrastructure. We are often given the answer, "We don't have to worry about it because we haven't yet deployed a wireless infrastructure." After hearing that stock answer, we conduct a simple wireless LAN assessment (with the CIO's permission, of course). We walk down a hall with a wireless card, laptop, and wireless LAN detection software. Almost always we find renegade, completely unsecure wireless networks in use that were set up by employees outside of formal IT roles. The situation is similar to what we saw with Internet technology a decade ago. Back then, we would ask corporate officers what their organizations were doing to secure their Internet gateways. They would say that they did not have one, but we would quickly discover that the organization was laced with homegrown Internet connectivity without regard to security.

Network Stumbling, War Driving, and War Walking

Attackers have taken to the streets in their search for convenient ways to gain access to organizations' wireless networks. By getting within a few hundred yards of a wireless access point, an attacker can detect its presence and, if the access point has not been properly secured, possibly gain access to the target network. The process of searching for wireless access points is known in some circles as *network stumbling*. Alternatively, using an automobile to drive around town looking for wireless access points is known as *war driving*. As you might guess, the phrases *war walking* and even *war biking* have been coined to describe the search for wireless access points using other modes of transportation. I suppose it is only a matter of time before someone attempts *war hanggliding*.

When network stumbling, attackers set up a rig consisting of a laptop PC, wireless card, and antenna for discovering wireless access points. Additionally, a global positioning system (GPS) unit can help record the geographic location of discovered access points for later attack. Numerous software tools are available for this task as well. One of the most popular is NetStumbler (available at www.netstumbler.com), an easy-to-use GUI-based tool written by Marius Milner. NetStumbler runs on Windows systems, including Win95, 98, and 2000, and a PocketPC version called *Mini-Stumbler* has been released. For UNIX, several war-driving scripts have been released, with Wi-scan (available at www.dis.org/wl/) among the most popular.

This wireless LAN discovery process works because most access points respond, indicating their presence and their services set identifier (SSID) to a broadcast request from a wireless card. The SSID acts like a name for the wireless access point so that users can differentiate between different wireless LANs in close proximity. However, the SSID provides no real security. Some users think that a difficult-to-guess SSID will get them extra security. They are wrong. Even if the access point is configured not to respond to a broadcast request for an SSID, the SSIDs are sent in cleartext and can be intercepted.

In a recent war-driving trip in a taxi in Manhattan, an attacker discovered 455 access points in one hour. Some of these access points had their SSIDs set to the name of the company using the access point, gaining the attention of attackers focusing on juicy targets.

After discovering target networks, many attackers will attempt to get an IP address on the network, using the Dynamic Host Configuration Protocol (DHCP). Most wireless LANs freely give out addresses to anyone asking for them. After getting an address via DHCP, the attacker will attempt to access the LAN itself. Some LANs use the Wired Equivalent Privacy (WEP) protocol to provide cryptographic authentication and confidentiality. While WEP greatly improves the security of a wireless LAN, it has some significant vulnerabilities that could allow an attacker to determine an access point's keys. An attacker can crack WEP keys by gathering a significant amount of traffic (usually over 500 MB) using a tool such as Airsnort (available at airsnort.shmoo.com/).

Defending against Wireless LAN Attacks

So, how do you defend against wireless LAN attacks in your environment? There are several levels of security that you could implement for your wireless LAN, ranging from totally unsecure to a strong level of protection. Techniques for securing your wireless LAN include:

- *Set the SSID to an obscure value.* As described above, SSIDs are not a security feature and should not be treated as such. Setting the SSID to an obscure value adds very little from a security perspective. However, some access points can be configured to prohibit responses to SSID broadcast requests. If your access point offers that capability, you should activate it.
- *Use MAC address filtering.* Each wireless card has a unique hardware-level address called the media access control (MAC) address. A wireless access point can be configured so that it will allow traffic only from specific MAC addresses. While this MAC filtering does improve security a bit, it is important to note that an attacker can spoof wireless card MAC addresses.
- *Use WEP, with periodic rekeying.* While WEP keys can be broken using Airsnort, the technology significantly improves the security of a wireless LAN. Some vendors even support periodic generation of new WEP keys after a given timeout. If an attacker does crack a WEP key, it is likely that they break the old key, while a newer key is in use on the network. If your access points support dynamic rotating of WEP keys, such as Cisco's Aironet security solution, activate this feature.
- *Use a virtual private network (VPN).* Because SSID, MAC, and even WEP solutions have various vulnerabilities as highlighted above, the best method for securing wireless LANs is to use a VPN.

VPNs provide end-to-end security without regard to the unsecured wireless network used for transporting the communication. The VPN client encrypts all data sent from the PC before it gets sent into the air. The wireless access point simply collects encrypted streams of bits and forwards them to a VPN gateway before they can get access to the internal network. In this way, the VPN ensures that all data is strongly encrypted and authenticated before entering the internal network.

Of course, before implementing these technical solutions, you should establish specific policies for the use of wireless LANs in your environment. The particular wireless LAN security policies followed by an organization depend heavily on the need for security in that organization. The following list, which I wrote with John Burgess of Predictive Systems, contains recommended security policies that could apply in many organizations. This list can be used as a starting point, and pared down or built up to meet specific needs.

- All wireless access points/base stations connected to the corporate network must be registered and approved by the organization's computer security team. These access points/base stations are subject to periodic penetration tests and audits. Unregistered access points/ base stations on the corporate network are strictly forbidden.
- All wireless network interface cards (i.e., PC cards) used in corporate laptop or desktop computers must be registered with the corporate security team.
- All wireless LAN access must use corporate-approved vendor products and security configurations.
- All computers with wireless LAN devices must utilize a corporate-approved virtual private network (VPN) for communication across the wireless link. The VPN will authenticate users and encrypt all network traffic.
- Wireless access points/base stations must be deployed so that all wireless traffic is directed through a VPN device before entering the corporate network. The VPN device should be configured to drop all unauthenticated and unencrypted traffic.

While the policies listed above fit the majority of organizations, the policies listed below may or may not fit, depending on the technical level of employees and how detailed an organizations' security policy and guidelines are:

- The wireless SSID provides no security and should not be used as a password. Furthermore, wireless card MAC addresses can be easily gathered and spoofed by an attacker. Therefore, security schemes should not be based solely on filtering wireless MAC addresses because they do not provide adequate protection for most uses.
- WEP keys can be broken. WEP may be used to identify users, but only together with a VPN solution.
- The transmit power for access points/base stations near a building's perimeter (such as near exterior walls or top floors) should be turned down. Alternatively, wireless systems in these areas could use directional antennas to control signal bleed out of the building.

With these types of policies in place and a suitable VPN solution securing all traffic, the security of an organization's wireless infrastructure can be vastly increased.

Active and Passive Operating System Fingerprinting

Once access is gained to a network (through network stumbling, a renegade unsecured modem, or a weakness in an application or firewall), attackers usually attempt to learn about the target environment so they can hone their attacks. In particular, attackers often focus on discovering the operating system (OS) type of their targets. Armed with the OS type, attackers can search for specific vulnerabilities of those operating systems to maximize the effectiveness of their attacks.

To determine OS types across a network, attackers use two techniques: (1) the familiar, time-tested approach called active OS fingerprinting, and (2) a technique with new-found popularity, passive OS fingerprinting. We will explore each technique in more detail.

Active OS Fingerprinting

The Internet Engineering Task Force (IETF) defines how TCP/IP and related protocols should work. In an ever-growing list of Requests for Comment (RFCs), this group specifies how systems should respond when

specific types of packets are sent to them. For example, if someone sends a TCP SYN packet to a listening port, the IETF says that a SYN ACK packet should be sent in response. While the IETF has done an amazing job of defining how the protocols we use every day should work, it has not thoroughly defined every case of how the protocols should fail. In other words, the RFCs defining TCP/IP do not handle all of the meaningless or perverse cases of packets that can be sent in TCP/IP. For example, what should a system do if it receives a TCP packet with the code bits SYN-FIN-URG-PUSH all set? I presume such a packet means to SYNchronize a new connection, FINish the connection, do this URGently, and PUSH it quickly through the TCP stack. That is nonsense, and a standard response to such a packet has not been devised.

Because there is no standard response to this and other malformed packets, different vendors have built their OSs to respond differently to such bizarre cases. For example, a Cisco router will likely send a different response than a Windows NT server for some of these unexpected packets. By sending a variety of malformed packets to a target system and carefully analyzing the responses, an attacker can determine which OS it is running.

An active OS fingerprinting capability has been built into the Nmap port scanner (available at www.insecure.org/nmap). If the OS detection capability is activated, Nmap will send a barrage of unusual packets to the target to see how it responds. Based on this response, Nmap checks a user-customizable database of known signatures to determine the target OS type. Currently, this database houses over 500 known system types.

A more recent addition to the active OS fingerprinting realm is the Xprobe tool by Fyodor Yarochkin and Ofir Arkin. Rather than manipulating the TCP code bit options like Nmap, Xprobe focuses exclusively on the Internet Control Message Protocol (ICMP). ICMP is used to send information associated with an IP-based network, such as ping requests and responses, port unreachable messages, and instructions to quench the rate of packets sent. Xprobe sends between one and four specially crafted ICMP messages to the target system. Based on a very carefully constructed logic tree on the sending side, Xprobe can determine the OS type. Xprobe is stealthier than the Nmap active OS fingerprinting capability because it sends far fewer packets.

Passive OS Fingerprinting

While active OS fingerprinting involves sending packets to a target and analyzing the response, passive OS fingerprinting does not send any traffic while determining a target's OS type. Instead, passive OS fingerprinting tools include a sniffer to gather data from a network. Then, by analyzing the particular packet settings captured from the network and consulting a local database, the tool can determine what OS type sent that traffic. This technique is far stealthier than active OS fingerprinting because the attacker sends no data to the target machine. However, the attacker must be in a position to analyze traffic sent from the target system, such as on the same LAN or on a network where the target frequently sends packets.

One of the best passive OS fingerprinting tools is p0f (available at www.stearns.org/p0f/), originally written by Michal Zalewski and now maintained by William Stearns. P0f determines the OS type by analyzing several fields sent in TCP and IP traffic, including the rounded-up initial time-to-live (TTL), window size, maximum segment size, don't fragment flag, window scaling option, and initial packet size. Because different OSs set these initial values to varying levels, p0f can differentiate between 149 different system types.

Defending against Operating System Fingerprinting

To minimize the impact an attacker can have using knowledge of your OS types, you should have a defined program for notification, testing, and implementation of system patches. If you keep your systems patched with the latest security fixes, an attacker will be far less likely to compromise your machines even if they know which OS you are running. One or more people in your organization should have assigned tasks of monitoring vendor bulletins and security lists to determine when new patches are released. Furthermore, once patches are identified, they should be thoroughly but quickly tested in a quality assurance environment. After the full functionality of the tested system is verified, the patches should be rolled into production.

While a solid patching process is a must for defending your systems, you may also want to analyze some of the work in progress to defeat active OS fingerprinting. Gaël Roualland and Jean-Marc Saffroy wrote the IP personality patch for Linux systems, available at ippersonality.sourceforge.net/. This tool allows a system administrator to configure a Linux system running kernel version 2.4 so that it will have any response of the administrator's choosing for Nmap OS detection. Using this patch, you could make your Linux machine look like a Solaris system, a Macintosh, or even an old Windows machine during an Nmap scan. Although you may

not want to put such a patch onto your production systems due to potential interference with critical processes, the technique is certainly worth investigating.

To foil passive OS fingerprinting, you may want to consider the use of a proxy-style firewall. Proxy firewalls do not route packets, so all information about the OS type transmitted in the packet headers is destroyed by the proxy. Proxy firewalls accept a connection from a client, and then start a new connection to the server on behalf of that client. All packets on the outside of the firewall will have the OS fingerprints of the firewall itself. Therefore, the OS type of all systems inside the firewall will be masked. Note that this technique does not work for most packet filter firewalls because packet filters route packets and, therefore, transmit the fingerprint information stored in the packet headers.

Recent Worm Advances

A computer worm is a self-replicating computer attack tool that propagates across a network, spreading from vulnerable system to vulnerable system. Because they use one set of victim machines to scan for and exploit new victims, worms spread on an exponential basis. In recent times, we have seen a veritable zoo of computer worms with names like Ramen, L10n, Cheese, Code Red, and Nimda. New worms are being released at a dizzying rate, with a new generation of worm hitting the Internet every two to six months. Worm developers are learning lessons from the successes of each generation of worms and expanding upon them in subsequent attacks. With this evolutionary loop, we are rapidly approaching an era of super-worms. Based on recent advances in worm functions and predictions for the future, we will analyze the characteristics of the coming super-worms we will likely see in the next six months.

Rapidly Spreading Worms

Many of the worms released in the past decade have spread fairly quickly throughout the Internet. In July 2001, Code Red was estimated to have spread to 250,000 systems in about six hours. Fortunately, recent worms have had rather inefficient targeting mechanisms, a weakness that actually impeded their speeds. By randomly generating addresses and not taking into account the accurate distribution of systems in the Internet address space, these worms often wasted time looking for nonexistent systems or scanning machines that were already conquered.

After Code Red, several articles appeared on the Internet describing more efficient techniques for rapid worm distribution. These articles, by Nicholas C. Weaver and the team of Stuart Staniford, Gary Grim, and Roelof Jonkman, described the hypothetical Warhol and Flash worms, which theoretically could take over all vulnerable systems on the Internet in 15 minutes or even less. Warhol and Flash, which are only mathematical models and not actual worms (yet), are based on the idea of fast-forwarding through an exponential spread. Looking at a graph of infected victims over time for a conventional worm, a hockey-stick pattern appears. Things start out slowly as the initial victims succumb to the worm. Only after a critical mass of victims succumbs to the attack does the worm rapidly spread. Warhol and Flash jump past this initial slow spread by prescanning the Internet for vulnerable systems. Through automated scanning techniques from static machines, an attacker can find 100,000 or more vulnerable systems before ever releasing the worm. The attacker then loads these known vulnerable addresses into the worm. As the worm spreads, the addresses of these prescanned vulnerable systems would be split up among the segments of the worm propagating across the network. By using this initial set of vulnerable systems, an attacker could easily infect 99 percent of vulnerable systems on the Internet in less than an hour. Such a worm could conquer the Internet before most people have even heard of the problem.

Multi-Platform Worms

The vast majority of worms we have seen to date focused on a single platform, often Windows or Linux. For example, Nimda simply ripped apart as many Microsoft products as it could, exploiting Internet Explorer, the IIS Web server, Outlook, and Windows file sharing. While it certainly was challenging, Nimda's Windows-centric approach actually limited its spread. The security community implemented defenses by focusing on repairing Windows systems.

While single-platform worms can cause trouble, be on the lookout for worms that are far less discriminating from a platform perspective. New worms will contain exploits for Windows, Solaris, Linux, BSD, HP-UX, AIX, and other operating systems, all built into a single worm. Such worms are even more difficult to eradicate because security personnel and system administrators will have to apply patches in a coordinated fashion to many types of machines. The defense job will be more complex and require more time, allowing the worm to cause more damage.

Morphing and Disguised Worms

Recent worms have been relatively easy to detect. Once spotted, the computer security community has been able to quickly determine their functionalities. Once a worm has been isolated in the lab, some brilliant folks have been able to rapidly reverse-engineer each worm's operation to determine how best to defend against it.

In the very near future, we will face new worms that are far stealthier and more difficult to analyze. We will see polymorphic worms, which change their patterns every time they run and spread to a new system. Detection becomes more difficult because the worm essentially recodes itself each time it runs. Additionally, these new worms will encrypt or otherwise obscure much of their own payloads, hiding their functionalities until a later time. Reverse-engineering to determine the worm's true functions and purpose will become more difficult because investigators will have to extract the crypto keys or overcome the obfuscation mechanisms before they can really figure out what the worm can do. This time lag for the analysis will allow the worm to conquer more systems before adequate defenses are devised.

Zero-Day Exploit Worms

The vast majority of worms encountered so far are based on old, off-the-shelf exploits to attack systems. Because they have used old attacks, a patch has been readily available for administrators to fix their machines quickly after infection or to prevent infection in the first place. Using our familiar example, Code Red exploited systems using a flaw in Microsoft's IIS Web server that had been known for over a month and for which a patch had already been published.

In the near future, we are likely going to see a worm that uses brand-new exploits for which no patch exists. Because they are brand new, such attacks are sometimes referred to as *zero-day exploits*. New vulnerabilities are discovered practically every day. Oftentimes, these problems are communicated to a vendor, who releases a patch. Unfortunately, these vulnerabilities are all — too easy to discover, and it is only a matter of time before a worm writer discovers a major hole and first devises a worm that exploits it. Only after the worm has propagated across the Internet will the computer security community be capable of analyzing how it spreads so that a patch can be developed.

More Damaging Attacks

So far, worms have caused damage by consuming resources and creating nuisances. The worms we have seen to date have not really had a malicious payload. Once they take over hundreds of thousands of systems, they simply continue to spread without actually doing something nasty. Do not get me wrong; fighting Code Red and Nimda consumed much time and many resources. However, these attacks did not really do anything *beyond* simply consuming resources.

Soon, we may see worms that carry out some plan once they have spread. Such a malicious worm may be released in conjunction with a terrorist attack or other plot. Consider a worm that rapidly spreads using a zero-day exploit and then deletes the hard drives of ten million victim machines. Or, perhaps worse, a worm could spread and then transfer the financial records of millions of victims to a country's adversaries. Such scenarios are not very far-fetched, and even nastier ones could be easily devised.

Worm Defenses

All of the pieces are available for a moderately skilled attacker to create a truly devastating worm. We may soon see rapidly spreading, multi-platform, morphing worms using zero-day exploits to conduct very damaging attacks. So, what can you do to get ready? You need to establish both reactive and proactive defenses.

Incident Response Preparation

From a reactive perspective, your organization must establish a capability for determining when new vulnerabilities are discovered, as well as rapidly testing patches and moving them into production. As described above, your security team should subscribe to various security mailing lists, such as Bugtraq (available at www.securityfocus.com), to help alert you to such vulnerabilities and the release of patches. Furthermore, you must create an incident response team with the skills and resources necessary to discover and contain a worm attack.

Vigorously Patch and Harden Your Systems

From the proactive side, your organization must carefully harden your systems to prevent attacks. For each platform type, your organization should have documentation describing to system administrators how to build the machine to prevent attacks. Furthermore, you should periodically test your systems to ensure they are secure.

Block Unnecessary Outbound Connections

Once a worm takes over a system, it attempts to spread by making outgoing connections to scan for other potential victims. You should help stop worms in their tracks by severely limiting all outgoing connections on your publicly available systems (such as your Web, DNS, e-mail, and FTP servers). You should use a border router or external firewall to block all outgoing connections from such servers, unless there is a specific business need for outgoing connections. If you do need some outgoing connections, allow them only to those IP addresses that are absolutely critical. For example, your Web server needs to send responses to users requesting Web pages, of course. But does your Web server ever need to *initiate* connections to the Internet? Likely, the answer is no. So, do yourself and the rest of the Internet a favor by blocking such outgoing connections from your Internet servers.

Nonexecutable System Stack Can Help Stop Some Worms

In addition to overall system hardening, one particular step can help stop many worms. A large number of worms utilize buffer overflow exploits to compromise their victims. By sending more data than the program developer allocated space for, a buffer overflow attack allows an attacker to get code entered as user input to run on the target system. Most operating systems can be inoculated against simple stack-based buffer overflow exploits by being configured with nonexecutable system stacks. Keep in mind that nonexecutable stacks can break some programs (so test these fixes before implementing them), and they do not provide a bulletproof shield against all buffer overflow attacks. Still, preventing the execution of code from the stack will stop a huge number of both known and as-yet-undiscovered vulnerabilities in their tracks. Up to 90 percent of buffer overflows can be prevented using this technique. To create a nonexecutable stack on a Linux system, you can use the free kernel patch at www.openwall.com/linux. On a Solaris machine, you can configure the system to stop execution of code from the stack by adding the following lines to the/etc/system file:

```
set noexec_user_stack = 1
set noexec_user_stack_log = 1
```

On a Windows NT/2000 machine, you can achieve the same goal by deploying the commercial program SecureStack, available at www.securewave.com.

Sniffing Backdoors

Once attackers compromise a system, they usually install a backdoor tool to allow them to access the machine repeatedly. A backdoor is a program that lets attackers access the machine on their own terms. Normal users are required to type in a password or use a cryptographic token; attackers use a backdoor to bypass these normal security controls. Traditionally, backdoors have listened on a TCP or UDP port, silently waiting in the background for a connection from the attacker. The attacker uses a client tool to connect to these backdoor servers on the proper TCP or UDP port to issue commands.

These traditional backdoors can be discovered by looking at the listening ports on a system. From the command prompt of a UNIX or Windows NT/2000/XP machine, a user can type "netstat-na" to see which TCP and UDP ports on the local machine have programs listening on them. Of course, normal usage of a machine will cause some TCP and UDP ports to be listening, such as TCP port 80 for Web servers, TCP port 25 for mail servers, and UDP port 53 for DNS servers. Beyond these expected ports based on specific server

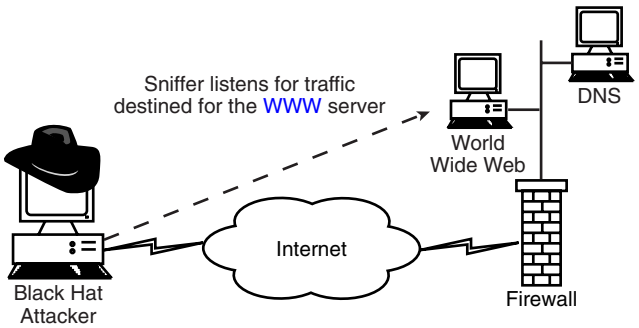


EXHIBIT 13.1 A promiscuous sniffing backdoor.

types, a suspicious port turned up by the `netstat` command could indicate a backdoor listener. Alternatively, a system or security administrator could remotely scan the ports of the system, using a port-scanning tool such as Nmap (available at www.insecure.org/nmap). If Nmap's output indicates an unexpected listening port, an attacker may have installed a backdoor.

Because attackers know that we are looking for their illicit backdoors listening on ports, a major trend in the attacker community is to avoid listening ports altogether for backdoors. You may ask, "How can they communicate with their backdoors if they aren't listening on a port?" To accomplish this, attackers are integrating sniffing technology into their backdoors to create sniffing backdoors. Rather than configuring a process to listen on a port, a sniffing backdoor uses a sniffer to grab traffic from the network. The sniffer then analyzes the traffic to determine which packets are supposed to go to the backdoor. Instead of listening on a port, the sniffer employs pattern matching on the network traffic to determine what to scoop up and pass to the backdoor. The backdoor then executes the commands and sends responses to the attacker. An excellent example of a sniffing backdoor is the Cd00r program written by FX. Cd00r is available at http://www. phenoelit.de/stuff/cd00r.c.

There are two general ways of running a sniffing backdoor, based on the mode used by the sniffer program to gather traffic: the so-called nonpromiscuous and promiscuous modes. A sniffer that puts an Ethernet interface in promiscuous mode gathers all data from the LAN without regard to the actual destination address of the traffic. If the traffic passes by the interface, the Ethernet card in promiscuous mode will suck in the traffic and pass it to the backdoor. Alternatively, a nonpromiscuous sniffer gathers traffic destined only for the machine on which the sniffer runs. Because these differences in sniffer types have significant implications on how attackers can use sniffing backdoors, we will explore nonpromiscuous and promiscuous backdoors separately below.

Nonpromiscuous Sniffing Backdoors

As their name implies, nonpromiscuous sniffing backdoors do not put the Ethernet interface into promiscuous mode. The sniffer sees only traffic going to and from the single machine where the sniffing backdoor is installed. When attackers use a nonpromiscuous sniffing backdoor, they do not have to worry about a system administrator detecting the interface in promiscuous mode.

In operation, the nonpromiscuous backdoor scours the traffic going to the victim machine looking for specific ports or other fields (such as a cryptographically derived value) included in the traffic. When the special traffic is detected, the backdoor wakes up and interacts with the attacker.

Promiscuous Sniffing Backdoors

By putting the Ethernet interface into promiscuous mode to gather all traffic from the LAN, promiscuous sniffing backdoors can make an investigation even more difficult. To understand why, consider the scenario shown in Exhibit 13.1. This network uses a tri-homed firewall to separate the DMZ and internal network from the Internet. Suppose an attacker takes over the Domain Name System (DNS) server on the DMZ and installs a promiscuous sniffing backdoor. Because this backdoor uses a sniffer in promiscuous mode, it can gather all

traffic from the LAN. The attacker configures the sniffing backdoor to listen in on all traffic with a destination address of the Web server (not the DNS server) to retrieve commands from the attacker to execute. In our scenario, the attacker does not install a backdoor or any other software on the Web server. Only the DNS server is compromised.

Now the attacker formulates packets with commands for the backdoor. These packets are all sent with a destination address of the Web server (*not* the DNS server). The Web server does not know what to do with these commands, so it will either discard them or send a RESET or related message to the attacker. However, the DNS server with the sniffing backdoor will see the commands on the LAN. The sniffer will gather these commands and forward them to the backdoor where they will be executed. To further obfuscate the situation, the attacker can send all responses from the backdoor using the spoofed source address of the Web server.

Given this scenario, consider the dilemma faced by the investigator. The system administrator or an intrusion detection system complains that there is suspicious traffic going to and from the Web server. The investigator conducts a detailed and thorough analysis of the Web server. After a painstaking process to verify the integrity of the applications, operating system programs, and kernel on the Web server machine, the investigator determines that this system is intact. Yet backdoor commands continue to be sent to this machine. The investigator would only discover what is really going on by analyzing other systems connected to the LAN, such as the DNS server. The investigative process is significantly slowed down by the promiscuous sniffing backdoor.

Defending against Sniffing Backdoor Attacks

It is important to note that the use of a switch on the DMZ network between the Web server and DNS server does not eliminate this dilemma. As described in Chapter 11, attackers can use active sniffers to conduct ARP cache poisoning attacks and successfully sniff a switched environment. An active sniffer such as Dsniff (available at http://www.monkey.org/~dugsong/dsniff/) married to a sniffing backdoor can implement this type of attack in a switched environment.

So if a switch does not eliminate this problem, how can you defend against this kind of attack? First, as with most backdoors, system and security administrators must know what is supposed to be running on their systems, especially processes running with root or system-level privileges. Keeping up with this information is not a trivial task, but it is especially important for all publicly available servers such as systems on a DMZ. If a security or system administrator notices a new process running with escalated privileges, the process should be investigated immediately. Tools such as lsof for UNIX (available at ftp://vic.cc.purdue.edu/pub/tools/ unix/lsof/) or Inzider for Windows NT/2000 (available at http://ntsecurity. nu/toolbox/inzider/) can help to indicate the files and ports used by any process. Keep in mind that most attackers will not name their backdoors "cd00r" or "backdoor," but instead will use less obvious names to camouflage their activities. In my experience, attackers like to name their backdoors "SCSI" or "UPS" to prevent a curious system administrator from questioning or shutting off the attackers' processes.

Also, while switches do not eliminate attacks with sniffers, a switched environment can help to limit an attacker's options, especially if it is carefully configured. For your DMZs and other critical networks, you should use a switch and hard-code all ARP entries in each host on the LAN. Each system on your LAN has an ARP cache holding information about the IP and MAC addresses of other machines on the LAN. By hard-coding all ARP entries on your sensitive LANs so that they are static, you minimize the possibility of ARP cached poisoning. Additionally, implement port-level security on your switch so that only specific Ethernet MAC addresses can communicate with the switch.

Conclusions

The computer underground and information security research fields remain highly active in refining existing methods and defining completely new ways to attack and compromise computer systems. Advances in our networking infrastructures, especially wireless LANs, are not only giving attackers new avenues into our systems, but they are also often riddled with security vulnerabilities. With this dynamic environment, defending against attacks is certainly a challenge. However, these constantly evolving attacks can be frustrating and exciting at the same time, while certainly providing job security to solid information security practitioners. While we need to work diligently in securing our systems, our reward is a significant intellectual challenge and decent employment in a challenging economy.

14

Counter-Economic Espionage

Craig A. Schiller, CISSP

> Today's economic competition is global. The conquest of markets and technologies has replaced former territorial and colonial conquests. We are living in a state of world economic war, and this is not just a military metaphor — the companies are training the armies, and the unemployed are the casualties.
>
> **— Bernard Esambert,**
> **President of the French Pasteur Institute,**
> **at a Paris Conference on Economic Espionage**

The Attorney General of the United States defined economic espionage as "the unlawful or clandestine targeting or acquisition of sensitive financial, trade, or economic policy information; proprietary economic information; or critical technologies." Note that this definition excludes the collection of open and legally available information that makes up the majority of economic collection. This means that aggressive intelligence collection that is entirely open and legal may harm U.S. companies but is not considered espionage, economic or otherwise. The FBI has extended this definition to include the unlawful or clandestine targeting or influencing of sensitive economic policy decisions.

Intelligence consists of two broad categories — open source and espionage. Open-source intelligence collection is the name given to legal intelligence activities. Espionage is divided into the categories of economic and military/political/governmental; the distinction is the targets involved. A common term, *industrial espionage* was used (and is still used to some degree) to indicate espionage between two competitors. As global competitors began to conduct these activities with possible assistance from their governments, the competitor-versus-competitor nature of industrial espionage became less of a discriminator. As the activities expanded to include sabotage and interference with commerce and proposal competitions, the term *economic espionage* was coined for the broader scope.

While the examples and cases discussed in this chapter focus mainly on the United States, the issues are universal. The recommendations and types of information gathered can and should be translated for any country.

Brief History

The prosperity and success of this country are due in no small measure to economic espionage committed by Francis Cabot Lowell during the Industrial Revolution. Britain replaced costly, skilled hand labor with water-driven looms that were simple and reliable. The looms were so simple that they could be operated by a few unskilled women and children. The British government passed strict patent laws and prohibited the export of technology related to the making of cotton. A law was passed making it illegal to hire skilled textile workers for work abroad. Those workers who went abroad had their property confiscated. It was against the law to make and export drawings of the mills.

So Lowell memorized and stole the plans to a Cartwright loom, a water-driven weaving machine. It is believed that Lowell perfected the art of *spying by driving around*. Working from Edinburgh, he and his wife traveled daily throughout the countryside, including Lancashire and Derbyshire, the hearts of the Industrial

Revolution. Returning home, he built a scale model of the loom. His company built its first loom in Waltham. Soon, his factories were capable of producing up to 30 miles of cloth a day.[1] This marked America's entry into the Industrial Revolution.

By the early 20th century, we had become "civilized" to the point that Henry L. Stimson, our Secretary of State, said for the record that "Gentlemen do not read other gentlemen's mail" while refusing to endorse a code-breaking operation. For a short time the U.S. Government was the only government that believed this fantasy. At the beginning of World War II, the United States found itself almost completely blind to activities inside Germany and totally dependent on other countries' intelligence services for information. In 1941 the United States recognized that espionage was necessary to reduce its losses and efficiently engage Germany. To meet this need, first the COI and then the OSS were created under the leadership of General "Wild Bill" Donovan.

It would take tremendous forces to broaden this awakening to include economic espionage.

Watershed: End of Cold War, Beginning of Information Age

In the late 1990s, two events occurred that radically changed information security for many companies. The end of the Cold War — marked by the collapse of the former Soviet Union — created a pool of highly trained intelligence officers without targets. In Russia, some continued to work for the government, some began to work in the newly created private sector, and some provided their services for the criminal element. Some did all three. The world's intelligence agencies began to focus their attention on economic targets and information war, just in time for watershed event number-two — the beginning of the information age.

John Lienhard, M.D. Anderson Professor of Mechanical Engineering and History at the University of Houston, is the voice and driving force behind the "Engines of Our Ingenuity," a syndicated program for public radio. He has said that the change of our world into an information society is not like the Industrial Revolution. No; this change is more like the change from a hunter-gatherer society to an agrarian society. A change of this magnitude happened only once or twice in all of history. Those who were powerful in the previous society may have no power in the new society. In the hunter-gatherer society, the strongest man and best hunter rules. But where is he in an agrarian society? There, the best hunter holds little or no power. During the transition to an information society, those with power in the old ways will not give it up easily. Now couple the turmoil caused by this shift with the timing of the "end" of the Cold War.

The currency of the new age is information. The power struggle in the new age is the struggle to gather, use, and control information. It is at the beginning of this struggle that the Cold War ended, making available a host of highly trained information gatherers to countries and companies trying cope with the new economy. Official U.S. acknowledgment of the threat of economic espionage came in 1996 with the passage of the Economic Espionage Act.

For the information security professional, the world has fundamentally changed. Until 1990, a common practice had been to make the cost of an attack prohibitively expensive. How do you make an attack prohibitively expensive when your adversaries have the resources of governments behind them?

Most information security professionals have not been trained and are not equipped to handle professional intelligence agents with deep pockets. Today, most business managers are incapable of fathoming that such a threat exists.

Role of Information Technology in Economic Espionage

In the 1930s, the German secret police divided the world of espionage into five roles.[2] Exhibit 14.1 illustrates some of the ways that information technology today performs these five divisions of espionage functionality.

In addition to these roles, information technology may be exploited as a target, used as a tool, used for storage (for good or bad), used as protection for critical assets as a weapon, used as a transport mechanism, or used as an agent to carry out tasks when activated.

- *Target.* Information and information technology can be the target of interest. The goal of the exploitation may be to discover new information assets (breach of confidentiality), deprive one of exclusive owner-

EXHIBIT 14.1 Five Divisions of Espionage Functionality

Role	WWII Description	IT Equivalent
Collectors	Located and gathered desired information	People or IT (hardware or software) agents, designer viruses that transmit data to the Internet
Transmitters	Forwarded the data to Germany, by coded mail or shortwave radio	E-mail, browsers with convenient 128-bit encryption, FTP, applications with built-in collection and transmission capabilities (e.g., comet cursors, Real Player, Media Player, or other spyware), covert channel applications
Couriers	Worked on steamship lines and transatlantic clippers, and carried special messages to and from Germany	Visiting country delegations, partners/suppliers, temporary workers, and employees that rotate in and out of companies with CD-R/CD-RW, Zip disks, tapes, drawings, digital camera images, etc.
Drops	Innocent-seeming addresses of businesses or private individuals, usually in South American or neutral European ports; reports were sent to these addresses for forwarding to Germany	E-mail relays, e-mail anonymizers, Web anonymizers, specially designed software that spreads information to multiple sites (the reverse of distributed DoS) to avoid detection
Specialists	Expert saboteurs	Viruses, worms, DDoS, Trojan horses, chain e-mail, hoaxes, using e-mail to spread dissension, public posting of sensitive information about salaries, logic bombs, insiders sabotaging products, benchmarks, etc.

ship, acquire a form of the asset that would permit or facilitate reverse-engineering, corrupt the integrity of the asset — either to diminish the reputation of the asset or to make the asset become an agent — or to deny the availability of the asset to those who rely on it (denial of service).

- *Tool.* Information technology can be the tool to monitor and detect traces of espionage or to recover information assets. These tools include intrusion detection systems, log analysis programs, content monitoring programs, etc. For the bad guys, these tools would include probes, enumeration programs, viruses that search for PGP keys, etc.
- *Storage.* Information technology can store stolen or illegal information. IT can store sleeper agents for later activation.
- *Protection.* Information technology may have the responsibility to protect the information assets. The protection may be in the form of applications such as firewalls, intrusion detection systems, encryption tools, etc., or elements of the operating system such as file permissions, network configurations, etc.
- *Transport.* Information technology can be the means by which stolen or critical information is moved, whether burned to CDs, e-mailed, FTP'd, hidden in a legitimate http stream, or encoded in images or music files.
- *Agent.* Information technology can be used as an agent of the adversary, planted to extract significant sensitive information, to launch an attack when given the appropriate signal, or to receive or initiate a covert channel through a firewall.

Implications for Information Security

Implication 1

A major tenet of our profession has been that, because we cannot always afford to prevent information system-related losses, we should make it prohibitively expensive to compromise those systems. How does one do that when the adversary has the resources of a government behind him? Frankly, this tenet only worked on adversaries who were limited by time, money, or patience. Hackers with unlimited time on their hands — and a bevy of unpaid researchers who consider a difficult system to be a trophy waiting to be collected — turn this tenet into Swiss cheese.

This reality has placed emphasis on the onion model of information security. In the onion model you assume that all other layers will fail. You build prevention measures but you also include detection measures that will tell you that those measures have failed. You plan for the recovery of critical information, assuming that your prevention and detection measures will miss some events.

Implication 2

Information security professionals must now be able to determine if their industry or their company is a target for economic espionage. If their company/industry is a target, then the information security professionals should adjust their perceptions of their potential adversaries and their limits. One of the best-known quotes from the *Art of War* by Sun Tsu says, "Know your enemy." Become familiar with the list of countries actively engaging in economic espionage against your country or within your industry. Determine if any of your vendors, contractors, partners, suppliers, or customers come from these countries. In today's global economy, it may not be easy to determine the country of origin. Many companies move their global headquarters to the United States and keep only their main R&D offices in the country of origin. Research the company and its founders. Learn where and how they gained their expertise. Research any publicized accounts regarding economic espionage/intellectual property theft attributed to the company, the country, or other companies from the country. Pay particular attention to the methods used and the nature of the known targets. Contact the FBI or its equivalent and see if they can provide additional information. Do not forget to check your own organization's history with each company. With this information you can work with your business leaders to determine what may be a target within your company and what measures (if any) may be prudent.

> He who protects everything, protects nothing.
>
> **— Napoleon**

Applying the wisdom of Napoleon implies that, within the semipermeable external boundary, we should determine which information assets truly need protection, to what degree, and from what threats. Sun Tsu speaks to this need as well. It is not enough to only know your enemy.

> Therefore I say, "Know the enemy and know yourself; in a hundred battles you will never be in peril."
>
> When you are ignorant of the enemy but know yourself, your chances of winning or losing are equal.
>
> If ignorant both of your enemy and yourself, you are certain in every battle to be in peril.
>
> **— Sun Tzu,**
> ***The Art of War* (III.31–33)**

A company can "know itself" using a variation from the business continuity concept of a business impact assessment (BIA). The information security professional can use the information valuation data collected during the BIA and extend it to produce information protection guides for sensitive and critical information assets. The information protection guides tell users which information should be protected, from what threats, and what to do if an asset is found unprotected. They should tell the technical staff about threats to each information asset and about any required and recommended safeguards.

A side benefit gained from gathering the information valuation data is that, in order to gather the value information, the business leaders must internalize questions of how the data is valuable and the degrees of loss that would occur in various scenarios. This is the most effective security awareness that money can buy.

After the information protection guides have been prepared, you should meet with senior management again to discuss the overall posture the company wants to take regarding information security and counter-economic espionage. Note that it is significant that you wait until after the information valuation exercise is complete before addressing the security posture. If management has not accepted the need for security, the question about desired posture will yield damaging results.

Here are some potential postures that you can describe to management:

- *Prevent all.* In this posture, only a few protocols are permitted to cross your external boundary.
- *City wall.* A layered approach, prevention, detection, mitigation, and recovery strategies are all, in effect, similar to the walled city in the Middle Ages. Traffic is examined, but more is permitted in and out. Because more is permitted, detection, mitigation, and recovery strategies are needed internally because the risk of something bad getting through is greater.
- *Aggressive.* A layered approach, but embracing new technology, is given a higher priority than protecting the company. New technology is selected, and then security is asked how they will deal with it.
- *Edge racer.* Only general protections are provided. The company banks on running faster than the competition. "We'll be on the next technology before they catch up with our current release." This is a common position before any awareness has been effective.

Implication 3

Another aspect of knowing your enemy is required. As security professionals we are not taught about spycraft. It is not necessary that we become trained as spies. However, the FBI, in its annual report to congress on economic espionage, gives a summary about techniques observed in cases involving economic espionage.

Much can be learned about modern techniques in three books written about the Mossad — *Gideon's Spies* by Gordon Thomas, and *By Way of Deception,* and *The Other Side of Deception*, both by Victor Ostrovsky and Claire Hoy. These describe the Mossad as an early adopter of technology as a tool in espionage, including their use of Trojan code in software sold commercially. The books describe software known as Promis that was sold to intelligence agencies to assist in tracking terrorists; and the authors allege that the software had a Trojan that permitted the Mossad to gather information about the terrorists tracked by its customers. *By Way of Deception* describes the training process as seen by Ostrovsky.

Implication 4

Think Globally, Act Locally

The Chinese government recently announced that the United States had placed numerous bugging devices on a plane for President Jiang Zemin. During the customization by a U.S. company of the interior of the plane for its use as the Chinese equivalent of Air Force One, bugs were allegedly placed in the upholstery of the president's chair, in his bedroom, and even in the toilet.

When the United States built a new embassy in Moscow, the then-extant Soviet Union insisted it be built using Russian workers. The United States called a halt to its construction in 1985 when it discovered it was too heavily bugged for diplomatic purposes. The building remained unoccupied for a decade following the discovery.

The *1998 Annual Report to Congress on Foreign Economic Collection and Industrial Espionage* concluded with the following statement:

> ...foreign software manufacturers solicited products to cleared U.S. companies that had been embedded with spawned processes and multithreaded tasks.

This means that foreign software companies sold products with Trojans and backdoors to targeted U.S. companies.

In response to fears about the Echelon project, in 2001 the European Union announced recommendations that member nations use open-source software to ensure that Echelon software agents are not present.

Security teams would benefit by using open-source software tools if they could be staffed sufficiently to maintain and continually improve the products. Failing that, security in companies in targeted industries should consider the origins of the security products they use. If your company knows it is a target for economic espionage, it would be wise to avoid using security products from countries actively engaged in economic espionage against your country. If unable to follow this strategy, the security team should include tools in the architecture (from other countries) that could detect extraneous traffic or anomalous behavior of the other security tools.

In this strategy you should follow the effort all the way through implementation. In one company, the corporate standard for firewall was a product of one of the most active countries engaging in economic espionage. Management was unwilling to depart from the standard. Security proposed the use of an intrusion detection system (IDS) to guard against the possibility of the firewall being used to permit undetected, unfiltered, and unreported access. The IDS was approved; but when procurement received the order, they discovered that the firewall vendor sold a special, optimized version of the same product and — without informing the security team — ordered the IDS from the vendor that the team was trying to guard against.

Implication 5

The system of rating computers for levels of security protection is incapable of providing useful information regarding products that might have malicious code that is included intentionally. In fact, companies that have intentions of producing code with these Trojans are able to use the system of ratings to gain credibility without merit.

It appears that the first real discovery by one of the ratings systems caused the demise of the ratings system and a cover-up of the findings. I refer to the MISSI ratings system's discovery of a potential backdoor in Checkpoint Firewall-1 in 1997. After this discovery, the unclassified X31 report[3] for this product and all previous reports were pulled from availability. The Internet site that provided them was shut down, and requestors were told that the report had been classified. The federal government had begun pulling Checkpoint Firewall-1 from military installations and replacing it with other companies' products. While publicly denying that these actions were happening, Checkpoint began correspondence with the NSA, owners of the MISSI process, to answer the findings of that study. The NSA provided a list of findings and preferred corrective actions to resolve the issue. In Checkpoint's response[4] to the NSA, they denied that the code in question, which involved SNMP and which referenced files containing IP addresses in Israel, was a backdoor. According to the NSA, two files with IP addresses in Israel "could provide access to the firewall via SNMPv2 mechanisms." Checkpoint's reply indicated that the code was dead code from Carnegie Mellon University and that the files were QA testing data that was left in the final released configuration files.

The X31 report, which I obtained through an FOIA request, contains no mention of the incident and no indication that any censorship had occurred. This fact is particularly disturbing because a report of this nature should publish all issues and their resolutions to ensure that there is no complicity between testers and the test subjects.

However, the letter also reveals two other vulnerabilities that I regard as backdoors, although the report classes them as software errors to be corrected. The Checkpoint response to some of these "errors" is to defend aspects of them as desirable. One specific reference claims that most of Checkpoint's customers prefer maximum connectivity to maximum security, a curious claim that I have not seen in their marketing material. This referred to the lack of an ability to change the implicit rules in light of the vulnerability of stateful inspection's handling of DNS using UDP, which existed in Version 3 and earlier.

Checkpoint agreed to most of the changes requested by the NSA; however, the exception is notable in that it would have required Checkpoint to use digital signatures to sign the software and data electronically to prevent someone from altering the product in a way that would go undetected. These changes would have provided licensees of the software with the ability to know that, at least initially, the software they were running was indeed the software and data that had been tested during the security review.

It is interesting to note that Checkpoint had released an internal memo nine months prior to the letter responding to the NSA claims in which they claimed nothing had ever happened.[5]

Both the ITSEC and Common Criteria security rating systems are fatally flawed when it comes to protection against software with intentional malicious code. Security companies are able to submit the software for rating and claim the rating even when the entire system has not been submitted. For example, a company can submit the assurance processes and documentation for a targeted rating. When it achieves the rating on just that

EXHIBIT 14.2 Military Critical Technologies (MCTs)

Information systems
Sensors and lasers
Electronics
Aeronautics systems technology
Armaments and energetic materials
Marine systems
Guidance, navigation, and vehicle signature control
Space systems
Materials
Manufacturing and fabrication
Information warfare
Nuclear systems technology
Power systems
Chemical/biological systems
Weapons effects and countermeasures
Ground systems
Directed and kinetic energy systems

portion, it can advertise the rating although the full software functionality has not been tested. For marketing types, they gain the benefit of claiming the rating without the expense of full testing. Even if the rating has an asterisk, the damage is done because many that authorize the purchase of these products only look for the rating. When security reports back to management that the rating only included a portion of the software functionality, it is portrayed as sour grapes by those who negotiated the "great deal" they were going to get. The fact is that there is no commercial push to require critical software such as operating systems and security software to include exhaustive code reviews, covert channel analysis, and to only award a rating when it is fully earned.

To make matters worse, if it appears that a company is going to get a poor rating from a test facility, the vendor can stop the process and start over at a different facility, perhaps in another country, with no penalty and no carry-over.

What Are the Targets?

The U.S. government publishes a list of military critical technologies (MCTs). A summary of the list is published annually by the FBI (see Exhibit 14.2).

There is no equivalent list for nonmilitary critical technologies. However, the government has added "targeting the national information infrastructure" to the National Security Threat List (NSTL). Targeting the national information infrastructure speaks primarily to the infrastructure as an object of potential disruption, whereas the MCT list contains technologies that foreign governments may want to acquire illegally. The NSTL consists of two tables. One is a list of issues (see Exhibit 14.3); the other is a classified list of countries engaged in collection activities against the United States. This is not the same list captured in Exhibit 14.4. Exhibit 14.4 contains the names of countries engaged in economic espionage and, as such, contains the names of countries that are otherwise friendly trading partners. You will note that the entire subject of economic espionage is listed as one of the threat list issues.

According to the FBI, the collection of information by foreign agencies continues to focus on U.S. trade secrets and science and technology products, particularly dual-use technologies and technologies that provide high profitability.

Examining the cases that have been made public, you can find intellectual property theft, theft of proposal information (bid amounts, key concepts), and requiring companies to participate in joint ventures to gain access to new country markets — then either stealing the IP or awarding the contract to an internal company with an identical proposal. Recently, a case involving HP found a planted employee sabotaging key bench-

EXHIBIT 14.3 National Security Threat List Issues

Terrorism
Espionage
Proliferation
Economic espionage
Targeting the national information infrastructure
Targeting the U.S. Government
Perception management
Foreign intelligence activities

EXHIBIT 14.4 Most Active Collectors of Economic Intelligence

China
Japan
Israel
France
Korea
Taiwan
India

marking tests to HP's detriment. The message from the HP case is that economic espionage also includes efforts beyond the collection of information, such as sabotage of the production line to cause the company to miss key delivery dates, deliver faulty parts, fail key tests, etc.

You should consider yourself a target if your company works in any of the technology areas on the MCT list, is a part of the national information infrastructure, or works in a highly competitive international business.

Who Are the Players?

Countries

This section is written from the published perspective of the U.S. Government. Readers from other countries should attempt to locate a similar list from their government's perspective. It is likely that two lists will exist: a "real" list and a "diplomatically correct" edition.

For the first time since its original publication in 1998, the *Annual Report to Congress on Foreign Economic Collection and Industrial Espionage 2000* lists the most active collectors of economic intelligence. The delay in providing this list publicly is due to the nature of economic espionage. To have economic espionage you must have trade. Our biggest trading partners are our best friends in the world. Therefore, a list of those engaged in economic espionage will include countries that are otherwise friends and allies. Thus the poignancy of Bernard Esambert's quote used to open this chapter.

Companies

Stories of companies affected by economic espionage are hard to come by. Public companies fear the effect on stock prices. Invoking the economic espionage law has proven very expensive — a high risk for a favorable outcome — and even the favorable outcomes have been inadequate considering the time, money, and commitment of company resources beyond their primary business. The most visible companies are those that have been prosecuted under the Economic Espionage Act, but there have only been 20 of those, including:

- Four Pillars Company, Taiwan, stole intellectual property and trade secrets from Avery Dennison.
- Laser Devices, Inc., attempted to illegally ship laser gun sights to Taiwan without Department of Commerce authorization.
- Gilbert & Jones, Inc., New Britain, Connecticut, exported potassium cyanide to Taiwan without the required licenses.
- Yuen Foong Paper Manufacturing Company, Taiwan, attempted to steal the formula for Taxol, a cancer drug patented and licensed by the Bristol-Myers Squibb (BMS) Company.
- Steven Louis Davis attempted to disclose trade secrets of the Gillette Company to competitors Warner-Lambert Co., Bic, and American Safety Razor Co. The disclosures were made by fax and e-mail. Davis worked for Wright Industries, a subcontractor of the Gillette Company.
- Duplo Manufacturing Corporation, Japan, used a disgruntled former employee of Standard Duplicating Machines Corporation to gain unauthorized access into a voicemail system. The data was used to compete against Standard. Standard learned of the issue through an unsolicited phone call from a customer.
- Harold Worden attempted to sell Kodak trade secrets and proprietary information to Kodak rivals, including corporations in the Peoples Republic of China. He had formerly worked for Kodak. He established his own consulting firm upon retirement and subsequently hired many former Kodak employees. He was convicted on one felony count of violating the Interstate Transportation of Stolen Property law.
- In 1977, Mitsubishi Electric bought one of Fusion Systems Corporation's microwave lamps, took it apart, then filed 257 patent actions on its components. Fusion Systems had submitted the lamp for a patent in Japan two years earlier. After 25 years of wrangling with Mitsubishi, the Japanese patent system, Congress, and the press, Fusion's board fired the company's president (who had spearheaded the fight) and settled the patent dispute with Mitsubishi a year later.
- The French are known to have targeted IBM, Corning Glass, Boeing, Bell Helicopter, Northrup, and Texas Instruments (TI). In 1991, a guard in Houston noticed two well-dressed men taking garbage bags from the home of an executive of a large defense contractor. The guard ran the license number of the van and found it belonged to the French Consul General in Houston, Bernard Guillet. Two years earlier, the FBI had helped TI remove a French sleeper agent. According to *Cyber Wars*[6] by Jean Guisnel, the French intelligence agency (the DGSE) had begun to plant young French engineers in various French subsidiaries of well-known American firms. Over the years they became integral members of the companies they had entered, some achieving positions of power in the corporate hierarchy. Guillet claims that the primary beneficiary of these efforts was the French giant electronics firm, Bull.

What Has Been Done? Real-World Examples

Partnering with a Company and Then Hacking the Systems Internally

In one case, very senior management took a bold step. In the spirit of the global community, they committed the company to use international partners for major aspects of a new product. Unfortunately, in selecting the partners, they chose companies from three countries listed as actively conducting economic espionage against their country. In the course of developing new products, the employees of one company were caught hacking sensitive systems. Security measures were increased but the employees hacked through them as well. The company of the offending partners was confronted. Its senior management claimed that the employees had acted alone and that their actions were not sanctioned. Procurement, now satisfied that their fragile quilt of partners was okay, awarded the accused partner company a lucrative new product partnership. Additionally, they erased all database entries regarding the issues and chastised internal employees who continued to voice suspicions. No formal investigation was launched. Security had no record of the incident. There was no information security function at the time of the incident.

When the information security function was established, it stumbled upon rumors that these events had occurred. In investigating, they found an internal employee who had witnessed the stolen information in use at the suspect partner's home site. They also determined that the offending partner had a history of economic espionage, perhaps the most widely known in the world. Despite the corroboration of the partner's complicity,

line management and procurement did nothing. Procurement knew that the repercussions within their own senior management and line management would be severe because they had pressured the damaged business unit to accept the suspected partner's earlier explanation. Additionally, it would have underscored the poor choice of partners that had occurred under their care and the fatal flaw in the partnering concept of very senior management. It was impossible to extricate the company from this relationship without causing the company to collapse. IT line management would not embrace this issue because they had dealt with it before and had been stung, although they were right all along.

Using Language to Hide in Plain Sight

Israeli Air Force officers assigned to the Recon/Optical Company passed on technical information beyond the state-of-the-art optics to a competing Israeli company, El Op Electro-Optics Industries Ltd. Information was written in Hebrew and faxed. The officers tried to carry 14 boxes out of the plant when the contract was terminated. The officers were punished upon return to Israel — for getting caught.[7]

In today's multinational partnerships, language can be a significant issue for information security and for technical support. Imagine the difficulty in monitoring and supporting computers for five partners, each in a different language.

The *Annual Report to Congress 2000*[8] reveals that the techniques used to steal trade secrets and intellectual property are limitless. The insider threat, briefcase and laptop computer thefts, and searching hotel rooms have all been used in recent cases. The information collectors are using a wide range of redundant and complementary approaches to gather their target data. At border crossings, foreign officials have conducted excessive attempts at elicitation. Many U.S. citizens unwittingly serve as third-party brokers to arrange visits or circumvent official visitation procedures. Some foreign collectors have invited U.S. experts to present papers overseas to gain access to their expertise in export-controlled technologies. There have been recent solicitations to security professionals asking for research proposals for security ideas as a competition for awarding grants to conduct studies on security topics. The solicitation came from one of the most active countries engaging in economic espionage. Traditional clandestine espionage methods (such as agent recruitment, U.S. volunteers, and co-optees) are still employed. Other techniques include:

- Breaking away from tour groups
- Attempting access after normal working hours
- Swapping out personnel at the last minute
- Customs holding laptops for an extended period of time
- Requests for technical information
- Elicitation attempts at social gatherings, conferences, trade shows, and symposia
- Dumpster diving (searching a company's trash for corporate proprietary data)
- Using unencrypted Internet messages

To these I would add holding out the prospect of lucrative sales or contracts, but requiring the surrender or sharing of intellectual property as a condition of partnering or participation.

What Can We, as Information Security Professionals, Do?

We must add new skills and improve our proficiency in others to meet the challenge of government funded/supported espionage. Our investigative and forensic skills need improvement over the level required for nonespionage cases. We need to be aware of the techniques that have been and may be used against us. We need to add the ability to elicit information without raising suspicion. We need to recognize when elicitation is attempted and be able to teach our sales, marketing, contracting, and executive personnel to recognize such attempts. We need sources that tell us where elicitation is likely to occur. For example, at this time, the Paris Air Show is considered the number-one economic espionage event in the world.

We need to be able to raise the awareness of our companies regarding the perceived threat and real examples from industry that support those perceptions. Ensure that you brief the procurement department. Establish preferences for products from countries not active in economic espionage. When you must use a product from a country active in economic espionage, attempt to negotiate an indemnification against loss. Have procurement

add requirements that partners/suppliers provide proof of background investigations, particularly if individuals will be on site.

Management and procurement should be advised that those partners with intent to commit economic espionage are likely to complain to management that the controls are too restrictive, that they cannot do their jobs, or that their contract requires extraordinary access. You should counter these objectives before they occur by fully informing management and procurement about awareness, concerns, and measures to be taken. The measures should be applied to all suppliers/partners. Ensure that these complaints and issues will be handed over to you for an official response. Treat each one individually and ask for specifics rather than generalities.

If procurement has negotiated a contract that commits the company to extraordinary access, your challenge is greater. Procurement may insist that you honor their contract. At this time you will discover where security stands in the company's pecking order. A stance you can take is, "Your negotiated contract does not and cannot relieve me of my obligation to protect the information assets of this corporation." It may mean that the company has to pay penalties or go back to the negotiating table. You should not have to sacrifice the security of the company's information assets to save procurement some embarrassment.

We need to develop sources to follow developments in economic espionage in industries and businesses similar to ours. Because we are unlikely to have access to definitive sources about this kind of information, we need to develop methods to vet the information we find in open sources. The FBI provides advanced warning to security professionals through ANSIR (Awareness of National Security Issues and Responses) systems. Interested security professionals for U.S. corporations should provide their e-mail addresses, positions, company names and addresses, and telephone and fax numbers to ansir@leo.gov. A representative of the nearest field division office will contact you. The FBI has also created InfraGard (http:// www.infragard.net/fieldoffice.htm) chapters for law enforcement and corporate security professionals to share experiences and advice.[9]

InfraGard is dedicated to increasing the security of the critical infrastructures of the United States. All InfraGard participants are committed to the proposition that a robust exchange of information about threats to and actual attacks on these infrastructures is an essential element in successful infrastructure protection efforts. The goal of InfraGard is to enable information flow so that the owners and operators of infrastructures can better protect themselves and so that the U.S. Government can better discharge its law enforcement and national security responsibilities.

Barriers Encountered in Attempts to Address Economic Espionage

A country is made up of many opposing and cooperating forces. Related to economic espionage, for information security, there are two significant forces. One force champions the businesses of that country. Another force champions the relationships of that country to other countries. Your efforts to protect your company may be hindered by the effect of the opposition of those two forces. This was evident in the first few reports to Congress by the FBI on economic espionage. The FBI was prohibited from listing even the countries that were most active in conducting economic espionage. There is no place in the U.S. Government that you can call to determine if a partner you are considering has a history of economic espionage, or if a software developer has been caught with backdoors, placing Trojans, etc.

You may find that, in many cases, the FBI interprets the phrase *information sharing* to mean that you share information with them. In one instance, a corporate investigator gave an internal e-mail that was written in Chinese to the FBI, asking that they translate it. This was done to keep the number of individuals involved in the case to a minimum. Unless you know the translator and his background well, you run the risk of asking someone that might have ties to the Chinese to perform the translation. Once the translation was performed, the FBI classified the document as secret and would not give the investigator the translated version until the investigator reasoned with them that he would have to translate the document with an outside source unless the FBI relented.

Part of the problem facing the FBI is that there is no equivalent to a DoD or DoE security clearance for corporate information security personnel. There are significant issues that complicate any attempt to create such a clearance. A typical security clearance background check looks at criminal records. Background investigations may go a step further and check references, interview old neighbors, schoolmates, colleagues, etc. The most rigorous clearance checks include viewing bank records, credit records, and other signs of fiscal responsibility. They may include a psychological evaluation. They are not permitted to include issues of national origin or religion unless the United States is at war with a particular country. In those cases, the DoD has granted the clearance

but placed the individuals in positions that would not create a conflict of interest. In practice, this becomes impossible. Do you share information about all countries and religious groups engaging in economic espionage, except for those to which the security officer may have ties? Companies today cannot ask those questions of its employees. Unfortunately, unless a system of clearances is devised, the FBI will always be reluctant to share information, and rightfully so.

Another aspect of the problem facing the FBI today is the multinational nature of corporations today. What exactly is a U.S. corporation? Many companies today were conceived in foreign countries but established their corporate headquarters in the United States, ostensibly to improve their competitiveness in the huge U.S. marketplace. What of U.S. corporations that are wholly owned by foreign corporations? Should they be entitled to assistance, to limited assistance, or to no assistance? If limited assistance, how are the limits determined?

Within your corporation there are also opposing and cooperating forces. One of the most obvious is the conflict between marketing/sales and information security. In many companies, sales and marketing personnel are the most highly paid and influential people in the company. They are, in most cases, paid largely by commission. This means that if they do not make the sale, they do not get paid. They are sometimes tempted to give the potential customer anything they want, in-depth tours of the plant, details on the manufacturing process, etc., in order to make the sale. Unless you have a well-established and accepted information protection guide that clearly states what can and cannot be shared with these potential customers, you will have little support when you try to protect the company.

The marketing department may have such influence that they cause your procurement personnel to abandon reason and logic in the selection of critical systems and services. A Canadian company went through a lengthy procurement process for a massive wide area network contract. An RFP was released. Companies responded. A selection committee met and identified those companies that did not meet the RFP requirements. Only those companies that met the RFP requirements were carried over into the final phase of the selection process. At this point, marketing intervened and required that procurement re-add two companies to the final selection process — companies that had not met the requirements of the RFP. These two companies purchased high product volumes from this plant. Miracle of miracles, one of the two unqualified companies won the contract.

It is one thing for the marketing department to request that existing customers be given some preference from the list of qualified finalists. It is quite another to require that unqualified respondents be given any consideration.

A product was developed in a country that conducts economic espionage operations against U.S. companies in your industry sector. This product was widely used throughout your company, leaving you potentially vulnerable to exploitation or exposed to a major liability. When the issue was raised, management asked if this particular product had a Trojan or evidence of malicious code. The security officer responded, "No, but due to the nature of this product, if it did contain a Trojan or other malicious code, it could be devastating to our company. Because there are many companies that make this kind of product in countries that do not conduct economic espionage in our industry sector, we should choose one of those to replace this one and thus avoid the risk."

Management's response was surprising. "Thank you very much, but we are going to stay with this product and spread it throughout the corporation — but do let us know if you find evidence of current backdoors and the like." One day the security team learned that, just as feared, there had indeed been a backdoor; in fact, several. The news was reported to management. Their response was unbelievable. "Well, have they fixed it?" The vendor claimed to have fixed it, but that was not the point. The point was that they had placed the code in the software to begin with, and there was no way to tell if they had replaced the backdoor with another. Management responded, "If they have fixed the problem, we are going to stay with the product, and that is the end of it. Do not bring this subject up again." In security you must raise every security concern that occurs with a product, even after management has made up its mind. To fail to do so would set the company up for charges of negligence should a loss occur that relates to that product. "Doesn't matter; do not raise this subject again."

So why would management make a decision like this? One possible answer has to do with pressure from marketing and potential sales to that country. Another has to do with embarrassment. Some vice president or director somewhere made a decision to use the product to begin with. They may even have had to fall on a sword or two to get the product they wanted. Perhaps it is because a more powerful director had already chosen this product for his site. This director may have forced the product's selection as the corporate standard so that staff would not be impacted. One rumor has it that the product was selected as a corporate standard

because the individual choosing the standard was being paid a kickback by a relative working for a third-party vendor of the product. If your IT department raises the issue, it runs the risk of embarrassing one or more of these senior managers and incurring their wrath. Your director may feel intimidated enough that he will not even raise the issue.

Even closer to home is the fact that the issue was raised to your management in time to prevent the spread of the questionable product throughout the corporation. Now if the flag is raised, someone may question why it was not raised earlier. That blame would fall squarely on your director's shoulders.

Does it matter that both the vice president and the director have fiduciary responsibility for losses related to these decisions should they occur? Does it matter that their decisions would not pass the prudent man test and thus place them one step closer to being found negligent? No, it does not. The director is accepting the risk — not the risk to the corporation, but the risk that damage might occur during his watch. The vice president probably does not know about the issue or the risks involved but could still be implicated via the concept of respondent superior. The director may think he is protecting the vice president by keeping him out of the loop — the concept of plausible deniability — but the courts have already tackled that one. Senior management is responsible for the actions of those below them, regardless of whether they know about the actions.

Neither of these cases exists if the information security officer reports to the CEO. There is only a small opportunity for it to exist if the information security officer reports to the CIO. As the position sinks in the management structure, the opportunity for this type of situation increases.

The first time you raise the specter of economic espionage, you may encounter resistance from employees and management. "Our company isn't like that. We don't do anything important. No one I know has ever heard of anything like that happening here. People in this community trust one another."

Some of those who have been given evidence that such a threat does exist have preferred to ignore the threat, for to acknowledge it would require them to divert resources (people, equipment, or money) from their own initiatives and goals. They would prefer to "bet the company" that it would not occur while they are there. After they are gone it no longer matters to them.

When you raise these issues as the information security officer, you are threatening the careers of many people — from the people who went along with it because they felt powerless to do anything, to the senior management who proposed it, to the people in between who protected the concept and decisions of upper management in good faith to the company. Without a communication path to the CEO and other officers representing the stockholders, you do not have a chance of fulfilling your fiduciary liability to them.

> The spy of the future is less likely to resemble James Bond, whose chief assets were his fists, than the Line X engineer who lives quietly down the street and never does anything more violent than turn a page of a manual or flick on his computer.
>
> **— Alvin Toffler,**
> ***Power Shift: Knowledge, Wealth and Violence at the Edge of the 21st Century***

References

1. *War by Other Means,* John J. Fialka, W.W. Norton Company, 1997.
2. *Sabotage! The Secret War Against America,* Michael Sayers and Albert E. Kahn, Harper & Brothers, 1942, p. 25.
3. NSA X3 Technical Report X3-TR001–97 Checkpoint Firewall-1 Version 3.0a, Analysis and Penetration Test Report.
4. Letter of reply from David Steinberg, Director, Federal Checkpoint Software, Inc. to Louis F. Giles, Deputy Chief Commercial Solutions & Enabling Technology; 9800 Savage Road Suite 6740, Ft. Meade, MD, dated September 10, 1998.
5. E-mail from Craig Johnson dated June 3, 1998, containing memo dated Jan 19, 1998, to all U.S. Sales of Checkpoint.
6. *Cyber Wars*, Jean Guisnel, Perseus Books, 1997.
7. *War by Other Means*, John J Fialka, W.W. Norton Company, 1997, pp. 181–184.

8. *Annual Report to Congress on Foreign Economic Collection and Industrial Espionage — 2000*, prepared by the National Counterintelligence Center.
9. Infragard National By-Laws, undated, available online at http://www.infragard.net/applic_ requirements/natl_bylaws.htm.

15
Penetration Testing

Stephen D. Fried, CISSP

This chapter provides a general introduction to the subject of penetration testing and provides the security professional with the background needed to understand this special area of security analysis. Penetration testing can be a valuable tool for understanding and improving the security of a computer or network. However, it can also be used to exploit system weaknesses and attack systems and steal valuable information. By understanding the need for penetration testing, and the issues and processes surrounding its use, a security professional will be better able to use penetration testing as a standard part of the analysis toolkit.

This chapter presents penetration testing in terms of its use, application, and process. It is not intended as an in-depth guide to specific techniques that can be used to test penetration-specific systems. Penetration testing is an art that takes a great deal of skill and practice to do effectively. If not done correctly and carefully, the penetration test can be deemed invalid (at best) and, in the worst case, actually damage the target systems. If the security professional is unfamiliar with penetration testing tools and techniques, it is best to hire or contract someone with a great deal of experience in this area to advise and educate the security staff of an organization.

What is Penetration Testing?

Penetration testing is defined as a formalized set of procedures designed to bypass the security controls of a system or organization for the purpose of testing that system's or organization's resistance to such an attack. Penetration testing is performed to uncover the security weaknesses of a system and to determine the ways in which the system can be compromised by a potential attacker. Penetration testing can take several forms (which will be discussed later) but, in general, a test consists of a series of "attacks" against a target. The success or failure of the attacks, and how the target reacts to each attack, will determine the outcome of the test.

The overall purpose of a penetration test is to determine the subject's ability to withstand an attack by a hostile intruder. As such, the tester will be using the tricks and techniques a real-life attacker might use. This simulated attack strategy allows the subject to discover and mitigate its security weak spots before a real attacker discovers them.

The reason penetration testing exists is that organizations need to determine the effectiveness of their security measures. The fact that they want tests performed indicates that they believe there might be (or want to discover) some deficiency in their security. However, while the testing itself might uncover problems in the organization's security, the tester should attempt to discover and explain the underlying cause of the lapses in security that allowed the test to succeed. Simply stating that the tester was able to walk out of a building with sensitive information is not sufficient. The tester should explain that the lapse was due to inadequate attention by the guard on duty or a lack of guard staff training that would enable them to recognize valuable or sensitive information.

There are three basic requirements for a penetration test. First, the test must have a defined goal and that goal should be clearly documented. The more specific the goal, the easier it will be to recognize the success or failure of the test. A goal such as "break into the XYZ corporate network," while certainly attainable, is not as precise as "break into XYZ's corporate network from the Internet and gain access to the research department's file server." Each test should have a single goal. If the tester wishes to test several aspects of security at a business

or site, several separate tests should be performed. This will enable the tester to more clearly distinguish between successful tests and unsuccessful attempts.

The test should have a limited time period in which it is to be performed. The methodology in most penetration testing is to simulate the types of attacks that will be experienced in the real world. It is reasonable to assume that an attacker will expend a finite amount of time and energy trying to penetrate a site. That time may range from one day to one year or beyond; but after that time is reached, the attacker will give up. In addition, the information being protected may have a finite useful "lifetime." The penetration test should acknowledge and accept this fact. Thus, part of the goal statement for the test should include a time limit that is considered reasonable based on the type of system targeted, the expected level of the threat, and the lifetime of the information.

Finally, the test should have the approval of the management of the organization that is the subject of the test. This is extremely important, as only the organization's management has the authority to permit this type of activity on its network and information systems.

Terminology

There are several terms associated with penetration testing. These terms are used throughout this chapter to describe penetration testing and the people and events involved in a penetration test.

The ***tester*** is the person or group who is performing the penetration test. The purpose of the tester is to plan and execute the penetration test and analyze the results for management. In many cases, the tester will be a member of the company or organization that is the subject of the test. However, a company may hire an outside firm to conduct the penetration test if it does not have the personnel or the expertise to do it itself.

An ***attacker*** is a real-life version of a tester. However, where the tester works with a company to improve its security, the attacker works against a company to steal information or resources.

An ***attack*** is the series of activities performed by the tester in an attempt to circumvent the security controls of a particular target. The attack may consist of physical, procedural, or electronic methods.

The ***subject*** of the test is the organization upon whom the penetration test is being performed. The subject can be an entire company or it can be a smaller organizational unit within that company.

A ***target*** of a penetration test is the system or organization that is being subjected to a particular attack at any given time. The target may or may not be aware that it is being tested. In either case, the target will have a set of defenses it presents to the outside world to protect itself against intrusion. It is those defenses that the penetration test is designed to test. A full penetration test usually consists of a number of attacks against a number of different targets.

Management is the term used to describe the leadership of an organization involved in the penetration test. There may be several levels of management involved in any testing effort, including the management of the specific areas of the company being tested, as well as the upper management of the company as a whole. The specific levels of management involved in the penetration testing effort will have a direct impact on the scope of the test. In all cases, however, it is assumed that the tester is working on behalf of (and sponsored by) at least one level of management within the company.

The ***penetration test*** (or, more simply, the ***test***) is the actual performance of a simulated attack on the target.

Why Test?

There are several reasons why an organization will want a penetration test performed on its systems or operations. The first (and most prevalent) is to determine the effectiveness of the security controls the organization has put into place. These controls may be technical in nature, affecting the computers, network, and information systems of the organization. They may be operational in nature, pertaining to the processes and procedures a company has in place to control and secure information. Finally, they may be physical in nature. The tester may be trying to determine the effectiveness of the physical security a site or company has in place. In all cases, the goal of the tester will be to determine if the existing controls are sufficient by trying to get around them.

The tester may also be attempting to determine the vulnerability an organization has to a particular threat. Each system, process, or organization has a particular set of threats to which it feels it is vulnerable. Ideally, the organization will have taken steps to reduce its exposure to those threats. The role of the tester is to determine the effectiveness of these countermeasures and to identify areas for improvement or areas where

additional countermeasures are required. The tester may also wish to determine whether the set of threats the organization has identified is valid and whether or not there are other threats against which the organization might wish to defend itself.

A penetration test can sometimes be used to bolster a company's position in the marketplace. A test, executed by a reputable company and indicating that the subject's environment withstood the tester's best efforts, can be used to give prospective customers the appearance that the subject's environment is secure. The word "appearance" is important here because a penetration test cannot examine all possible aspects of the subject's environment if it is even moderate in size. In addition, the security state of an enterprise is constantly changing as new technology replaces old, configurations change, and business needs evolve. The "environment" the tester examines may be very different from the one the customer will be a part of. If a penetration test is used as proof of the security of a particular environment for marketing purposes, the customer should insist on knowing the details, methodology, and results of the test.

A penetration test can be used to alert the corporation's upper management to the security threat that may exist in its systems or operations. While the general knowledge that security weaknesses exist in a system, or specific knowledge of particular threats and vulnerabilities may exist among the technical staff, this message may not always be transmitted to management. As a result, management may not fully understand or appreciate the magnitude of the security problem. A well-executed penetration test can systematically uncover vulnerabilities that management was unaware existed. The presentation of concrete evidence of security problems, along with an analysis of the damage those problems can cause to the company, can be an effective wake-up call to management and spur them into paying more attention to information security issues. A side effect of this wake-up call may be that once management understands the nature of the threat and the magnitude to which the company is vulnerable, it may be more willing to expend money and resources to address not only the security problems uncovered by the test but also ancillary security areas needing additional attention by the company. These ancillary issues may include a general security awareness program or the need for more funding for security technology. A penetration test that uncovers moderate or serious problems in a company's security can be effectively used to justify the time and expense required to implement effective security programs and countermeasures.

Types of Penetration Testing

The typical image of a penetration test is that of a team of high-tech computer experts sitting in a small room attacking a company's network for days on end or crawling through the ventilation shafts to get into the company's "secret room." While this may be a glamorous image to use in the movies, in reality the penetration test works in a variety of different (and very nonglamorous) ways.

The first type of testing involves the physical infrastructure of the subject. Very often, the most vulnerable parts of a company are not found in the technology of its information network or the access controls found in its databases. Security problems can be found in the way the subject handles its physical security. The penetration tester will seek to exploit these physical weaknesses. For example, does the building provide adequate access control? Does the building have security guards, and do the guards check people as they enter or leave a building? If intruders are able to walk unchecked into a company's building, they will be able to gain physical access to the information they seek. A good test is to try to walk into a building during the morning when everyone is arriving to work. Try to get in the middle of a crowd of people to see if the guard is adequately checking the badges of those entering the building.

Once inside, check if sensitive areas of the building are locked or otherwise protected by physical barriers. Are file cabinets locked when not in use? How difficult is it to get into the communications closet where all the telephone and network communication links terminate? Can a person walk into employee office areas unaccompanied and unquestioned? All the secure and sensitive areas of a building should be protected against unauthorized entry. If they are not, the tester will be able to gain unrestricted access to sensitive company information.

While the physical test includes examining protections against unauthorized entry, the penetration test might also examine the effectiveness of controls prohibiting unauthorized exit. Does the company check for theft of sensitive materials when employees exit the facility? Are laptop computers or other portable devices registered and checked when entering and exiting the building? Are security guards trained not only on what types of equipment and information to look for, but also on how equipment can be hidden or masked and why this procedure is important?

Another type of testing examines the operational aspects of an organization. Whereas physical testing investigates physical access to company computers, networks, or facilities, operational testing attempts to determine the effectiveness of the operational procedures of an organization by attempting to bypass those procedures. For example, if the company's help desk requires each user to give personal or secret information before help can be rendered, can the tester bypass those controls by telling a particularly believable "sob story" to the technician answering the call? If the policy of the company is to "scramble" or demagnetize disks before disposal, are these procedures followed? If not, what sensitive information will the tester find on disposed disks and computers? If a company has strict policies concerning the authority and process required to initiate ID or password changes to a system, can someone simply claiming to have the proper authority (without any actual proof of that authority) cause an ID to be created, removed, or changed? All these are attacks against the operational processes a company may have, and all of these techniques have been used successfully in the past to gain entry into computers or gain access to sensitive information.

The final type of penetration test is the electronic test. Electronic testing consists of attacks on the computer systems, networks, or communications facilities of an organization. This can be accomplished either manually or through the use of automated tools. The goal of electronic testing is to determine if the subject's internal systems are vulnerable to an attack through the data network or communications facilities used by the subject.

Depending on the scope and parameters of a particular test, a tester may use one, two, or all three types of tests. If the goal of the test is to gain access to a particular computer system, the tester may attempt a physical penetration to gain access to the computer's console or try an electronic test to attack the machine over the network. If the goal of the test is to see if unauthorized personnel can obtain valuable research data, the tester may use operational testing to see if the information is tracked or logged when accessed or copied and determine who reviews those access logs. The tester may then switch to electronic penetration to gain access to the computers where the information is stored.

What Allows Penetration Testing to Work?

There are several general reasons why penetration tests are successful. Many of them are in the operational area; however, security problems can arise due to deficiencies in any of the three testing areas.

A large number of security problems arise due to a lack of awareness on the part of a company's employees of the company's policies and procedures regarding information security and protection. If employees and contractors of a company do not know the proper procedures for handling proprietary or sensitive information, they are much more likely to allow that information to be left unprotected. If employees are unaware of the company policies on discussing sensitive company information, they will often volunteer (sometimes unknowingly) information about their company's future sales, marketing, or research plans simply by being asked the right set of questions. The tester will exploit this lack of awareness and modify the testing procedure to account for the fact that the policies are not well-known.

In many cases, the subjects of the test will be very familiar with the company's policies and the procedures for handling information. Despite this, however, penetration testing works because often people do not adhere to standardized procedures defined by the company's policies. Although the policies may say that system logs should be reviewed daily, most administrators are too busy to bother. Good administrative and security practices require that system configurations should be checked periodically to detect tampering, but this rarely happens. Most security policies indicate minimum complexities and maximum time limits for passwords, but many systems do not enforce these policies. Once the tester knows about these security procedural lapses, they become easy to exploit.

Many companies have disjointed operational procedures. The processes in use by one organization within a company may often conflict with the processes used by another organization. Do the procedures used by one application to authenticate users complement the procedures used by other applications, or are there different standards in use by different applications? Is the access security of one area of a company's network lower than that of another part of the network? Are log files and audit records reviewed uniformly for all systems and services, or are some systems monitored more closely than others? All these are examples of a lack of coordination between organizations and processes. These examples can be exploited by the tester and used to get closer to the goal of the test. A tester needs only to target the area with the lower authentication standards, the lower access security, or the lower audit review procedures in order to advance the test.

Many penetration tests succeed because people often do not pay adequate attention to the situations and circumstances in which they find themselves. The hacker's art of social engineering relies heavily on this fact.

Social engineering is a con game used by intruders to trick people who know secrets into revealing them. People who take great care in protecting information when at work (locking it up or encrypting sensitive data, for example) suddenly forget about those procedures when asked by an acquaintance at a party to talk about their work. Employees who follow strict user authentication and system change control procedures suddenly "forget" all about them when they get a call from the "Vice President of Such and Such" needing something done "right away." Does the "Vice President" himself usually call the technical support line with problems? Probably not, but people do not question the need for information, do not challenge requests for access to sensitive information even if the person asking for it does not clearly have a need to access that data, and do not compare the immediate circumstances with normal patterns of behavior.

Many companies rely on a single source for enabling an employee to prove identity, and often that source has no built-in protection. Most companies assign employee identification (ID) numbers to their associates. That number enables access to many services the company has to offer, yet is displayed openly on employee badges and freely given when requested. The successful tester might determine a method for obtaining or generating a valid employee ID number in order to impersonate a valid employee.

Many hackers rely on the anonymity that large organizations provide. Once a company grows beyond a few hundred employees, it becomes increasingly difficult for anyone to know all employees by sight or by voice. Thus, the IT and HR staff of the company need to rely on other methods of user authentication, such as passwords, key cards, or the above-mentioned employee ID number. Under such a system, employees become anonymous entities, identified only by their ID number or their password. This makes it easier to assume the identity of a legitimate employee or to use social engineering to trick people into divulging information. Once the tester is able to hide within the anonymous structure of the organization, the fear of discovery is reduced and the tester will be in a much better position to continue to test.

Another contributor to the successful completion of most penetration tests is the simple fact that most system administrators do not keep their systems up-to-date with the latest security patches and fixes for the systems under their control. A vast majority of system break-ins occur as a result of exploitation of known vulnerabilities — vulnerabilities that could have easily been eliminated by the application of a system patch, configuration change, or procedural change. The fact that system operators continue to let systems fall behind in security configuration means that testers will continuously succeed in penetrating their systems.

The tools available for performing a penetration test are becoming more sophisticated and more widely distributed. This has allowed even the novice hacker to pick up highly sophisticated tools for exploiting system weaknesses and applying them without requiring any technical background in how the tool works. Often these tools can try hundreds of vulnerabilities on a system at one time. As new holes are found, the hacker tools exploit them faster than the software companies can release fixes, making life even more miserable for the poor administrator who has to keep pace. Eventually, the administrator will miss something, and that something is usually the one hole that a tester can use to gain entry into a system.

Basic Attack Strategies

Every security professional who performs a penetration test will approach the task somewhat differently, and the actual steps used by the tester will vary from engagement to engagement. However, there are several basic strategies that can be said to be common across most testing situations.

First, do not rely on a single method of attack. Different situations call for different attacks. If the tester is evaluating the physical security of a location, the tester may try one method of getting in the building; for example walking in the middle of a crowd during the morning inrush of people. If that does not work, try following the cleaning people into a side door. If that does not work, try something else. The same method holds true for electronic attacks. If one attack does not work (or the system is not susceptible to that attack), try another.

Choose the path of least resistance. Most real attackers will try the easiest route to valuable information, so the penetration tester should use this method as well. If the test is attempting to penetrate a company's network, the company's firewall might not be the best place to begin the attack (unless, of course, the firewall was the stated target of the test) because that is where all the security attention will be focused. Try to attack lesser-guarded areas of a system. Look for alternate entry points; for example, connections to a company's business partners, analog dial-up services, modems connected to desktops, etc. Modern corporate networks have many more connection points than just the firewall, so use them to the fullest advantage.

Feel free to break the rules. Most security vulnerabilities are discovered because someone has expanded the limits of a system's capabilities to the point where it breaks, thus revealing a weak spot in the system. Unfortunately, most users and administrators concentrate on making their systems conform to the stated policies of the organization. Processes work well when everyone follows the rules, but can have unpredictable results when those rules are broken or ignored. Therefore, when performing a test attack, use an extremely long password; enter a 1000-byte URL into a Web site; sign someone else's name into a visitors log; try anything that represents abnormality or nonconformance to a system or process. Real attackers will not follow the rules of the subject system or organization — nor should the tester.

Do not rely exclusively on high-tech, automated attacks. While these tools may seem more "glamorous" (and certainly easier) to use, they may not always reveal the most effective method of entering a system. There are a number of "low-tech" attacks that, while not as technically advanced, may reveal important vulnerabilities and should not be overlooked. Social engineering is a prime example of this type of approach. The only tools required to begin a social engineering attack are the tester's voice, a telephone, and the ability to talk to people. Yet despite the simplicity of the method (or, perhaps, because of it), social engineering is incredibly effective as a method of obtaining valuable information.

"Dumpster diving" can also be an effective low-tech tool. Dumpster diving is a term used to describe the act of searching through the trash of the subject in an attempt to find valuable information. Typical information found in most Dumpsters includes old system printouts, password lists, employee personnel information, drafts of reports, and old fax transmissions. While not nearly as glamorous as running a port scan on a subject's computer, it also does not require any of the technical skill that port scanning requires. Nor does it involve the personal interaction required of social engineering, making it an effective tool for testers who may not be highly skilled in interpersonal communications.

One of the primary aims of the penetration tester is to avoid detection. The basic tenet of penetration testing is that information can be obtained from a subject without his or her knowledge or consent. If a tester is caught in the act of testing, this means, by definition, that the subject's defenses against that particular attack scenario are adequate. Likewise, the tester should avoid leaving "fingerprints" that can be used to detect or trace an attack. These fingerprints include evidence that the tester has been working in and around a system. The fingerprints can be physical (e.g., missing reports, large photocopying bills) or they can be virtual (e.g., system logs detailing access by the tester, or door access controls logging entry and exit into a building). In either case, fingerprints can be detected and detection can lead to a failure of the test.

Do not damage or destroy anything on a system unless the destruction of information is defined as part of the test and approved (in writing) by management. The purpose of a penetration test is to uncover flaws and weaknesses in a system or process — not to destroy information. The actual destruction of company information not only deprives the company of its (potentially valuable) intellectual property, but it may also be construed as unethical behavior and subject the tester to disciplinary or legal action. If the management of the organization wishes the tester to demonstrate actual destruction of information as part of the test, the tester should be sure to document the requirement and get written approval of the management involved in the test. Of course, in the attempt to "not leave fingerprints," the tester might wish to alter the system logs to cover the tester's tracks. Whether or not this is acceptable is an issue that the tester should discuss with the subject's management before the test begins.

Do not pass up opportunities for small incremental progress. Most penetration testing involves the application of many tools and techniques in order to be successful. Many of these techniques will not completely expose a weakness in an organization or point to a failure of an organization's security. However, each of these techniques may move the tester closer and closer to the final goal of the test. By looking for a single weakness or vulnerability that will completely expose the organization's security, the tester may overlook many important, smaller weaknesses that, when combined, are just as important. Real-life attackers can have infinite patience; so should the tester.

Finally, be prepared to switch tactics. Not every test will work, and not every technique will be successful. Most penetration testers have a standard "toolkit" of techniques that work on most systems. However, different systems are susceptible to different attacks and may call for different testing measures. The tester should be prepared to switch to another method if the current one is not working. If an electronic attack is not yielding the expected results, switch to a physical or operational attack. If attempts to circumvent a company's network connectivity are not working, try accessing the network through the company's dial-up connections. The attack that worked last time may not be successful this time, even if the subject is the same company. This may either be because something has changed in the target's environment or the target has (hopefully) learned its lesson

from the last test. Finally, unplanned opportunities may present themselves during a test. Even an unsuccessful penetration attempt may expose the possibility that other types of attack may be more successful. By remaining flexible and willing to switch tactics, the tester is in a much better position to discover system weaknesses.

Planning the Test

Before any penetration testing can take place, a clear testing plan must be prepared. The test plan will outline the goals and objectives of the test, detail the parameters of the testing process, and describe the expectations of both the testing team and the management of the target organization.

The most important part of planning any penetration test is the involvement of the management of the target organization. Penetration testing without management approval, in addition to being unethical, can reasonably be considered "espionage" and is illegal in most jurisdictions. The tester should fully document the testing engagement in detail and get written approval from management before proceeding. If the testing team is part of the subject organization, it is important that the management of that organization knows about the team's efforts and approves of them. If the testing team is outside the organizational structure and is performing the test "for hire," the permission of management to perform the test should be included as part of the contract between the testing organization and the target organization. In all cases, be sure that the management that approves the test has the authority to give such approval. Penetration testing involves attacks on the security infrastructure of an organization. This type of action should not be approved or undertaken by someone who does not clearly have the authority to do so.

By definition, penetration testing involves the use of simulated attacks on a system or organization with the intent of penetrating that system or organization. This type of activity will, by necessity, require that someone in the subject organization be aware of the testing. Make sure that those with a need to know about the test do, in fact, know of the activity. However, keep the list of people aware of the test to an absolute minimum. If too many people know about the test, the activities and operations of the target may be altered (intentionally or unintentionally) and negate the results of the testing effort. This alteration of behavior to fit expectations is known as the Hawthorne effect (named after a famous study at Western Electric's Hawthorne factory whose employees, upon discovering that their behavior was being studied, altered their behavior to fit the patterns they believed the testers wanted to see.)

Finally, during the course of the test, many of the activities the tester will perform are the very same ones that real-life attackers will use to penetrate systems. If the staff of the target organization discovers these activities, they may (rightly) mistake the test for a real attack and catch the "attacker" in the act. By making sure that appropriate management personnel are aware of the testing activities, the tester will be able to validate the legitimacy of the test.

An important ethical note to consider is that the act of penetration testing involves intentionally breaking the rules of the subject organization in order to determine its security weaknesses. This requires the tester to use many of the same tools and methods that real-life attackers use. However, real hackers sometime break the law or engage in highly questionable behavior in order to carry out their attacks. The security professional performing the penetration test is expected to draw the line between bypassing a company's security procedures and systems, and actually breaking the law. These distinctions should be discussed with management prior to the commencement of the test, and discussed again if any ethical or legal problems arise during the execution of the test.

Once management has agreed to allow a penetration test, the parameters of the test must be established. The testing parameters will determine the type of test to be performed, the goals of the tests, and the operating boundaries that will define how the test is run. The primary decision is to determine precisely what is being tested. This definition can range from broad ("test the ability to break into the company's network") to extremely specific ("determine the risk of loss of technical information about XYZ's latest product"). In general, more specific testing definitions are preferred, as it becomes easier to determine the success or failure of the test. In the case of the second example, if the tester is able to produce a copy of the technical specifications, the test clearly succeeded. In the case of the first example, does the act of logging in to a networked system constitute success, or does the tester need to produce actual data taken from the network? Thus, the specific criteria for success or failure should be clearly defined.

The penetration test plan should have a defined time limit. The time length of the test should be related to the amount of time a real adversary can be expected to attempt to penetrate the system and also the reasonable

lifetime of the information itself. If the data being attacked has an effective lifetime of two months, a penetration test can be said to succeed if it successfully obtains that data within a two-month window.

The test plan should also explain any limits placed on the test by either the testing team or management. If there are ethical considerations that limit the amount of "damage" the team is willing to perform, or if there are areas of the system or operation that the tester is prohibited from accessing (perhaps for legal or contractual reasons), these must be clearly explained in the test plan. Again, the testers will attempt to act as real-life attackers and attackers do not follow any rules. If management wants the testers to follow certain rules, these must be clearly defined. The test plan should also set forth the procedures and effects of "getting caught" during the test. What defines "getting caught" and how that affects the test should also be described in the plan.

Once the basic parameters of the test have been defined, the test plan should focus on the "scenario" for the test. The scenario is the position the tester will assume within the company for the duration of the test. For example, if the test is attempting to determine the level of threat from company insiders (employees, contractors, temporary employees, etc.), the tester may be given a temporary job within the company. If the test is designed to determine the level of external threat to the organization, the tester will assume the position of an "outsider." The scenario will also define the overall goal of the test. Is the purpose of the test a simple penetration of the company's computers or facilities? Is the subject worried about loss of intellectual property via physical or electronic attacks? Are they worried about vandalism to their Web site, fraud in their electronic commerce systems, or protection against denial-of-service attacks? All these factors help to determine the test scenario and are extremely important in order for the tester to plan and execute an effective attack.

Performing the Test

Once all the planning has been completed, the test scenarios have been established, and the tester has determined the testing methodology, it is time to perform the test. In many aspects, the execution of a penetration test plan can be compared to the execution of a military campaign. In such a campaign, there are three distinct phases: reconnaissance, attack, and (optionally) occupation.

During the reconnaissance phase (often called the "discovery" phase), the tester will generally survey the "scene" of the test. If the tester is planning a physical penetration, the reconnaissance stage will consist of examining the proposed location for any weaknesses or vulnerabilities. The tester should look for any noticeable patterns in the way the site operates. Do people come and go at regular intervals? If there are guard services, how closely do they examine people entering and leaving the site? Do they make rounds of the premises after normal business hours, and are those rounds conducted at regular times? Are different areas of the site occupied at different times? Do people seem to all know one another, or do they seem to be strangers to each other. The goal of physical surveillance is to become as completely familiar with the target location as possible and to establish the repeatable patterns in the site's behavior. Understanding those patterns and blending into them can be an important part of the test.

If an electronic test is being performed, the tester will use the reconnaissance phase to learn as much about the target environment as possible. This will involve a number of mapping and surveillance techniques. However, because the tester cannot physically observe the target location, electronic probing of the environment must be used. The tester will start by developing an electronic "map" of the target system or network. How is the network laid out? What are the main access points, and what type of equipment runs the network? Are the various hosts identifiable, and what operating systems or platforms are they running? What other networks connect to this one? Is dial-in service available to get into the network, and is dial-out service available to get outside?

Reconnaissance does not always have to take the form of direct surveillance of the subject's environment. It can also be gathered in other ways that are more indirect. For example, some good places to learn about the subject are:

- Former or disgruntled employees
- Local computer shows
- Local computer club meetings
- Employee lists, organization structures
- Job application handouts and tours
- Vendors who deliver food and beverages to the site

All this information will assist the tester in determining the best type of attack(s) to use based on the platforms and services available. For each environment (physical or electronic), platform, or service found during the reconnaissance phase, there will be known attacks or exploits that the tester can use. There may also be new attacks that have not yet made it into public forums. The tester must rely on the experience gained in previous tests and the knowledge of current events in the field of information security to keep abreast of possible avenues of attack.

The tester should determine (at least preliminarily) the basic methods of attack to use, the possible countermeasures that may be encountered, and the responses that may be used to those countermeasures.

The next step is the actual attack on the target environment. The attack will consist of exploiting the weaknesses found in the reconnaissance phase to gain entry to the site or system and to bypass any controls or restrictions that may be in place. If the tester has done a thorough job during the reconnaissance phase, the attack phase becomes much easier.

Timing during the attack phase can be critical. There may be times when the tester has the luxury of time to execute an attack, and this provides the greatest flexibility to search, test, and adjust to the environment as it unfolds. However, in many cases, an abundance of time is not available. This may be the case if the tester is attempting to enter a building in between guard rounds, attempting to gather information from files during the owner's lunch hour, or has tripped a known alarm and is attempting to complete the attack before the system's intrusion response interval (the amount of time between the recognition of a penetration and the initiation of the response or countermeasure) is reached. The tester should have a good idea of how long a particular attack should take to perform and should have a reasonable expectation that it can be performed in the time available (barring any unexpected complications).

If, during an attack, the tester gains entry into a new computer or network, the tester may elect to move into the occupation phase of the attack. Occupation is the term used to indicate that the tester has established the target as a base of operations. This may be because the tester wants to spend more time on the target gathering information or monitoring the state of the target, or the tester may want to use the target as a base for launching attacks against other targets. The occupation phase presents perhaps the greatest danger to the tester, because the tester will be exposed to detection for the duration of the time he or she is resident in the target environment. If the tester chooses to enter the occupation phase, steps should be taken to make the tester's presence undetectable to the greatest extent possible.

It is important to note that a typical penetration test may repeat the reconnaissance/attack/occupation cycle many times before the completion of the test. As each new attack is prepared and launched, the tester must react to the attack results and decide whether to move on to the next step of the test plan, or abandon the current attack and begin the reconnaissance for another type of attack. Through the repeated and methodical application of this cycle, the tester will eventually complete the test.

Each of the two basic test types — physical and electronic — has different tools and methodologies. Knowledge of the strengths and weaknesses of each type will be of tremendous help during the execution of the penetration test. For example, physical penetrations generally do not require an in-depth knowledge of technical information. While they may require some specialized technical experience (bypassing alarm systems, for example), physical penetrations require skills in the area of operations security, building and site operations, human nature, and social interaction.

The "tools" used during a physical penetration vary with each tester, but generally fall into two general areas: abuse of protection systems and abuse of social interaction. Examples of abuse of protection systems include walking past inattentive security guards, piggybacking (following someone through an access-controlled door), accessing a file room that is accidentally unlocked, falsifying an information request, or picking up and copying information left openly on desks. Protection systems are established to protect the target from typical and normal threats. Knowledge of the operational procedures of the target will enable the tester to develop possible test scenarios to test those operations in the face of both normal and abnormal threats.

Lack of security awareness on the part of the victim can play a large part in any successful physical penetration test. If people are unaware of the value of the information they possess, they are less likely to protect it properly. Lack of awareness of the policies and procedures for storing and handling sensitive information is abundant in many companies. The penetration tester can exploit this in order to gain access to information that should otherwise be unavailable.

Finally, social engineering is perhaps the ultimate tool for effective penetration testing. Social engineering exploits vulnerabilities in the physical and process controls, adds the element of "insider" assistance, and

combines it with the lack of awareness on the part of the subject that they have actually contributed to the penetration. When done properly, social engineering can provide a formidable attack strategy.

Electronic penetrations, on the other hand, generally require more in-depth technical knowledge than do physical penetrations. In the case of many real-life attackers, this knowledge can be their own or "borrowed" from somebody else. In recent years, the technical abilities of many new attackers seem to have decreased, while the high availability of penetration and attack tools on the Internet, along with the sophistication of those tools, has increased. Thus, it has become relatively simple for someone without a great deal of technical knowledge to "borrow" the knowledge of the tool's developer and inflict considerable damage on a target. There are, however, still a large number of technically advanced attackers out there with the skill to launch a successful attack against a system.

The tools used in an electronic attack are generally those that provide automated analysis or attack features. For example, many freely available host and network security analysis tools provide the tester with an automated method for discovering a system's vulnerabilities. These are vulnerabilities that the skilled tester may be able to find manually, but the use of automated tools provides much greater efficiency. Likewise, tools like port scanners (that tell the tester what ports are in use on a target host), network "sniffers" (that record traffic on a network for later analysis), and "war dialers" (that systematically dial phone numbers to discover accessible modems) provide the tester with a wealth of knowledge about weaknesses in the target system and possible avenues the tester should take to exploit those weaknesses.

When conducting electronic tests there are three basic areas to exploit: the operating system, the system configuration, and the relationship the system has to other systems. Attacks against the operating system exploit bugs or holes in the platform that have not yet been patched by the administrator or the manufacturer of the platform. Attacks against the system configuration seek to exploit the natural tendency of overworked administrators not to keep up with the latest system releases and to overlook such routine tasks as checking system logs, eliminating unused accounts, or improper configuration of system elements. Finally, the tester can exploit the relationship a system has with respect other systems to which it connects. Does it have a trust relationship with a target system? Can the tester establish administrative rights on the target machine through another machine? In many cases, a successful penetration test will result not from directly attacking the target machine, but from first successfully attacking systems that have some sort of "relationship" to the target machine.

Reporting Results

The final step in a penetration test is to report the findings of the test to management. The overall purpose and tone of the report should actually be set at the beginning of the engagement with management's statement of their expectation of the test process and outcome. In effect, what the tester is asked to look for will determine, in part, the report that is produced. If the tester is asked to examine a company's overall physical security, the report will reflect a broad overview of the various security measures the company uses at its locations. If the tester is asked to evaluate the controls surrounding a particular computer system, the report will most likely contain a detailed analysis of that machine.

The report produced as a result of a penetration test contains extremely sensitive information about the vulnerabilities the subject has and the exact attacks that can be used to exploit those vulnerabilities. The penetration tester should take great care to ensure that the report is only distributed to those within the management of the target who have a need-to-know. The report should be marked with the company's highest sensitivity label. In the case of particularly sensitive or classified information, there may be several versions of the report, with each version containing only information about a particular functional area.

The final report should provide management with a replay of the test engagement in documented form. Everything that happened during the test should be documented. This provides management with a list of the vulnerabilities of the target and allows them to assess the methods used to protect against future attacks.

First, the initial goals of the test should be documented. This will assist anyone who was not part of the original decision-making process in becoming familiar with the purpose and intent of the testing exercise. Next, the methodology used during the test should be described. This will include information about the types of attacks used, the success or failure of those attacks, and the level of difficulty and resistance the tester experienced during the test. While providing too much technical detail about the precise methods used may be overly revealing and (in some cases) dangerous, the general methods and procedures used by the testing team should be included in the report. This can be an important tool for management to get a sense of how easy or difficult it was for the testing team to penetrate the system. If countermeasures are to be put in place,

they will need to be measured for cost-effectiveness against the value of the target and the vulnerabilities found by the tester. If the test revealed that a successful attack would cost the attacker U.S.$10 million, the company might not feel the need for additional security in that area. However, if the methodology and procedures show that an attack can be launched from the Internet for the price of a home computer and an Internet connection, the company might want to put more resources into securing the target.

The final report should also list the information found during the test. This should include information about what was found, where it was found, how it was found, and the difficulty the tester had in finding it. This information is important to give management a sense of the depth and breadth of the security problems uncovered by the test. If the list of items found is only one or two items long, it might not trigger a large response (unless, of course, the test was only looking for those one or two items). However, if the list is several pages long, it might spur management into making dramatic improvements in the company's security policies and procedures.

The report should give an overall summary of the security of the target in comparison with some known quantity for analysis. For example, the test might find that 10 percent of the passwords on the subject's computers were easily guessed. However, previous research or the tester's own experience might show that the average computer on the Internet or other clients contains 30 percent easily guessed passwords. Thus, the company is actually doing better than the industry norm. However, if the report shows that 25 percent of the guards in the company's buildings did not check for employee badges during the test, that would most likely be considered high and be cause for further action.

The report should also compare the initial goals of the test to the final result. Did the test satisfy the requirements set forth by management? Were the results expected or unexpected, and to what degree? Did the test reveal problems in the targeted area, or were problems found in other unrelated areas? Was the cost or complexity of the tests in alignment with the original expectations of management?

Finally, the report should also contain recommendations for improvement of the subject's security. The recommendations should be based on the findings of the penetration test and include not only the areas covered by the test, but also ancillary areas that might help improve the security of the tested areas. For example, inconsistent system configuration might indicate a need for a more stringent change control process. A successful social engineering attempt that allowed the tester to obtain a password from the company's help desk might lead to better user authentication requirements.

Conclusion

Although it seems to parallel the activities of real attackers, penetration testing, in fact, serves to alert the owners of computers and networks to the real dangers present in their systems. Other risk analysis activities, such as automated port scanning, war dialing, and audit log reviews, tend to point out the theoretical vulnerabilities that might exist in a system. The owner of a computer will look at the output from one of these activities and see a list of holes and weak spots in a system without getting a good sense of the actual threat these holes represent. An effective penetration test, however, will show that same system owner the actual damage that can occur if those holes are not addressed. It brings to the forefront the techniques that can be used to gain access to a system or site and makes clear the areas that need further attention. By applying the proper penetration testing techniques (in addition to the standard risk analysis and mitigation strategies), the security professional can provide a complete security picture of the subject's enterprise.

The Self-Hack Audit

Stephen James

Payoff

As organizations continue to link their internal networks to the Internet, system managers and administrators are becoming increasingly aware of the need to secure their systems. The self-hack audit (SHA) is an approach that uses hacker methods to identify and eliminate security weaknesses in a network before they are discovered by a hacker. This article describes the most common hacker techniques that have allowed unauthorized persons to gain access to computer resources and provides steps for network administrators to improve network security.

Introduction

In today's electronic environment, the threat of being hacked is no longer an unlikely incident, occurring in a few unfortunate organizations. New reports of hacker incidents and compromised systems appear almost daily. As organizations continue to link their internal networks to the Internet, system managers and administrators are becoming increasingly aware of the need to secure their systems. Implementing basic password controls is no longer adequate to guard against unauthorized access to data. Organizations are now looking for more up-to-date techniques to assess and secure their systems. The most popular and practical technique emerging is the self-hack audit (SHA). The SHA is an approach that uses hacker methods to identify and eliminate security weaknesses in a network before they are discovered by a hacker.

This article provides a methodology for the SHA and presents a number of popular hacker techniques that have allowed hackers to penetrate various systems in the past. Each description is followed by a number of suggested system administration steps or precautions that should be followed to help prevent such attacks. Although some of the issues discussed are specific to UNIX systems, the concepts can be applied to all systems in general.

Objectives of the Self-Hack Audit

The basic objective of the SHA is to identify all potential control weaknesses that may allow unauthorized persons to gain access to the system. The network administrator must be familiar and use all known hacker techniques for overcoming system security. Depending on the nature of the audit, the objective may be either to extend a user's current levels of access (which may be no access)or to destroy (i.e., sabotage) the system.

Overview of the Self-Hack Audit Methodology

To perform a useful SHA, the different types of hackers must be identified and understood. The stereotype of a hacker as a brilliant computer science graduate sitting in a laboratory in a remote part of the world is a dangerous misconception. Although such hackers exist, the majority of security breaches are performed by staff members of the breached organization. Hackers can be categorized into four types:

- Persons within an organization who are authorized to access the system. An example may be a legitimate staff member in the Accounting department who has access to Accounts Payable application menu functions.

- Persons within an organization who are not authorized to access the system. These individuals may include personnel such as the cleaning staff.

- Persons outside an organization who are authorized to access the system. An example may be a remote system support person from the organization's software vendor.

- Persons outside an organization who are not authorized to access the system. An example is an Internet user in an overseas country who has no connection with the organization.

The objective of the SHA is to use any conceivable method to compromise system security. Each of the four hacker types must be considered to assess fully all potential security exposures.

Popular Hacker Techniques

The following sections describe the techniques most commonly used by hackers to gain access to various corporate systems. Each section discusses the hacker technique and proposes basic controls that can be implemented to help mitigate these risks. The network administrator should attempt each of these techniques and should tailor the procedures to suit the organization's specific environment.

Accessing the Log-in Prompt

One method of gaining illegal access to a computer system is through the Log-in prompt. This situation may occur when the hacker is physically within the facility or is attempting to access the system through a dial-in connection.

Physical Access.

An important step in securing corporate information systems is to ensure that physical access to computer resources is adequately restricted. Any internal or external person who gains physical access to a terminal is given the opportunity to attempt to sign on at the log-in prompt.

To reduce the potential for unauthorized system access by way of a terminal within the organization's facility, the network administrator should ensure that:

- Terminals are located in physically secure environments.

- Appropriate access control devices are installed on all doors and windows that may be used to access areas where computer hardware is located.

- Personal computers that are connected to networks are password-protected if they are located in unrestricted areas. A hacker trying to access the system would be required to guess a legitimate password before gaining access through the log-in prompt.

- Users do not write their passwords on or near their work areas.

Dial-in Access.

Another method of accessing the log-in prompt is to dial in to the host. Many “daemon dialers” are readily available on the Internet. These programs, when given a range of numbers to dial, can identify valid modem numbers. Once a hacker discovers an

organization's modem number, he or she can dial in and, in most cases, immediately gain access to the log-in prompt.

To minimize the potential for security violations by way of dial-in network access, the network administrator should ensure that:

- Adequate controls are in place for dial-in sessions, such as switching off the modem when not in use, using a call-back facility, or requiring an extra level of authentication, such as a one-time password, for dial-in sessions.
- The organization's logo and name are removed from the log-in screen so that the hacker does not know which system has been accessed.
- A warning message alerts unauthorized persons that access to the system is an offense and that their activities may be logged. This is a legal requirement in some countries.

Obtaining Passwords

Once the hacker has gained access to an organization's log-in prompt, he or she can attempt to sign on to the system. This procedure requires a valid user ID and password combination.

Brute Force Attacks.

Brute force attacks involve manual or automated attempts to guess valid passwords. A simple password guessing program can be written in approximately 60 lines of C code or 40 lines of PERL. Many password guessing programs are available on the Internet. Most hackers have a “password hit list,” which is a collection of default passwords automatically assigned to various system accounts whenever they are installed. For example, the default password for the guest account in most UNIX systems is “guest.”

To protect the network from unauthorized access, the network administrator should ensure that:

- All user accounts are password protected.
- Password values are appropriately selected to avoid guessing.
- Default passwords are changed once the system is installed.
- Failed log-in attempts are logged and followed up appropriately.
- User accounts are locked out after a predefined number of sign-on failures.
- Users are forced to select passwords that are difficult to guess.
- Users are forced to change their passwords periodically throughout the year.
- Unused user accounts are disabled.
- Users are educated and reminded regularly about the importance of proper password management and selection.

Password Cracking.

Most UNIX sites store encrypted passwords together with corresponding user accounts in a file called /etc/passwd. Should a hacker gain access to this file, he or she can simply run a password cracking program such as Crack. Crack works by encrypting a standard dictionary with the same encryption algorithm used by UNIX systems (called crypt). It then compares each encrypted dictionary word against the entries in the password file until it finds a match. Crack is freely available via an anonymous File Transfer Protocol from ftp.cert.org at/pub/tools/crack.

To combat the hacker's use of password-cracking software, the network administrator should ensure that:

- Encrypted passwords are stored in a shadow password file and that the file is adequately protected.
- All “weak” passwords are identified by running Crack against the password file.
- Software such as Npasswd or Passwd+ is used to force users to select passwords that are difficult to guess.
- Users do not write their passwords on or near their work environments.
- Only the minimum number of users have access to the command line to minimize the risk of copying the /etc/passwdfile.

Keystroke Logging.

It takes less than 30 seconds to type in a short script to capture sign-on sessions. A hacker can use a diskette to install a keystroke-logging program onto a workstation. Once this Trojan Horse is installed, it works in the background and captures every sign-on session, based on trigger key words. The hacker can read the captured keystrokes from a remote location and gain access to the system. This technique is very simple and almost always goes unnoticed.

To prevent a hacker's access to the system by way of a keystroke-logging program, the network administrator should ensure that:

- Privileged accounts (e.g., root) require one-time passwords.
- The host file system and individual users' workstations are periodically scanned for Trojan Horses that could include keystroke-logging programs.
- Adequate physical access restrictions to computer hardware are in place to prevent persons from loading Trojan Horses.

Packet Sniffing.

The Internet offers a wide range of network monitoring tools, including network analyzers and “packet sniffers.” These tools work by capturing packets of data as they are transmitted along a communications segment. Once a hacker gains physical access to a PC connected to a LAN and loads this software, he or she is able to monitor data as it is transferred between locations. Alternatively, the hacker can attach a laptop to a network port in the office and capture data packets.

Remembering that network traffic often is not encrypted, there is a high chance that the hacker will capture valid user account and password combinations, especially between the

hours of 8:00 a.m. and 9:00 a.m. Tcpdump is a tool for UNIX systems used to monitor network traffic and is freely available via an anonymous FTP from ftp.ee.lbl.gov at tcpdump2.2.1.tar.z.

To reduce the possibility of account and password leaks through packet sniffers, the network administrator should ensure that:

- Communications lines are segmented as much as practical.
- Sign-on sessions and other sensitive data are transmitted in an encrypted format by using software such as Kerberos.
- Privileged accounts (e.g., root) sign on using one-time passwords.
- Physical access to communications lines and computer hardware is restricted.

Social Engineering.

Hackers often select a user account that has not been used for a period of time (typically about two weeks) and ensure that it belongs to a user whom the administrator is not likely to recognize by voice. Hackers typically target accounts that belong to interstate users or users in another building. Once they have chosen a target, they assume a user's identity and call the administrator or the help desk, explaining that they have forgotten their passwords. In most cases, the administrator or help desk will reset passwords for the hackers over the telephone.

In an effort to keep the network safe from this type of infiltration, the network administrator should ensure that:

- All staff are regularly reminded and educated about the importance of data security and about proper password management.
- The organization has documented and controlled procedures for resetting passwords over the telephone.
- Staff do not fall prey to social engineering attacks. Staff members must be aware of the possibility that a hacker may misrepresent himself or herself as a member of the information systems department and ask for a password.

General Access Methods

Hackers use a variety of methods to gain access to a host system from another system.

Internet Protocol Address Spoofing.

In a typical network, a host allows other “trusted” hosts to communicate with it without requiring authentication (i.e., without requiring a user account and password combination). Hosts are identified as trusted by configuring files such as the .rhost and /etc/hosts.equiv files. Any host other than those defined as trusted must provide authentication before it is allowed to establish communication links.

Internet protocol (IP) spoofing involves an untrusted host connecting to the network and pretending to be a trusted host. This access is achieved by the hacker changing its IP number to that of a trusted host. In other words, the intruding host fools the host on the local network into not challenging it for authentication.

To avoid this type of security violation, the network administrator should ensure that:

- Firewalls and routers are appropriately configured so that they reject IP spoofing attacks.

- Only appropriate hosts are defined as trusted within /etc/hosts.equiv, and file permissions over this file are adequate.

- Only appropriate hosts are defined within users' /.rhost files. If practical, these files should be removed.

Unattended Terminals.

It is quite common to find user terminals left signed on and unattended for extended periods of time, such as during lunch time. Assuming that the hacker can gain physical access to users' work areas (or assuming that the hacker is an insider), this situation is a perfect opportunity for a hacker to compromise the system's security. A hacker may use an unattended terminal to process unauthorized transactions, insert a Trojan Horse, download a destructive virus, modify the user's .rhost file, or change the user's password so that the hacker can sign on later.

The network administrator can minimize the threat from access through unattended terminals by ensuring that:

- User sessions are automatically timed out after a predefined period of inactivity, or password protected screen savers are invoked.

- Users are regularly educated and reminded about the importance of signing off their sessions whenever they expect to leave their work areas unattended.

- Adequate controls are in place to prevent unauthorized persons from gaining physical access to users' work areas.

Writeable Set User ID Files.

UNIX allows executable files to be granted root privileges by making file permissions set user ID (SUID) root. Hackers often search through the file system to identify all SUID files and to determine whether they are writeable. Should they be writeable, the hacker can insert a simple line of code within the SUID program so that the next time it is executed, it will write to the /etc/passwd file and this will enable the hacker to gain root privileges. The following UNIX command will search for SUID root files throughout the entire file system: find / -user root -perm -4000 -print

The network administrator can reduce the possibility of illegal access through SUID files by ensuring that:

- Only the minimum number of programs are assigned SUID file permissions.

- Programs that are SUID are not writeable by users other than root.

- Executables defined within the system cron tables (especially the root cron table) are not writeable by users other than root because they are effectively SUID root.

Computer Emergency Response Team Advisories.

The Computer Emergency Response Team (CERT)issues advisories whenever a new security exposure has been identified. These exposures often allow unauthorized users to gain root access to systems. Hackers always keep abreast of the latest CERT advisories

to identify newly found bugs in system software. CERT can be accessed via an anonymous FTP at info.cert.org.

The network administrator should ensure that:

- All CERT advisories have been reviewed and acted on in a controlled and timely manner.
- Checksums are used to ensure the integrity of CERT patches before they are implemented.

Hacker Bulletin Boards.

The Internet has a large number of hacker bulletin boards and forums that act as an invaluable source of system security information. The most popular hacker bulletin board is the “2600” discussion group. Hackers from around the world exchange security information relating to various systems and often publish security sensitive information relating to specific organizations or hacker techniques relating to specific programs.

The network administrator should ensure that the organization's data security officer regularly reviews hacker bulletin boards to identify new techniques and information that may be relevant to the organization's system environment.

Internet Software.

The Internet offers a large number of useful tools, such as SATAN, COPS, and ISS, which can assist data security officers and administrators in securing computer resources. These tools scan corporate systems to identify security exposures. However, these tools are also available to hackers and can assist them in penetrating systems.

To identify and resolve potential security problems, the network administrator should ensure that:

- The latest version of each security program is obtained and run in a regular manner. Each identified exposure should be promptly resolved.
- The system is subject to regular security audits by both the data security officer and independent external consultants.

Conclusion

Hacker activity is a real and ongoing threat that will continue to increase as businesses connect their internal corporate networks to the Internet. This article has described the most common hacker techniques that have allowed unauthorized persons to gain access to computer resources. The self-hack audit is becoming an increasingly critical technique for identifying security weaknesses that, if not detected and resolved in a timely manner, could allow hackers to penetrate the corporate system. System administrators and data security officers should keep abreast of the latest hacker techniques by regularly reading all CERT publications and hacker bulletin boards.

Author Biographies

Stephen James

Stephen James is one of Australia's leading computer security experts, specializing in UNIX and Internet security as well as hacker studies. He is a senior consultant with Price Waterhouse (Sydney).

16
Penetration Testing

Chuck Bianco, FTTR, CISA, CISSP

Penetration testing is not a be-all, end-all for security. Organizations must first perform risk assessments that determine the components of sound security policies and procedures. After the development, approval, and installation of security policies, organizations should install several control mechanisms to measure the success or failure of the risk analysis and security systems. One such control is a properly constructed penetration test.

What Is a Penetration Test?

Penetration testing involves examining the security of systems and architectures. It reviews the effectiveness of the security of the organization's Internet presence. This includes all the holes and information that might damage the organization. The tester uses his creativity and resourcefulness to behave in the same manner as a hacker would.

The tester uses hacking tools and related techniques to challenge the efficiency and competence of the security design. The tester hopes to find problems before the hackers do and to recommend fixes and solutions to identified vulnerabilities. Although penetration testing assesses security from the Internet side or the organization's network, it is not a full security assessment or a guarantee that your site is secure.

It is only a complement to a full range of security measures. Your company should already have a complete security policy based on a risk analysis of the data and items you need to protect. If you do not have a security policy in place, you may choose to use penetration testing to assist you in writing the security policy.

The penetration test is simply another security tool to assist in protecting your company's assets. There are several different types of penetration tests, depending on the depth of the test and the threats measured. Both outsiders and employees or trusted third parties can launch attacks on the company. The testing may be broad-based or narrow, depending on risk assessments, the maturity of security policies, prior testing histories, etc.

You may wish to test your systems from internal attacks or develop specialized penetration tests later.

Why Do It?

Many institutions offer Internet banking and related E-commerce activities. Some offer services through service bureaus and others offer the services on institution-run transactional Web sites. All institutions should ensure that they use all systems in a safe and sound manner. Intruders hack both institutions and service bureaus. These hacks place the assets of the institution in peril. The FBI claims that almost 60 percent of all business sites have been the victims of unauthorized access. Some companies have lost money. Many have been the victims of a denial-of-service (DoS) attack, in which a hacker sends more information than your system can handle. This causes your system to slow down or stop working. Examiners and auditors frequently find that the institution does not know whether or not it has suffered a security breach. According to the Computer Emergency Response Team (CERT) and the U.S. Department of Energy Computer Incident Advisory Center (CIAC), hackers invaded more than 25,000 sites in 2001.

Intrusions can lead to loss of money, data, and productivity. Hackers, spies, and competitors can all steal, regardless of whether or not an intrusion occurs. For example, hackers can take advantage of bugs in Web sites

to gain unauthorized information. We have even discovered many examples where poorly designed Web sites allowed visitors access to unauthorized information. Therefore, even authorized visitors can copy information and can sell confidential customer information and strategic information to competitors. These attacks can damage the institution's reputation and expose it to legal action. The intruder can also install entrances for future activity, such as backdoors, Trojan horses, and program worms. A well-planned test reenacts all such actions. Penetration testing will normally provide evidence of exposures before they occur. In the case of found Trojan horses and viruses, it will act as a detective control.

Penetration testing not only improves security but it helps to train your staff about security breaches. It provides evidence of proper care and diligence in the event of lawsuits filed because of an intrusion. Moreover, penetration testing authenticates vendors' claims about their product features. We advise you to have the test performed by a disinterested third party. For example, if the tester recommends that you purchase his product after he completes the test, he may not recommend the most effective solution. He also may not find security weaknesses in his products. The testing must be impartial and provide a view of the entire security system.

All institutions that offer E-commerce products should perform annual penetration tests. In no way does this mean that an annual test is sufficient to ensure effective security. We believe that the institution should conduct such tests at least once per year and present the testing report of findings to the board of directors. However, the security plan must indicate how much penetration testing is sufficient. For many sites, an annual penetration test is the equivalent of having the security guard only check if someone locked the front gate after closing time about once a year. Many testers offer yearly contracts for regular testing, which most organizations find extremely helpful in keeping up with the number of exploits and holes published daily.

Institutions using service bureaus should insist on annual penetration testing of the service bureau. Ideally, the institution will take part in the penetration test. The service bureau should issue report findings to its client institutions. The institution should use this report to design a limited penetration test at the institution. An exception to this requirement occurs when the institution takes an active part in the penetration test of the service bureau.

Costs

Costs of such tests can vary from as little as $2000 for targeted tests to several hundred thousand dollars. The risk assessment or Standard of Due Care Study and your security policy determine the extent of the test and necessary costs. Institutions will include penetration testing costs in cost/benefit studies as part of the business analysis decision.

Limits

The institution should carefully design the scope of the penetration test to protect the company from inadvertent downtime and loss of business due to a successful intrusion during the test. While it may also be impractical to allow the tester to have access to production systems, testing does not have to be perilous if done at low traffic times.

While the tester may be limited because the employees know about the penetration test, this knowledge only hampers penetration testing if the tester is also attempting to measure human security controls. Some testers prefer that company personnel know about the test in advance, so that the employees can tighten security before testing. For example, weekly penetration tests will cause the employees to apply patches the moment they come out, rather than waiting for a penetration test report showing they are not doing their jobs. Moreover, professional testers will notify the company as soon as they find any high risks and have it fix them immediately. They will still include the risks in the report, but the tester does not leave the company at risk during the testing and report-writing time.

The company must take great care to carefully design the limits and scope of the penetration test; yet it must also allow the tester sufficient access to evaluate security effectiveness. The organization should define exactly what the tester can and cannot test. These requirements should go in the contract and be defined by IP addresses.

The test can include, but is not limited to, the following tools and techniques (see http://www.cccure.org/modules.php?name = Downloads&d_op = viewdownloaddetails&lid = 9&ttitle = Domain_1.zip for more detail):

- Network mapping and port scanning
- Vulnerability scanning
- Wardialing
- Sniffing
- Spoofing
- Session hijacking
- Various denial-of-service (DoS) and distributed DoS (DDoS) attacks
- Stack-based buffer overflows
- Password cracking
- Backdoors
- Trojan horses and rootkits

Disadvantages include the following:

- Penetration testing can cause severe line-management problems without the involvement of senior management.
- Penetration testing is a waste of time if it is the only security measure taken by the company.
- It is very expensive, especially if improperly planned.
- The tester can use the information he finds against you.

Who You Should Avoid

Your institution should never enlist a convicted felon to test your security system.

What You Should Tell the Tester

- You should provide your institution's legal company name and address as well as the name of a contact person who they can always contact (day or night).
- You should also provide the limits and scope of the testing without denying the tester the opportunity to use his creativity. However, you must ensure that you instruct the tester that the testing should not damage anything and to document any problems caused or found.
- You should detail what systems or networks are off-limits and during what hours the testing will take place. Some experts suggest that you handle this like a firewall — list what you will allow and prohibit everything else. Be prepared to pay extra for testing at strange hours. Ensure that you have qualified employees on site during those strange hours to reboot downed systems.
- You should also indicate if you own the transaction Web site or use an ISP.
- Specify whether you will allow social engineering attacks (deception, trickery, or coercion are at the heart of social engineering techniques). Many testers believe that social engineering attacks may do more harm than good because they affect employee morale. Therefore, you may wish to limit publication of the successful social engineering attacks or redact the names of employees the tester fooled into providing information.
- Specify whether you will allow DoS attacks. If you allow these attacks, schedule them for a non-operations time and have someone babysitting the network while the attack happens. However, never allow distributed denial-of-service attacks, as they involve other companies; they always bring systems down and harm your Internet service provider and all routers in between.
- Specify whether the tester will cover his tracks or leave evidence on the system, such as text messages. The tester should never leave a backdoor program in your system. You may decide that a report of areas where the tester could have entered is sufficient.
- Specify exactly what the purpose of the test is:
  - Is it to get into your system, provide proof of successful entrance, and stop?
  - Will the tester place something on your system, such as a file or message, as proof that he gained entrance to the system?

  - Will you authorize the tester to gain system administrator privileges that allow him unlimited access to accounts?
  - Should the tester gain access to files or e-mail?
  - The tester should collect data indirectly by doing research on the Internet. This is mandatory for a penetration test. The Internet presence measures the footprints your employees leave on the Internet.
- Ask the tester to provide a list of things he or she will do to facilitate the test.
- Will the social engineering attacks be limited strictly to remote attacks, such as phone calls to employees, or will the hacker also conduct them in person? (In-person attacks include reviewing information in trash receptacles, posing as maintenance personnel, service bureau personnel, or employees of the institution, following employees into secured areas (tailgating), etc.) Many experts believe that on-site penetration testing is really auditing. Some companies have their employees perform the on-site social engineering tests in conjunction with the outside tester. Social engineering can also include e-mailing employees or inviting them to visit a certain Web site.
- Require that the tester indicates in his report how he got the data and if he believes your site is secured against the top-20 tools currently available in the wild. Require that he give some examples of how he located these tools and which ones they are. It is not sufficient that your site is currently safe from the exploits these tools attempt. The tester should measure your network's response to each tool's unique signature or method. For example, some tools are poorly written and may accidentally bring down a network, even though that was not the intent of the tool. In this way, you determine if the tester just uses a commercial scanning tool, or if he really tries to hack into your system. Many experts believe that no one tool is more than 10 percent effective in penetration testing.

What You Should Not Tell the Tester

You should not provide technical information that a hacker would not know in advance, such as information regarding:

- Firewalls
- Routers
- Filters
- Concentrators
- Configuration rules

What You Should Do before You Finalize the Contract

- You should determine the vendor's policy on hiring:
  - Obtain proof of liability insurance
  - How long has the testing company been in business?
  - How long has the testing team been together?
  - Ask for a description of the vendor's testing procedures. Avoid vendors who will not explain their entire testing procedure.
- Ask the vendor how you will reach them during the testing process. Avoid vendors you cannot reach at any time during the test.
- Ask the vendors about the dangers of denial-of-service attacks. Avoid vendors who encourage denial-of-service attacks without telling you how dangerous they are.
- Ask for and insist on merit examples of past work.
- Ask the vendor for redacted examples of his final product. Avoid a vendor who will not supply specific examples of his final product.
- Demand that the vendor sign a nondisclosure agreement. Avoid vendors who refuse to do so.
- Avoid vendors who offer refunds on security tests in cases of "secure networks." Professional security testers operate as a service and will not offer refunds in most every case.
- Have your contract reviewed by your attorney before signing.

- Require copies of files and data that the tester is able to access during the attacks. Specify whether these outputs will be paper or digital. Ask for traffic dumps, logs, and raw data. The tester should also provide the IP address from which the test is coming.

What You Should Tell Your Staff

Try to limit the number of employees who know about the test to the technicians responsible for the networks and computer systems. Assign one employee as the Internal Trusted Agent (ITA). The tester and ITA will communicate with each other if needed during the test. Your employees should know that automated intrusion detection systems block out the tester's IP after a few seconds of scanning. They should not assume that all activity is part of the test. You could actually be under attack from a hacker. Ensure that the technicians know a scan is coming and from where.

What the Tester Should Provide at the Conclusion of the Test

The tester should provide both a brief executive summary (one or two pages) indicating test results, and a detailed listing of all findings and results and what methodology of attacks he used. He should indicate what weaknesses he found and include recommendations for improvement. He should write his report so that nontechnical people understand it. At a minimum, the report should include the following items:

- What could be tested
- What was tested
- When and from where the test happened
- The performance effects on the test, and vice versa
- A detailed executive summary in nontechnical terms that includes the good and bad
- The tools used for findings
- Information security findings
- Holes, bugs, and misconfigurations in technical detail with suggestions on fixing them
- Network map
- Any weaknesses discovered
- Passwords and logins discovered
- Specific firewall/router behavior findings against a list of attacks (not tools)

Your next move depends on his findings. If he finds many problems, you should begin by fixing the problems. You should also:

- Review all security policies and procedures.
- Ensure staff is trained in security.
- Determine if you need to conduct a full security assessment.
- Review corporate and disaster recovery planning.

Notes

1. *The Open Source Security Testing Methodology Manual*, by Peter Herzog, http:/www.isecom.com.

Acknowledgments

Many industry experts contributed to this chapter. Thanks to Chris Hare of Nortel Networks and Mike Hines of Purdue University. I am very grateful to those who made significant contributions. Hal Tipton of HFT Associates in Villa Park, California, and author of numerous IT security books; Clement Dupuis of CGI in Canada and moderator of the CISSP Open Study Guide Web Site; and Pete Herzog, moderator of the Open Source Security Testing Methodology Forum.

The contents of this document are my own and do not represent those of any government agency.

Domain 2
Telecommunications, Network, and Internet Security

The Telecommunications, Network, and Internet Security domain encompasses the structures, transmission methods, transport formats, and security measures used to provide integrity, availability, authentication, and confidentiality for transmissions over private and public communications networks and media.

Information technology has become ubiquitous due, in large part, to the extent of network connectivity. Telecommunication methodologies allow for the timely transport of information — from corner to corner, across the country, and around the globe. It is no surprise that this domain is one of the largest, because it encompasses the security of communications technologies, as well as the ever-expanding realms of the intranet, Internet and extranet.

Firewalls, which continue to play an important role in protecting an organization's perimeter, are explored in this domain. Firewalls are basically barriers between two networks that screen traffic, both inbound and outbound, and through a set of rules, allow or deny transmission connections. In this domain, we compare the multiple aspects of the filtering devices.

While perimeter firewalls provide some level of protection, an organization's information, e.g., electronic mail, must still flow into and outside of the organization. Unfortunately, keeping these communication channels open allows for potential compromise. This domain covers the potential vulnerabilities of the free flow of information, and the protection mechanisms and services available. The computer viruses of the late 1980s appear tame compared with the rogue code that is rampant today. The networked globe allows for speedy replication. Malicious programs that take advantage of the weaknesses (or functionality) of vendor systems, traverse the Internet at a dizzying speed. While companies are implementing defensive postures as fast as they can, in many instances, internal organizations lack the capacity or the tools to fortify their own infrastructures. In some cases, such as is documented in this domain, niche messaging vendors offer services to augment internal security, addressing threats such as e-mail spamming and malicious viruses. They also offer a 24 hour by 7 day monitoring capability and, in many instances, a pre-emptive notification capability, that many organizations cannot accommodate with internal resources.

One of the most successful means of protecting data in transit is the use of encapsulation and encryption employed in virtual private networking. In this domain, we explore the concepts and principles of virtual private networks (VPNs), which allow for the transfer of private information across the public networks while maintaining the security of the data. With benefits that include the ability to do secure business with partners, offer new channels for goods and service delivery, and reach new markets at reduced costs, VPNs hold great promise. In this domain, we look at ways to evaluate, deploy and leverage VPN technologies, as well as divulge the potential vulnerabilities inherent in those technologies.

Computer and communication technologies are rapidly evolving, devices are growing smaller and more functional at the same time, allowing the consumer more mobility, flexibility and agility. Nowhere is this more true than in the wireless space. Moreover, wireless networks are more cost-effective, since installing and configuring cable and connected devices are not required. The desire to have access to information without the need to tether someone to a wired device is becoming a corporate mandate. And yet, the wireless world has its own set of vulnerabilities. In this domain, we address securing the wireless environment, at the physical layer, on the local area network and over the Internet.

Contents

Section 2.2 Internet/Intranet/Extranet

Section 2.3 E-mail Security

Section 2.4 Secure Voice Communications

Section 2.5 Network Attacks and Countermeasures

17
Understanding SSL

Chris Hare, CISSP, CISA

Secure Socket Layer (SSL) is a common term in the language of the network. Users, administrators, and security professionals alike have come to learn the benefits of SSL. However, like so many technology elements, most do not understand how it works. This chapter examines what SSL is, how it works, and the role of certificates.

What Is SSL?

SSL is a method of authenticating both ends of a communication session and providing encryption services to prevent unauthorized access or modification of the data while in transit between the two endpoints. SSL is most commonly associated with protecting the data transferred in a Web browser session, although SSL is not limited to just a Web browser.

SSL is widely used in financial, healthcare, and electronic commerce applications. With the advent of SSL, users can now access banking records, make payments, and transfer funds through a financial institution's Web sites. Likewise, users can access healthcare information and even make online purchases from a favorite provider. All of this is possible without SSL; however, with the authentication and encryption capabilities, purchasers can provide their payment information immediately.

Aside from protecting Web-based transactions and other protocols, SSL is also being used to establish virtual private network (VPN) connections to a remote network.

Many network protocols in use today offer little or no protection of the data, allowing information to be transferred "in the clear." Consequently, confidentiality and integrity of the data processed in the protocol is a major concern for users and security professionals. Without additional protection, data protection is totally reliant upon the underlying network design, which itself is prone to problems.

The phenomenal growth of the Internet and its use for E-commerce, information sharing, government, and banking indicates more and more confidential information is being transferred over the Internet than ever before. SSL addresses the confidentiality issue by encrypting the data transmission between the client and server. Using encryption prevents eavesdropping of the communication. Additionally, the server is always authenticated to the client and the client may optionally authenticate to the server.

The intent of the SSL protocol was to provide higher-level protocols, such as Telnet, FTP, and HTTP, increased protection in the data stream. The protection is afforded by encapsulating the higher-level protocol in the SSL session. When establishing the connection between the client and the server, the SSL layer negotiates the encryption algorithm and session key, in addition to authenticating the server. The server authentication is performed before any data is transmitted, thereby maintaining the privacy of the session.

Developed by Netscape Communications Corporation, SSL was first proposed as an Internet Request for Comments Draft in 1994. Although never accepted as an Internet Standard by the IETF, SSL has been implemented in many commercial applications, and several open source implementations are available today.

Server Certificates

Enabling SSL requires that the application server be capable of accepting an SSL request and the existence of a server certificate. Without the server certificate, SSL is not available, even if the server is configured to offer

it. The server certificate contains both public and private key components. The public certificate is provided to the client during the SSL handshake and the private component is kept on the server to verify requests and information encrypted with the server's public certificate.

The process of generating an SSL certificate is beyond the scope of the discussion. However, SSL certificates are available from a variety of certificate providers as well as OpenSSL implementations.

The SSL Handshake

There are two major phases in the SSL handshake. The first establishes the connection and authenticates the server, and the second authenticates the client. During phase 1, the client initiates the connection with the SSL server by sending a CLIENT-HELLO message.

The CLIENT-HELLO Message

The CLIENT-HELLO message contains a challenge from the client and the client's cipher specifications. If the client attempts to establish a connection with the SSL server using any message other than CLIENT-HELLO, it must be considered an error by the server, which in turn refuses the SSL connection request.

Within the CLIENT-HELLO message, the client specifies the following information:

- The client's SSL version
- The available cipher specifications
- A session ID if one is present
- A challenge, used for authentication

The session ID is a unique identifier indicating that the client has previously communicated with the server. If the session ID is still in the client's and the server's cache, there is no need to generate a new master key, because both ends still have a session ID from a previous connection. If the session ID is not found, then a new master key is required.

Once the client has sent the CLIENT-HELLO message to the server, the client suspends while awaiting the corresponding SERVER-HELLO message.

The SERVER-HELLO Message

When the server receives the CLIENT-HELLO message, it examines the provided data before responding. The server examines the parameters in the client's request, specifically to verify that it will support one of the ciphers and the client's SSL version. If the server cannot, it responds with an ERROR message to the client.

If the server can support the client's SSL version and one or more of the provided ciphers, it responds with a SERVER-HELLO message. The response includes the following information:

- The server's signed certificate
- A list of bulk ciphers and specifications
- A connection ID
- A response for the supplied SESSION ID if provided by the server

The server's signed certificate contains the server's public key, which will be used later during the connection phase if the client generates a new master key. The server provides:

- The bulk ciphers and specifications so both ends of the connection can agree upon the cipher to use in the communication
- The connection ID, which is a randomly generated value used by the client and server for a single connection

The server uses the provided SESSION ID to see if the session ID is found in the server's cache. If the session ID is not found, the server provides its certificate, and cipher specifications back to the client. The client then determines if a new master key is needed to continue the communications.

The CLIENT-MASTER-KEY Message

The client determines if a new master key is required, based on the response from the server for the provided session ID. The requirement for a new master key is based on the server responding positively to the provided SESSION ID, meaning that the data is in the server's cache. If the SESSION ID is not in the server's cache, then a new master key is required.

Generating a New Master Key

If a new master key is needed, the client generates the new master key using the data provided by the server in the SERVER-HELLO message and sends the new master key back to the server using a CLIENT-MASTER-KEY message. The CLIENT-MASTER-KEY message contains the following elements:

- The selected cipher chosen from the list provided by the server
- Any elements of the master key in cleartext
- An element of the master key encrypted using the server's public key
- Any data needed to initialize the key algorithm

The client uses the public key provided in the server's certificate to encrypt the new master key. After the server has received the new master key, it decrypts it using the private key corresponding to the server certificate. The master key consists of two components, one of which is transmitted to the server in the clear, and the other that is sent encrypted. The amount of master-key data sent in the clear depends on the encryption cipher in use, as explained in the section entitled "Determining the Encryption Cipher" later in this chapter.

Keys and More Keys

If no new master key is required, both ends of the connection generate new session keys using the existing master key, the challenge provided by the client, and the connection ID provided by the server.

The client and server use the master key to generate the session key pairs for this session. There are a total of four session keys generated, two for each end of the communication, as shown in Exhibit 17.1.

The draft Internet Request for Comments (RFC) for SSL represents the master key as a function between the server and client portions of the communications exchange. That is to say, the keys are generated using the following method:

```
CLIENT-READ-KEY = HASH(MASTER-KEY, "0," CHALLENGE, CONNECTION-ID)
SERVER-WRITE-KEY = HASH(MASTER-KEY, "0," CHALLENGE, CONNECTION-ID)
CLIENT-WRITE-KEY = HASH(MASTER-KEY, "1," CHALLENGE, CONNECTION-ID)
SERVER-READ -KEY = HASH(MASTER-KEY, "1," CHALLENGE, CONNECTION-ID)
```

The elements of the function are:

- The HASH is the cipher-specific function used to generate the keys.
- MASTER-KEY is the master key already exchanged between the client and server.
- CHALLENGE is the challenge data provided by the client in the CLIENT-HELLO message.
- CONNECTION-ID is the connection identifier provided by the server in the SERVER-HELLO message.

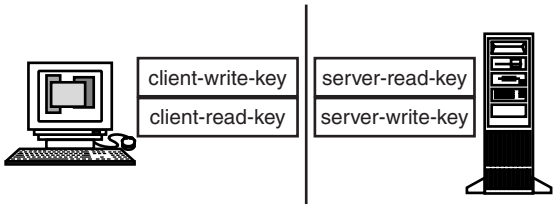


EXHIBIT 17.1 Two pairs of SSL keys are generated.

The "0" and "1" tell each side what key to generate. Notice the CLIENT-READ-KEY and the SERVER-WRITE-KEY both use the same "0" identifier. If they did not, the generated keys would not be related to each other and could not be used to encrypt and decrypt the data successfully. While the server is generating session keys, the client performs the same function, eliminating the need for key exchange across an untrusted network. The available ciphers are discussed later in the chapter.

The SERVER-VERIFY Message

Once the master key is decrypted, the server responds with a SERVER-VERIFY message. The SERVER-VERIFY response is sent after new session keys have been generated with an existing master key, or after the client has sent a specific CLIENT-MASTER-KEY request. Consequently, not every SSL handshake requires an explicit CLIENT-MASTER-KEY message.

The SERVER-VERIFY message contains an encrypted version of the challenge originally sent by the client in the CLIENT-HELLO message. Only the authentic server has the private key matching the certificate, the authenticity of the server has been validated, and only the authentic server can encrypt the challenge properly using the session keys. Consequently, these two actions verify the authenticity of the server. The transaction to this point is illustrated in Exhibit 17.2.

If the client and the server cannot agree on the ciphers to use in the communication, the client returns an ERROR message to the server.

Once the keys have been generated and the server responds with the SERVER-VERIFY message, the server has been verified and phase 2 is started.

Phase 2 consists of authenticating the client, as the server is authenticated in phase 1. The server sends a message to the client requesting additional information and credentials. The client then transmits them to the server or, if it has none, responds with an ERROR response. The server can ignore the error and continue, or stop the connection, depending on how the implementation is configured.

The CLIENT-FINISHED and SERVER-FINISHED Messages

When the client has finished authenticating the server, it sends a CLIENT-FINISHED message with the connection ID encrypted using the client's write key (client-write-key). However, both ends of the connection must continue to listen for and acknowledge other messages until they have both sent and received a FINISHED message. Only then has the SSL handshake completed (see Exhibit 17.3).

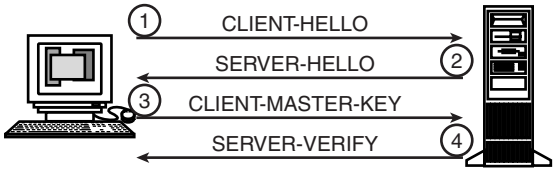


EXHIBIT 17.2 The SERVER-VERIFY message.

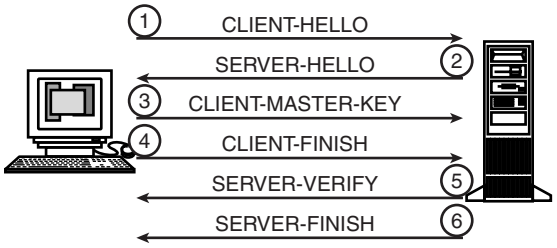


EXHIBIT 17.3 The full SSL handshake.

In most cases, the SSL handshake is completed without any further effort, as rarely does the server authenticate the client. Client authentication is typically through client certificates, which are discussed later in the chapter.

Determining the Encryption Cipher

The encryption cipher is negotiated between the client and the server, based upon the cipher specifications provided in the CLIENT-HELLO and SERVER-HELLO messages. The available ciphers are:

- RC4 and MD5
- 40-bit RC4 and MD5
- RC2 with CBC and MD5
- 40-bit RC2 with CBC and MD5
- IDEA with CBC and MD5

The MD5 128-bit key is not used in the encryption. The actual encryption algorithm used in the SSL data transfer is RC2, RC4, or IDEA, with key sizes ranging from 40 to 128 bits. The actual length of the encryption key depends on the cipher negotiation. The use of cryptography and specific key lengths is often controlled by international legislation, affecting the available ciphers.

While this is not an exhaustive list and other encryption protocols may be supported, the available ciphers offer protection of the data. However, the 40-bit ciphers operate differently. When using the RC4 and RC2 ciphers, the entire session key is sent encrypted between the client and the server. However, in SSL Version 1, the 40-bit ciphers were limited to a maximum key length of 40 bits. Consequently, it is possible for the client and the server not to have a cipher they can agree upon, meaning they cannot communicate.

With SSL Version 2, the key became 128 bits regardless of implementation. However, with the EXPORT40 implementations, only 40 bits of the session key are encrypted — the other 88 bits are not.

A discussion of the encryption algorithms used is beyond the scope of this discussion; the reader is urged to review the appropriate cryptography references for information on ciphers.

Client Certificates

Unlike server certificates that are involved in phase 1 of the SSL handshake, client certificates are part of phase 2. The REQUEST-CERTIFICATE and CLIENT-CERTIFICATE messages are used during phase 2.

Client certificates must be generated or acquired and installed in the application. The process of certification acquisition and installation is outside the scope of this discussion.

The REQUEST-CERTIFICATE Message

The REQUEST-CERTIFICATE message is sent from the server to the client when the server has been configured to require this authentication element. The message contains:

- The desired authentication type
- A challenge

The desired authentication types are:

```
SSL_AT_MD5_WITH_RSA_ENCRYPTION
```

This message requires that the client responds with a CLIENT-CERTIFICATE message (see the following section) by constructing an MD5 message digest of the challenge and encrypting it with the client's private key. The server can then validate the authenticity when the CLIENT-CERTIFICATE message is received by performing the same MD5 digest functions, decrypting the data sent using the client's public key, and comparing it with its own MD5 digest. If the values match, the client has been authenticated.

The CLIENT-CERTIFICATE Message

The CLIENT-CERTIFICATE message, sent in response to a REQUEST-CERTIFICATE from the server, provides the information for the server to authenticate the client. The CLIENT-CERTIFICATE message contains the following information:

- The certificate type
- The certificate data
- The response data

However, if the client has no certificate installed, the client provides a NO-CERTIFICATE-ERROR to the server, generally meaning that the connection is refused. The certificate type used on the client side is generally an X.509 signed certificate provided by an external certificate authority.

When assembling the response to the server, the client creates a digital signature of the following elements:

- The CLIENT-READ-KEY
- The CLIENT-WRITE-KEY
- The challenge data from the REQUEST-CERTIFICATE message
- The server's signed certificate from the SERVER-HELLO message

The digital signature is encrypted with the client's private key and transmitted to the server. The server can then verify the data sent and accept the authenticity if the data is valid.

Other authentication types can be used between the client and the server and can be added by either defining a new authentication type or by changing the algorithm identifier used in the encryption engines.

Message Flow

To clarify the discussion to this point, the following examples illustrate the message flow between the client and the server—the handshake. As is evident from discussing the various messages in the protocol, there are several variations possible in establishing the connection between the client and the server.

Session Identifier Available

This is the simplest example of message flows in the SSL transaction. It occurs when the client and the server have the session in their cache (see Exhibit 17.4).

1. The client initiates the connection and sends the CLIENT-HELLO message, which includes the challenge, session identifier, and cipher specifications.
2. The server responds with a SERVER-HELLO message and provides the connection identifier and server hit flag.
3. The client sends the server a CLIENT-FINISH message with the connection identifier and the client-write-key. Remember that the connection identifier is encrypted with the client-write-key.
4. The server provides the original challenge from the client encrypted with the server-write-key in the SERVER-VERIFY message.

And finally, the server transmits the SERVER-FINISH message with the session identifier encrypted with the server write key.

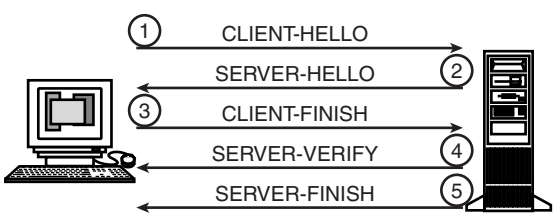


EXHIBIT 17.4 SSL session identifier available.

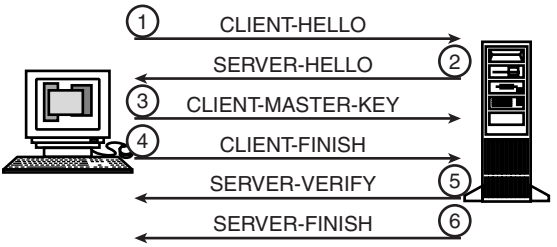


EXHIBIT 17.5 No session identifier.

No Session Identifier Available

This situation occurs when:

- The client has an identifier but the server does not.
- Neither the client nor the server has an identifier.

In this scenario (see Exhibit 17.5), the client connects and because there is no existing session identifier, the node must generate a new master key.

1. The client initiates the connection and sends the CLIENT-HELLO message, which includes the challenge and cipher specifications.
2. The server responds with a SERVER-HELLO message and provides the connection identifier, server certificate, and cipher specification.
3. The client selects the cipher, generates a new master key, and sends it to the server after encrypting it with the server's public key. This is the CLIENT-MASTER-KEY message.
4. The client sends the server a CLIENT-FINISH message with the connection identifier and the client-write-key. Remember that the connection identifier is encrypted with the client-write-key.
5. The server provides the original challenge from the client encrypted with the server-write-key in the SERVER-VERIFY message.

Finally, the server transmits the SERVER-FINISH message containing the new session identifier encrypted with the server-write-key.

The Entire Handshake Illustrated

This final example, shown in Exhibit 17.6, illustrates an SSL connection where the client must provide the new master key, new session keys are generated on both systems, and the server requests a client certificate.

1. The client initiates the connection and sends the CLIENT-HELLO message, which includes the challenge and cipher specifications.
2. The server responds with a SERVER-HELLO message and provides the connection identifier, server certificate, and cipher specification.
3. The client selects the cipher and generates a new master key, and sends it to the server after encrypting it with the server's public key. This is the CLIENT-MASTER-KEY message.
4. The client sends the server a CLIENT-FINISH message with the connection identifier and the client-write-key. Remember that the connection identifier is encrypted with the client-write-key.
5. The server provides the original challenge from the client encrypted with the server-write-key in the SERVER-VERIFY message.
6. The server sends the REQUEST-CERTIFICATE to the client, including the authentication type and challenge, encrypted with the server-write-key.
7. The client responds to the server, sending a CLIENT-CERTIFICATE message with the certificate type, the actual certificate, and the response to the challenge in the REQUEST-CERTIFICATE. All of the data is encrypted using the client-write-key.

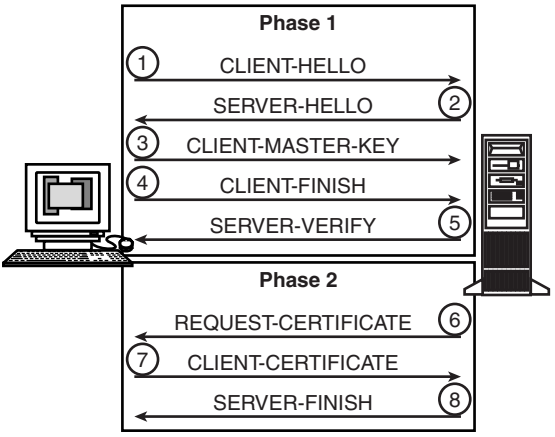


EXHIBIT 17.6 The complete SSL handshake.

Finally, the server transmits the SERVER-FINISH message containing the new session identifier encrypted with the server-write-key.

Is It All Encrypted?

The answer is no. Not all information during the handshake is actually sent encrypted, depending upon the phase of the handshake. Specifically, the following elements of the handshake are not encrypted:

- The CLIENT-HELLO message
- The SERVER-HELLO message
- The CLIENT-MASTER-KEY message
- The CLIENT-FINISHED
- SERVER-HELLO
- SERVER-FINISHED

Despite the messages that are not encrypted, sufficient information is sent in encrypted form so as to make it difficult to defeat. The encrypted messages include:

- SERVER-VERIFY
- CLIENT-CERTIFICATE
- REQUEST-CERTIFICATE

Depending on the situation, error messages can be encrypted or in cleartext, as described later in the chapter.

Once the session has been established, all further communications between the client and the server are encrypted.

Error Handling

Several errors can occur during the negotiations. These errors include:

- *NO-CIPHER-ERROR.* The client generates this error to the server indicating that there are no ciphers or key sizes supported by both ends of the connection. When this error occurs, the connection fails and cannot be recovered.
- *NO-CERTIFICATE-ERROR.* When the server requests a certificate from the client and there is no certificate available, the client returns this error message to the server. The server can choose to continue with the connection, depending on the local configuration.

- *BAD-CERTIFICATE-ERROR.* This error is generated when the certificate cannot be verified by the receiving party due to a bad digital signature or inappropriate information in the certificate. A common example of bad information in the certificate is when the host name in the certificate does not match the expected name. This error can be recovered and is not uncommon. Exhibit 17.7 illustrates the results when a Web client cannot verify a server certificate. The user is presented with a window similar to this, where he must choose to accept the certificate or not. Should the user choose not to accept the certificate, a window similar to that shown in Exhibit 17.8 would be shown to the user. The connection between the client and the server is not established.
- *UNSUPPORTED-CERTIFICATE-TYPE-ERROR.* Occasionally a server or client may receive a certificate that it does not have support for. This error is returned to the originating system.

After the Handshake

Once the handshake is complete, the client and the server exchange their messages using the services of the SSL transport. Because SSL allows higher-level protocols to protect their data while in transport, SSL has been used for a variety of purposes, including protecting HTTP-based traffic and SSL VPN sessions.

EXHIBIT 17. 7 Domain name mismatch error.

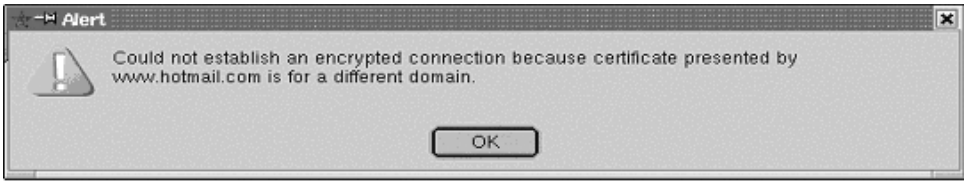


EXHIBIT 17.8 SSL connection is not established.

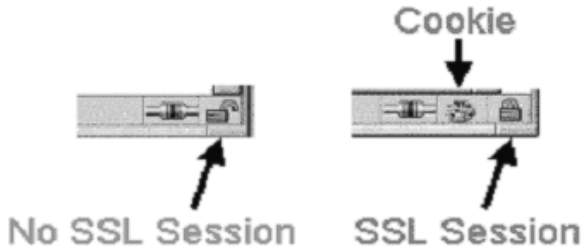


EXHIBIT 17.9 SSL on the Web.

SSL and the Web

The most well-known use of SSL is the protection of HTTP (World Wide Web) data when traveling across an untrusted network or carrying sensitive information. For example, E-commerce, secure online ordering, and bill payments are all performed on the Web using SSL as the protection layer.

The Web server must be capable of supporting SSL connections, and must have been properly configured with a server certificate, also known as a server-side certificate. The client specifies a Uniform Resource Locator (URL) with a https://prefix, indicating that the session is to be encapsulated within SSL.

The client contacts the Web server and the SSL handshake occurs. Once the SSL connection is established, the user sees a "key" or "lock" appear in the corner of their Web browser as seen in Exhibit 17.9.

Exhibit 17.9 illustrates a Web browser without an SSL connection, and the familiar lock indicating an SSL session has been established. Some Web servers will use SSL only for the specific transactions where protection is required, such as login forms, and credit card and E-commerce transactions.

SSL Tunnels

More recently, SSL has been used as the transport provider for virtual private networking. Commercial and open source software providers are including SSL VPN support in their products. One example is *stunnel*, an open source SSL VPN implementation for UNIX and Microsoft Windows-based systems.

SSL VPN solutions provide the same features as normal SSL applications, except the VPN implementation allows tunneling of non-SSL aware applications through the VPN to the target server or network. The VPN technology provides the encryption component, with no changes to the application required.

Attacking SSL

Like all network protocols and services, there are specific attacks that can be used against the SSL protocol or implementations of the protocol. Bear in mind that a weakness found in a specific implementation of the SSL protocol does not itself mean that SSL is flawed. What it means is that the implementation may be vulnerable to a specific attack or weakness, which does not inherently mean that all SSL implementations are vulnerable. For example, OpenSSL has been the subject of several attacks against its implementation of the protocol.

The attacks identified here do not constitute an all-inclusive list, but rather they represent some of the more commonly used attack methods that could be used to circumvent SSL.

Cipher Attacks

Because SSL uses several different technologies for the underlying encryption, attacks against the cryptographic engine or keys are inevitable. If a successful attack is found against any of the available cryptographic engines, SSL is no longer secure.

Consequently, any of the available methods of cryptographic analysis can be used. This includes recording a specific communications session and expending many CPU cycles to crack either the session or public key used.

Because many SSL sessions use 128-bit keys, the cost of launching an attack against a 128-bit key is still quite high. As new protocols and key lengths are supported within SSL, the work factor to defeat the cryptography increases.

Cleartext

Cleartext attacks are a fact of life with the SSL implementation. Because many messages in SSL are the same, such as HTTP GET commands, an attacker can build a dictionary where the entries are known values of specific words or phrases. The attacker then intercepts a session and compares the data in the session with the dictionary. Any match indicates the session key used and the entire data stream can be decrypted.

The work factor of the cleartext attack is quite high. For each bit added to the key, the dictionary size increases by a factor of two. This makes it virtually impossible to fabricate a dictionary with enough entries to defeat a 128-bit key using a cleartext attack methodology.

Given the high work factor associated with a cleartext attack, a brute-force attack, even with its high work factor, is considered the cheaper of the two. However, brute-force attacks also take an incredible amount of CPU horsepower and time. Even with today's high-speed computing equipment, the work factor associated with a brute-force attack against a 128-bit key is still considered an infinitely large problem.

Replay

Replay attacks involve the attacker recording a communication between the client and the server and later connecting to the server and playing back the recorded messages. While a replay attack is easy to originate, SSL uses a connection ID that is valid only for that connection. Consequently, the attacker cannot successfully use the recorded connection information. Because SSL uses a 128-bit value for the connection ID, an attacker would have to record at least 2^64 sessions to have a 50 percent chance of getting a valid session ID.

Man in the Middle

The man-in-the-middle attack (Exhibit 17.10) works by having the bad guy sit between the client and the server, with the attacker pretending to be the real server. By fooling the client into thinking it has connected to the real server, the attacker can decrypt the messages sent by the client, collect the data, and then retransmit it to the real server through an SSL session between the attacker and the real server.

The use of server certificates makes the man-in-the-middle attack more difficult. If the certificate is forged to match the real server's identity, the signature verification will fail. However, the attacker could create his or her own valid certificate, although it would not match the real server's name. If the certificate matches the attacker but does not match the name, the user will see a window in his browser similar to Exhibit 17.7. If the user ignores the message, and many do, he will not be aware of the connection problem.

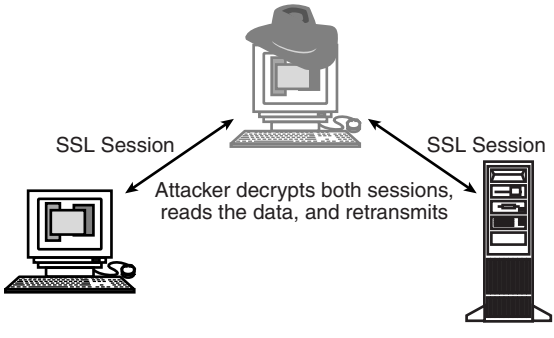


EXHIBIT 17.10 The man-in-the-middle attack.

Consequently, organizations would do well to inform their users of the connection problems and issues associated with SSL and teach them to report problems when they are encountered. It is far better to report a configuration error than to realize later that the data was compromised.

The Cost of Encryption

Encryption of any form has a cost in performance — SSL included. If the SSL server experiences a high level of traffic, then the server itself may suffer performance degradation due to the load of performing the SSL encryption and decryption. This performance degradation can be addressed in a number of ways.

The first possibility is to redesign the application to limit the actual amount of data that is transferred via SSL. For example, a Web application may only require SSL on specific pages, and by switching SSL on and off when required, the server's performance can be increased. The danger in this approach is the possibility for data that should be protected to be missed. Only a thorough analysis of the application, data, and data flows can determine where the application must be SSL protected.

The second solution is to change the system or network architecture and implement SSL accelerator hardware to offload the primary CPU from the actual SSL operations. SSL accelerator hardware can be installed into the actual server hardware or implemented in the network to perform the SSL handshake and all the encryption/decryption operations. While this can be a more expensive approach, it does not require any re-design or thorough analysis of the application. Because SSL accelerators are often implemented in an application layer switch, other benefits can be achieved, including load balancing.

Policy

Any organization providing information to others on either a public or private network will need to consider the requirements for SSL. Many situations where it is necessary to encrypt data on the public network may apply to the private network as well. Consequently, organizations must consider their security policy and assist in determining when SSL is required.

For example, SSL should be used on the public network to protect every transaction containing any form of personal information about the user, financial data, or information that the organization does not want generally visible on the public network. Additionally, SSL should be used on the private network to protect employee data and any information potentially subject to privacy legislation.

Finally, any information exchange falling into the realm of HIPAA, Gramm–Leech–Bliley, or Sarbanes–Oxley within the United States should strongly consider the use of SSL due to its data integrity properties. However, the specific legislation for a country and an organization's data classification and security policies will assist in determining when and where SSL is required.

Summary

This chapter has presented how the Secure Socket Layer encryption facility works. Focused at the protocol level, the security professional should understand how SSL actually functions and the number of steps involved in achieving the SSL connection. SSL is used as the basis for protecting almost all encrypted Web traffic to prevent the loss of sensitive information in an untrusted network. It can easily be stated that Internet based E-commerce would not be where it is today without SSL.

SSL provides data confidentiality and integrity elements in the handshake to avoid successful attacks, although there is a certain degree of human intervention and understanding associated with doing the correct thing when problems occur. Additionally, once the SSL session is established, data is protected in the session from eavesdropping and it cannot be altered during transmit — alterations cause the decryption to fail at the receiving end, maintaining the integrity of the data.

Consequently, organizations should make use of SSL encryption whenever they work with data across an untrusted network such as the Internet and consider using it to protect sensitive data within their own network, as the same network threats apply.

Acknowledgments

The author thanks Mignona Cote, a trusted friend and colleague, for her support during the development of this chapter. Mignona continues to provide ideas and challenges in topic selection and application, always with an eye for practical application of the information gained. Her insight into system and application controls serves her and her team effectively on an ongoing basis.

18

Packet Sniffers and Network Monitors

James S. Tiller, CISSP, CISA, and Bryan D. Fish, CISSP

Communications take place in forms that range from simple voice conversations to complicated manipulations of light. Each type of communication is based on two basic principles: wave theory and particle theory. In essence, communication can be established by the use of either, frequently in concert with a carrier or medium to provide transmission. An example is the human voice. The result of wave communications using the air as the signal-carrying medium is that two people can talk to each other. However, the atmosphere is a common medium, and anyone close enough to receive the same waves can intercept and surreptitiously listen to the discussion. For computer communications, the process is exponentially more complicated; the medium and type may change several times as the data is moved from one point to another. Nevertheless, computer communications are vulnerable in the same way that a conversation can be overheard. As communications are established, several vulnerabilities in the accessibility of the communication will exist in some form or another. The ability to intercept communications is governed by the type of communication and the medium that is employed. Given the proper time, resources, and environmental conditions, any communication — regardless of the type or medium employed — can be intercepted.

In the realm of computer communications, sniffers and network monitors are two tools that function by intercepting data for processing. Operated by a legitimate administrator, a network monitor can be extremely helpful in analyzing network activities. By analyzing various properties of the intercepted communications, an administrator can collect information used to diagnose or detect network performance issues. Such a tool can be used to isolate router problems, poorly configured network devices, system errors, and general network activity to assist in the determination of network design. In stark contrast, a sniffer can be a powerful tool to enable an attacker to obtain information from network communications. Passwords, e-mail, documents, procedures for performing functions, and application information are only a few examples of the information obtainable with a sniffer. The unauthorized use of a network sniffer, analyzer, or monitor represents a fundamental risk to the security of information.

This is a chapter in two parts. Part one introduces the concepts of data interception in the computer-networking environment. It provides a foundation for understanding and identifying those properties that make communications susceptible to interception. Part two addresses a means for evaluating the severity of such vulnerabilities. It goes on to discuss the process of communications interception with real-world examples. Primarily, this chapter addresses the incredible security implications and threats that surround the issues of data interception. Finally, it presents techniques for mitigating the risks associated with the various vulnerabilities of communications.

Functional Aspects of Sniffers

Network monitors and sniffers are equivalent in nature, and the terms are used interchangeably. In many circles, however, a network monitor is a device or system that collects statistics about the network.

Although the content of the communication is available for interpretation, it is typically ignored in lieu of various measurements and statistics. These metrics are used to scrutinize the fundamental health of the network.

On the other hand, a sniffer is a system or device that collects data from various forms of communications with the simple goal of obtaining the data and traffic patterns, which can be used for dark purposes. To alleviate any interpretation issues, the term "sniffer" best fits the overall goal of explaining the security aspects of data interception.

The essence of a sniffer is quite simple; the variations of sniffers and their capabilities are determined by the network topology, media type, and access point. Sniffers simply collect data that is made available to them. If placed in the correct area of a network, they can collect very sensitive types of data. Their ability to collect data can vary, depending on the topology and the complexity of the implementation, and is ultimately governed by the communications medium.

For computer communications, a sniffer can exist on a crucial point of the network, such as a gateway, allowing it to collect information from several areas that use the gateway. Alternatively, a sniffer can be placed on a single system to collect specific information relative to that system only.

Topologies, Media, and Location

There are several forms of network topologies, and each can use different media for physical communication.

Asynchronous Transfer Mode (ATM), Ethernet, Token Ring, and X.25 are examples of common network topologies that are used to control the transmission of data. Each uses some form of data unit packaging that is referred to as a frame or cell, and represents a manageable portion of the communication.

Coax, fiber, twisted-pair wire, and microwave are a few examples of computer communications media that can provide the foundation for the specific topology to transmit data units.

The location of a sniffer is a defining factor in the amount and type of information collected. The importance of location is relative to the topology and media being used. The topology defines the logical organization of systems on a network and how data is negotiated between them. The medium being utilized can assist in determining the environment simply based on its location. A basic example of this logical deduction is a simple Ethernet network spread across multiple floors in a building with a connection to the Internet. Ethernet is the topology at each floor and typically uses CAT5 cabling. Fiber cables can be used to connect each floor, possibly using FDDI as the topology. Finally, connection to the Internet typically consists of a serial connection using a V.35 cable. Using this deduction, it is safe to say that a sniffer with serial capabilities (logically and physically) placed at the Internet router can collect every packet to and from the Internet. It is also feasible to collect all the data between the floors if access to the FDDI network is obtained.

It is necessary to understand the relationship of the topology to the location and the environment, which can be affected by the medium. The medium being used is relevant in various circumstances, but this is inherently related to the location. Exhibit 18.1 explains in graphical format the relationship between the location of the sniffer, the topology, and the medium being used.

There are three buckets on the left of a scale at varying distances from the axis point, or moment. Bucket A, the furthest from the axis point, represents the weight that the sniffer's *location* carries in the success of the attack and the complexity of implementing a sniffer into the environment. Bucket A, therefore, provides greater leverage in the calculation of success relative to the difficulty of integration. Nearly equally important is the ***topology,*** represented by bucket B. Closer to the axis point, where the leverage is the least, is the ***medium*** represented by bucket C. Bucket C clearly has less impact on the calculation than the other two buckets.

Adding weight to a bucket is analogous to changing the value of the characteristic it represents. As the difficulty of the location, topology, or medium increases, more weight is added to the bucket. For example, medium bucket C may be empty if CAT5 is the available medium. The commonality of CAT5 and the ease of interacting with it without detection represents a level of simplicity. However, if a serial cable is intersected, the odds of detection are high and the availability of the medium in a large environment is limited; therefore, the bucket may be full. As the sophistication of each area is amplified, more weight is added to the corresponding bucket, increasing the complexity of the attack but enhancing the effectiveness of the assault.

This example attempts to convey the relationship between these key variables and the information collected by a sniffer. With further study, it is possible to move the buckets around on the bar to vary the impact each has on the scale.

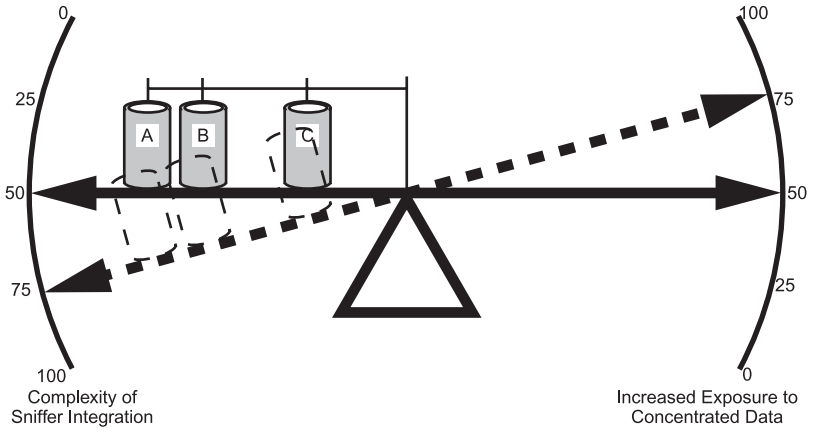


EXHIBIT 18.1 Location, topology, medium, and their relationship to the complexity of the sniffer-based attack and the information collected.

How Sniffers Work

As one would imagine, there are virtually unlimited forms of sniffers, as each one must work in a different way to collect information from the target medium. For example, a sniffer designed for Ethernet would be nearly useless in collecting data from microwave towers.

However, the volume of security risks and vulnerabilities with common communications seems to focus on standard network topologies. Typically, Ethernet is the target topology for local area networks (LANs) and serial is the target topology for wide area networks (WANs).

Ethernet Networks

The most common among typical networks are Ethernet topologies and IEEE 802.3, both of which are based on the same principle of Carrier-Sensing Multiple Access with Collision Detection (CSMA/CD) technology. Of the forms of communication in use today, Ethernet is one of the most susceptible to security breaches by the use of a sniffer. This is true for two primary reasons: installation base and communication type.

CSMA/CD is analogous to a conference call with several participants. Each person has the opportunity to speak if no one else is talking and if the participant has something to say. In the event two or more people on the conference call start talking at the same time, there is a short time during which everyone is silent, waiting to see whether to continue. Once the pause is over and someone starts talking without interruption, everyone on the call can hear the speaker. To complete the analogy, the speaker is addressing only one individual in the group, and that individual is identified by name at the beginning of the sentence.

Computers operating in an Ethernet environment interact in very much the same way. When a system needs to transmit data, it waits for an opportunity when no other system is transmitting. In the event two systems inject data onto the network at the same time, the electrical signals collide on the wire. This collision forces both systems to wait for an undetermined amount of time before retransmitting. The segment in which a group of systems participates is sometimes referred to as a collision domain, because all of the systems on the segment see the collisions. Also, just as the telephone was a common medium for the conference call participants, the physical network is a shared medium. Therefore, any system on a shared network segment is privy to all of the communications on that particular segment.

As data traverses a network, all of the devices on the network can see the data and act on certain properties of that data to provide communication services. A sniffer can reside at key locations on that network and inspect the details of that same data stream.

Ethernet is based on a Media Access Control (MAC) address, typically 48 bits assigned to the network interface card (NIC). This address uniquely identifies a particular Ethernet interface. Every Ethernet data frame contains the destination station's MAC address. As data is sent across the network, it is seen by every station on that segment. When a station receives a frame, it checks to see whether the destination MAC address of that frame is its own. As detailed in Exhibit 18.2, if the destination MAC address defined in the frame is that of the system, the data is absorbed and processed. If not, the frame is ignored and dropped.

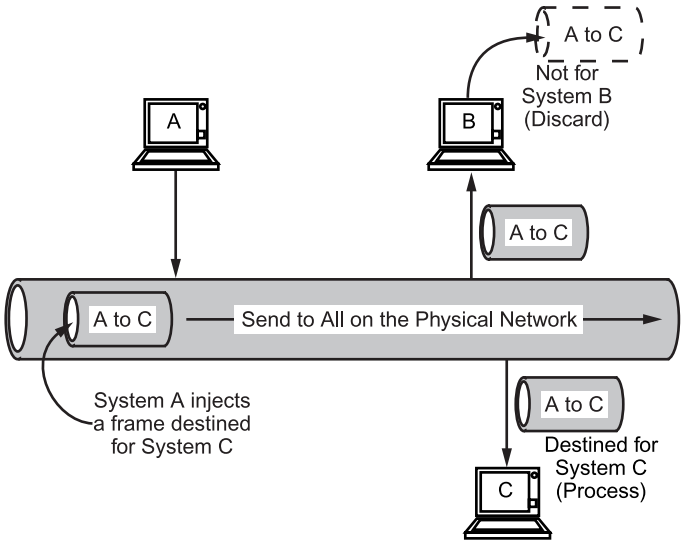


EXHIBIT 18.2 Standard Ethernet operations.

Promiscuous Mode

A typical sniffer operates in promiscuous mode. Promiscuous mode is a state in which the NIC accepts all frames, regardless of the destination MAC address of the frame. This is further detailed in Exhibit 18.3. The ability to support promiscuous mode is a prerequisite for an NIC to be used as a sniffer, as this allows it to capture and retain all of the frames that traverse the network.

For software-based sniffers, the installed NIC must support promiscuous mode to capture all of the data on the segment. If a software-based sniffer is installed and the NIC does not support promiscuous mode, the sniffer will collect only information sent directly to the system on which it is installed. This happens because the system's NIC only retains frames with its own MAC address.

For hardware-based sniffers — dedicated equipment whose sole purpose is to collect all data — the installed NIC must support promiscuous mode to be effective. The implementation of a hardware-based sniffer without the ability to operate in promiscuous mode would be nearly useless inasmuch as the device does not participate in normal network communications.

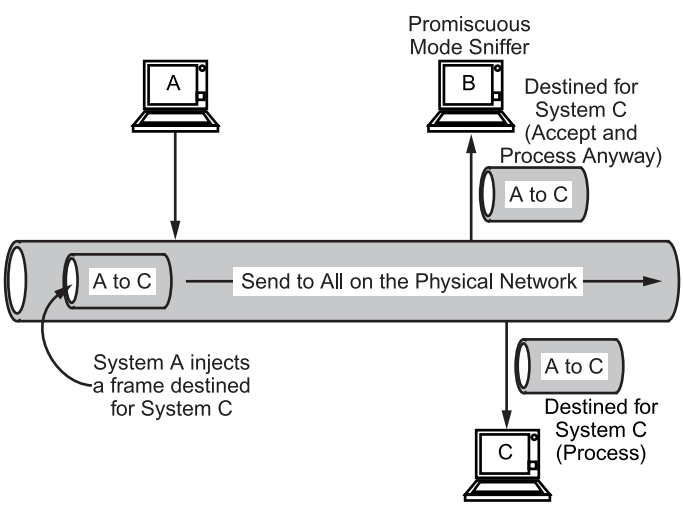


EXHIBIT 18.3 Promiscuous operations.

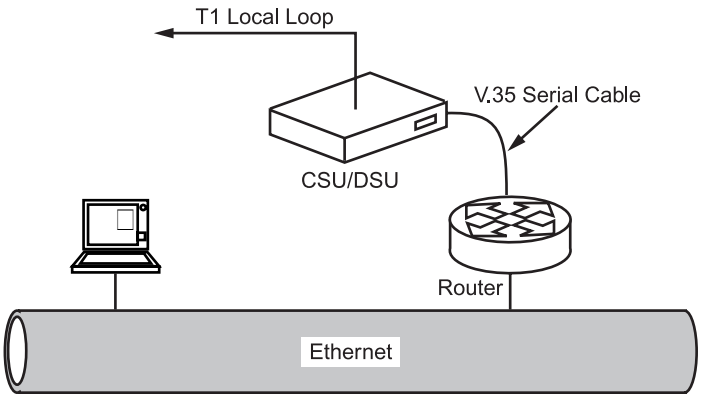


EXHIBIT 18.4 Common WAN connection.

There is an aspect of Ethernet that addresses the situation in which a system does not know the destination MAC address, or needs to communicate with all the systems of the network. A broadcast occurs when a system simply injects a frame that every other system will process. An interesting aspect of broadcasts is that a sniffer can operate in nonpromiscuous mode and still receive broadcasts from other segments. Although this information is typically not sensitive, an attacker can use the information to learn additional information about the network.

Wide Area Networks

Wide area network communications typify the relationship between topology, transmission medium, and location as compared with the level of access. In a typical Ethernet environment, nearly any network jack in the corner of a room can provide adequate access to the network for the sniffer to do its job. However, in some infrastructures, location can be a crucial factor in determining the effectiveness of a sniffer.

For WAN communications, the topology is much simpler. As a focal point device, such as a router processes data, the information is placed into a new frame and forwarded to a corresponding endpoint. Because all traffic is multiplexed into a single data stream, the location of the device can provide amazing access to network activities. Exhibit 18.4 illustrates a common implementation of WAN connectivity. However, the location is sensitive and not easily accessed without authorization.

One way the sniffer can gain access to the data stream is through a probe. A probe is an optional feature on some Channel Service Unit/Data Service Unit (CSU/DSU) devices; it is a device that provides connectivity between the customer premise equipment (CPE), such as a router, and the demarcation point of the serial line. As illustrated in Exhibit 18.5, a probe is implemented to capture all the frames that traverse the CSU/DSU.

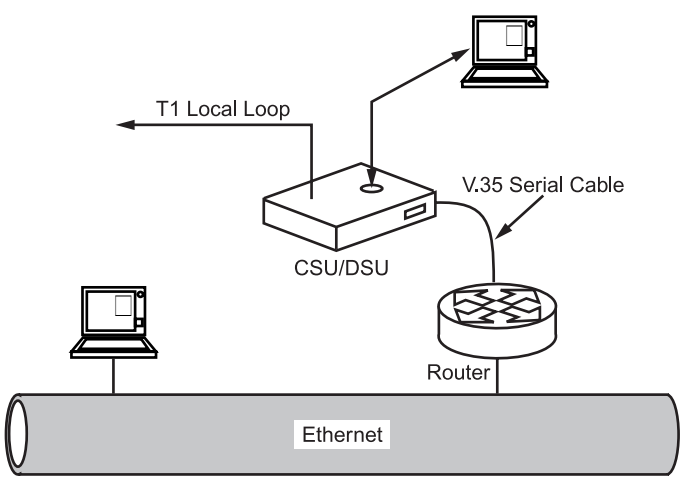


EXHIBIT 18.5 Sniffer probe used in a CSU/DSU.

Another way that the sniffer can gain access to the data stream is through a "Y" cable. A "Y" cable is connected between the CSU/DSU and the CPE. This is the most common location for a "Y" cable because of the complicated characteristics of the actual connection to the service provider's network, or local loop. Between the CSU/DSU and the CPE, a "Y" cable functions just like a normal cable. The third connector on the "Y" cable is free and can be attached to a sniffer. Once a "Y" cable is installed, each frame is electrically copied to the sniffer where it is absorbed and processed without disturbing the original data stream (see Exhibit 18.6). Unlike a probe, the sniffer installed with a "Y" cable must be configured for the topology being used. Serial communication can be provided by several framing formats, including Point-to-Point Protocol (PPP), High-Level Data Link Control (HDLC), and Frame Relay encapsulation. Once the sniffer is configured for the framing format of the topology — much as an Ethernet sniffer is configured for Ethernet frames — it can collect data from the communication stream.

Other Communication Formats

Microwave communications are typically associated with line-of-sight implementations. Each endpoint has a clear, unobstructed focal path to the other. Microwave is a powerful carrier that can be precisely focused to reduce unauthorized interaction. However, as shown in Exhibit 18.7, the microwaves can wash around the receiving dish, or simply pass through the dish itself. In either event, a sniffer can be placed behind one of the endpoint microwave dishes to receive some of the signal. In some cases, all the of the signal is available but weak, but it can be amplified prior to processing.

Wireless communications devices, such as cellular phones or wireless home telephones, are extremely susceptible to interception. These devices must transmit their signal through the air to a receiving station. Even though the location of the receiving station is fixed, the wireless device itself is mobile. Thus, signal transmission cannot rely on a line of sight, because a direct signal such as this would have to traverse a variety of paths during the course of a transmission. So, to enable wireless devices to communicate with the receiving station, they must broadcast their signal across a wide enough space to ensure that the device on the other end will receive some of the signal. Because the signal travels across such a wide area, an eavesdropper would have little trouble placing a device in a location that would receive the signal.

Security Considerations

Communication interception by unauthorized individuals represents the core concern for many aspects of information security. For information to remain private, the participants must be confident that the data is not being shared with others. However, this simple concept of communication protection is nearly impossible. All communications — especially those that utilize shared network links — have to be assumed to have a vulnerability to unauthorized interception and dissemination.

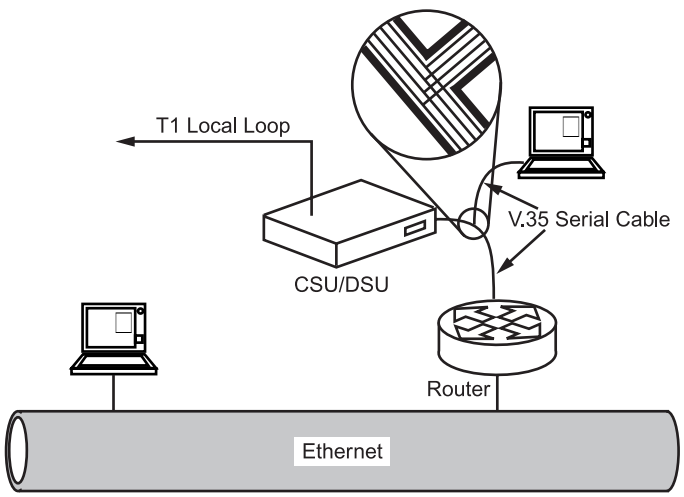


EXHIBIT 18.6 "Y" cable installation.

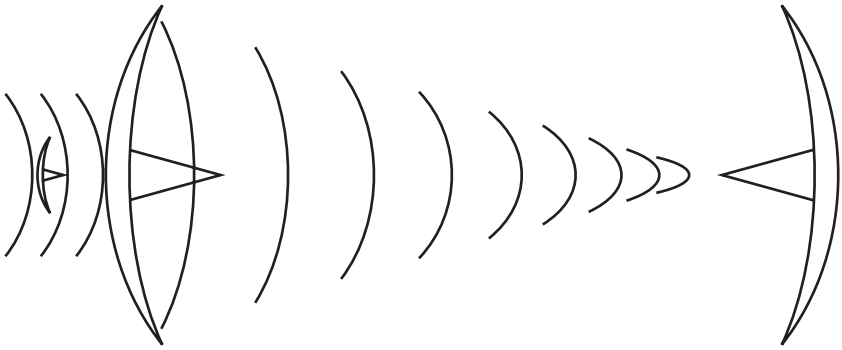


EXHIBIT 18.7 Microwave interception.

The availability of software-based sniffers is astounding. Combine the availability of free software with the fact that most modern NICs support promiscuous mode operations, and data interception becomes an expected occurrence rather than a novelty. Anyone with a PC, a connection to a network, and some basic, freely available software can wreak havoc on the security infrastructure.

The use of a sniffer as an attack tool is quite common, and the efforts of the attacker can be extremely fruitful. Even with limited access to remote networks that may receive only basic traffic and broadcasts, information about the infrastructure can be obtained to determine the next phase of the attack.

From an attacker's perspective, a sniffer serves one essential purpose: to eavesdrop on electronic conversations and gain access to information that would not otherwise be available. The attacker can use this electronic eavesdropper for a variety of attacks.

CIA

As elements of what is probably the most recognized acronym in the security industry, confidentiality, integrity, and availability (CIA) constitute the foundation of information security. Each one of these categories represents a vast collection of related information security concepts and practices.

Confidentiality corresponds to such concepts as privacy through the application of encryption in communications technology. Confidentiality typically involves ensuring that only authorized people have access to information. Integrity encompasses several aspects of data security that are to ensure that information has not had unauthorized modifications. The main objective of integrity is ensuring that data remains in the condition that was intended by the owner. In communications, the goal of integrity is to ensure that the data received has not been altered. The goal of availability is to ensure that information remains accessible to authorized users. Availability services do not attempt to distinguish between authorized and unauthorized users, but rely on other services to make that distinction. Availability services are designed to simply provide for the accessibility of the mechanisms and communication channels used to access information.

CIA embodies the core information security concepts that can be used to discuss the effectiveness of a sniffer. Sniffers can be used to attack these critical information properties directly, or to attack the mechanisms employed to guarantee these properties. An example of these mechanisms is authentication. Authentication is the process of verifying the identity of a user or resource so that a level of trust or access can be granted. Authentication also deals with verifying the source of a piece of information to establish the validity of that information. Authentication includes several processes and technologies to ultimately determine privileged access. Given the type of information exchange inherent in authentication, it has become a focal point for sniffer attacks. If an attacker obtains a password for a valid user name, other security controls may be rendered useless. This is also true for confidentiality and the application of encryption. If an attacker obtains the key being used to protect the data, it would be trivial to decrypt the data and obtain the information within.

Sniffer attacks expose any weakness in security technology and the application of that technology. As information is collected, various levels of vulnerabilities are exposed and acted upon to advance the attack. The goal of an attack may vary, but all of the core components of the security infrastructure must be functioning to reduce the risks.

This is highlighted by the interrelationship between the facets of CIA and the observation that, as one aspect fails, it may assist the attack in other areas. The goal of an attack may be attained if poor password protection is exploited or weak passwords are used that lead to the exposure of an encryption key. That key may have been used during a previous session that was collected by the sniffer. In that decrypted data may be instructions for a critical process that the attacker wishes to affect. The attacker can then utilize portions of data collected to reproduce the information, encrypt it, and retransmit it in a manner that produces the desired results.

Without adequate security, an attacker armed with a sniffer is limited only by his imagination. As security is added, the options available to the attacker are reduced but not eliminated. As more and more security is applied, the ingenuity and patience of the attacker is tested but not broken. The only real protection from a sniffer attack is not allowing one on the network.

Attack Methodologies

In various scenarios, a sniffer can be a formidable form of attack. If placed in the right location, a sniffer can be used to obtain proprietary information, or it can be used to gain information helpful in formulating a greater attack. In either case, information on a network can be used against the systems of that network.

There are many caveats regarding the level of success a sniffer can enjoy in a particular environment. Location is an obvious example. If the sniffer is placed in an area that is not privy to secret information, only limited data will be collected. Clearly, location and environment can have an impact on the type and amount of useful information captured. Therefore, attackers focus on specific concentrated areas of network activity in highly segmented networks.

Risks to Confidentiality

Confidentiality addresses issues of appropriate information disclosure. For information to remain confidential, systems and processes must ensure that unauthorized individuals are unable to access private information. The confidentiality implications introduced by a sniffer are clear. By surreptitiously absorbing conversations buried in network traffic, the attacker can obtain unauthorized information without employing conventional tactics. This contradicts the very definition of confidentiality.

Information security revolves around data and the protection of that data. Much of the information being shared, stored, or processed over computer networks is considered private by many of its owners. Confidentiality is fundamental to the majority of practicing information security professionals.

Encryption has been the obvious enabler for private exchanges, and its use dates back to Roman communications. Interestingly enough, computer communications are just now starting to implement encryption for confidentiality in communication domains that have traditionally been the most susceptible to sniffer attacks. Internal network communications, such as those within a LAN and WAN, do not utilize robust protection suites to ensure that data is not being shared with unauthorized individuals within the company. Terminal access emulation to a centralized AS/400 system is a prime example. Many companies have hundreds of employees accessing private data on centralized systems at banks, hospitals, insurance companies, financial firms, and government agencies. If the communication were to encroach onto an untrusted network, such as the Internet, encryption and data authentication techniques would not be questioned. Recently, the protection that has been normally afforded to external means of communication is being adopted for internal use because of the substantial risks that sniffers embody.

A properly placed sniffer would be privy to volumes of data, some of which may be open to direct interpretation. Internet services are commonly associated with the protection of basic private communications. However, at any point at which data is relayed from one system to another, its exposure must be questioned.

Ironically, the implementation of encryption can hinder the ultimate privacy. In a scenario in which poor communication encryption techniques are in use, the communication participants become overly trusting of the confidentiality of those communications. In reality, however, an attacker has leveraged a vulnerability in that weak encryption mechanism and is collecting raw data from the network. This example conveys the importance of properly implemented confidentiality protection suites. Confidentiality must be supported by tested and verified communication techniques that have considered an attack from many directions. This results in standards, guidelines, and best practices for establishing a trusted session with a remote system such that the data is afforded confidentiality. IPSec, PGP, SSL, SSH, ISAKMP, PKI, and S/MIME are only a few of the technologies that exist to ensure confidentiality on some level — either directly or by collateral effect. A sniffer can be employed to inspect every aspect of a communication setup, processing, and completion, allowing attackers to operate on the collected data at their leisure offline. This aspect of an offline attack on confidentiality

requires intensely robust communication standards to establish an encrypted session. If the standard or implementation is weak or harbors vulnerabilities, an attacker will defeat it.

Vulnerable Authentication Processes

Authentication deals with verification of the identity of a user, resource, or source of information and is a critical component in protecting the confidentiality, integrity, and availability of information. When one entity agrees to interact with another in electronic communications, there is an implicit trust that both parties will operate within the bounds of acceptable behavior. That trust is based on the fact that each entity believes that the other entity is, in fact, who it claims to be. Authentication mechanisms provide systems and users on a communication network with a reliable means for validating electronic claims of identity. Secure communications will not take place without proper authentication on the front end.

Trust is powerful in computer networking. If a user is trusted to perform a certain operation or process, she will be granted access to the system resources necessary to perform that function. Similarly, if a person is trusted in a communication session, the recipient of that person's messages will most likely believe what is being said. Trust, then, must be heavily guarded, and should not be granted without stringent proof of identity. Authentication mechanisms exist to provide that proof of identity.

Because authentication mechanisms govern trust, they are ripe targets for attack. If an attacker can defeat an authentication mechanism, he can virtually assume the identity of a trusted individual and immediately gain access to all of the resources and functions available to that individual. Even if the attacker gains access to a restricted user-level account, this is a huge first step that will likely lead to further penetration.

Sniffers provide an attacker with a means to defeat authentication mechanisms. The most straightforward example is a password sniffer. If authentication is based on a shared secret such as a password, then a candidate who demonstrates knowledge of that password will be authenticated and granted access to the system. This does a good job of guarding trust — until that shared secret is compromised. If an attacker learns the secret, he can present himself to the system for authentication and provide the correct password when prompted. This will earn him the trust of the system and access to the information resources inside.

Password sniffing is an obvious activity that can have an instant impact on security. Unless robust security measures are taken, passwords can be easily collected from a network. Most passwords are hashed or encrypted to protect them from sniffer-based attacks, but some services that are still heavily relied on do not protect the password. File Transfer Protocol (FTP), Telnet, and Hyper-Text Transfer Protocol (HTTP) are good examples of protocols that treat private information, such as usernames and passwords, as standard information and transmit them in the clear. This presents a significant threat to authentication processes on the network.

Communication Integrity

Integrity addresses inappropriate changes to the state of information. For the integrity of information to remain intact, systems and processes must ensure that an unauthorized individual cannot surreptitiously alter or delete that information. The implications of a sniffer on information integrity are not as clear-cut as those for confidentiality or authentication.

Sniffers are passive devices. However, by definition, an action is required to compromise the integrity of information. Sniffers and the information they procure are not always inherently valuable. The actions taken based on that information provide the real value either to an attacker or to an administrator. Sniffers, for example, can be used as part of a coordinated attack to capture and manipulate information and resubmit it, hence compromising its integrity. It is the sniffer's ability to capture the information in these coordinated attacks that allows the integrity of information to be attacked.

Session initiation provides an example. Essential information, such as protocol handshakes, format agreement, and authentication data, must be exchanged among the participants in order to establish the communication session. Although the attack is complicated, an attacker could use a sniffer to capture the initialization process, modify it, and use it later to falsify authentication. If the attacker is able to resend the personalized setup information to the original destination, the destination may believe that this is a legitimate session initialization request and allow the session to be established. In the event the captured data was from a privileged user, the copied credentials used for the attack could provide extensive access.

As with communications integrity, the threat of the sniffer from an availability standpoint is not direct. Because sniffers are passive devices, they typically do not insert even a single bit into the communication stream. Given this nature, a sniffer is poorly equipped to mount any form of denial-of-service (DoS) attack, the common name for attacks on resource availability. However, the sniffer can be used to provide important

information to a would-be DoS attacker, such as addresses of key hosts and network services or the presence server software versions known to be vulnerable to DoS attacks. The attacker can use this information to mount a successful DoS attack against the resource, thus compromising its availability.

While the primary target of a sniffer attack will not be the availability of information resources, the results of the attack can provide information useful in subsequent attacks on resource availability.

Growth over Time

Information collected by the attacker may not be valuable in and of itself. Rather, that information can be used in a learning process, enabling the attacker to gain access to the information or systems that are the ultimate targets.

An attacker can use a sniffer to learn useful pieces of information about the network, such as addresses of interesting devices, services and applications running on the various systems, and types of system activity. Each of these examples and many others can be combined into a mass of information that allows the attacker to form a more complete picture of the target environment and that ultimately assists in finding other vulnerabilities. Even in a well-secured environment, the introduction of a sniffer can amount to death by a thousand cuts. Information gathering can be quite dangerous, as seemingly innocuous bits of data are collected over time. The skilled attacker can use these bits of data to mount an effective attack against the network.

Attack Types

There are several types of sniffer attacks. These attacks are distinguishable by the network they target. The following sub-sections describe the various types of attacks.

LAN Based

As discussed throughout this chapter, LAN-based attacks represent the most common and easiest to perform attacks, and can reveal an amazing amount of private information. The proliferation of LAN sniffer attacks has produced several unique tools that can be employed by an attacker to obtain very specific data that pertains to the target environment. As a result of the commonality of Ethernet, tools were quickly developed to provide information about the status of the network. As people became aware of their simplicity and availability and the relative inability to detect their presence, these tools became a desired form of attack.

There are nearly infinite ways to implement a sniffer on a LAN. The level and value of the data collected is directly related to the location of the sniffer, network infrastructure, and other system vulnerabilities. As an example, it is certainly feasible that the attacker can learn a password to gain access to network systems from sniffing on a remote segment. Some network devices are configured by HTTP access, which does not directly support the protection of private information. As an administrator accesses the device, the attacker can easily obtain the necessary information to modify the configuration at a later time to allow greater access in the future.

Given the availability of sniffing tools, the properties of Ethernet, and the amount of unprotected ports in an office, what may appear to be a dormant system could actually be collecting vital information. One common method of LAN-based sniffer attack is the use of an inconspicuous, seemingly harmless system. A laptop can easily fit under a desk, on a bookshelf, in a box, or in the open; anywhere that network access can be obtained is a valid location. An attacker can install the laptop after-hours and collect it the next evening. The batteries may be exhausted by the next evening, but the target time is early morning when everyone is logging in and performing a great deal of session establishment. The attacker is likely to obtain many passwords and other useful fragments of information during this time.

Another aspect of LAN attacks is that the system performing the collection does not have to participate as a member of the network. To further explain, if a network is running TCP/IP as the protocol, the sniffer system does not need an IP address. As a matter of fact, it is highly desirable by the attacker not to obtain an IP address or interact with other network systems. Because the sniffer is interested only in layer 2 activities (i.e., frames, cells, or the actual packages defined by the topology), any interaction with layer 3, or protocol layer, could alert systems and administrators to the existence of an unauthorized system. Clearly, the fact that sniffers can operate autonomously increases the respect for the security implications of such a device.

WAN Based

Unlike a sniffer on a LAN, WAN-based attacks can collect information as it is sent from one remote network to another. A common WAN topology is Frame Relay (FR) encapsulation. The ability of an attacker to access

an FR cloud or group of configured circuits is limited, but the amount of information gained through such access is large.

There are three basic methods for obtaining data from a WAN, each growing in complexity but capable of collecting large amounts of private data. The first is access to the serial link between the router and the CSU/DSU, which was detailed earlier. Second, access to the carrier system would provide access not only to the target WAN but could conceivably allow the collection of data from other networks as well. This scenario is directly related to the security posture of the chosen carrier. It can be generally assumed that access to the carrier's system is limited and properly authenticated; however, it is not unheard of to find otherwise. The final form of access is to gather information from the digital line providing the Layer 1 connectivity. This can be accomplished, for example, with a fiber tap. The ability to access the provider's line is highly complicated and requires specialized tools in the proper location. This type of attack represents a typically accepted vulnerability, as the complexity of the attack reduces the risk associated with the threat. That is, if an attacker has the capability to intercept communications at this level, other means of access are more than likely available to the attacker.

Gateway Based

A gateway is a computer or device that provides access to other networks. It can be a simple router providing access to another local network, a firewall providing access to the Internet, or a switch providing virtual local area network (VLAN) segmentation to several networks. Nevertheless, a gateway is a focal point for network-to-network communications.

Installing a sniffer on a gateway allows the attacker to obtain information relative to internetworking activities, and in today's networked environments, many services are accessed on remote networks. By collecting data routed through a gateway, an attacker will obtain a great deal of data, with a high probability of finding valuable information within that data.

For example, Internet access is common and considered a necessity for doing business. E-mail is a fundamental aspect of Internet business activities. A sniffer installed on a gateway could simply collect all information associated with port 25 (SMTP). This would provide the attacker with volumes of surreptitiously gained e-mail information.

Another dangerous aspect of gateway-based attacks is simple neglect of the security of the gateway itself. A painful example is Internet router security. In the past few years, firewalls have become standard issue for Internet connectivity. However, in some implementations, the router that provides the connection to the Internet on the outside of the firewall is ignored. Granted, the internal network is afforded some degree of security from general attacks from the Internet, but the router can be compromised to gather information about the internal network indirectly. In some cases, this can be catastrophic. If a router is compromised, a privileged user's password could be obtained from the user's activities on the Internet. There is a strong possibility that this password is the same as that used for internal services, thus giving the attacker access to the inside network.

There are several scenarios of gateway-based sniffer attacks, each with varying degrees of impact. However, they all represent enormous potential to the attacker.

Server Based

Previously, the merits of traffic focal points as good sniffer locations were discussed. Given the type and amount of information that passes in and out of network servers, they become a focal point for sensitive information. Server-based sniffers take advantage of this observation, and target the information that flows in and out of the server. In this way, sniffers can provide ample amounts of information about the services being offered on the system and provide access to crucial information. The danger is that the attacker can isolate specific traffic that is relative to the particular system.

Common server-based sniffers operate much like normal sniffers in nonpromiscuous mode, capturing data from the NIC as information is passed into the operating system. An attacker can accomplish this type of sniffing with either of two basic methods: installing a sniffer, or using an existing one provided by the operating system.

It is well known that the majority of today's systems can be considered insecure, and most have various vulnerabilities for a number of reasons. Some of these vulnerabilities allow an attacker to obtain privileged access to the system. Having gained access, an attacker may choose to install a sniffer to gather more information as it is sent to the server. A good example is servers that frequently process requests to add users of various

services. Free e-mail services are common on the Internet, and in the event that users' passwords are gathered when they enroll, their e-mail will be completely accessible by an attacker.

By employing the system's existing utilities, an attacker needs only the necessary permissions to operate the sniffer. An example is tcpdump, described in detail later, which can be used by one user to view the activities of other users on the system. Improperly configured UNIX systems are especially vulnerable to these utility attacks because of the inherent nature of the multi-user operating environment.

Sniffer Countermeasures

A sniffer can be a powerful tool for an attacker. However, there are techniques that reduce the effectiveness of these attacks and eliminate the greater part of the risk. Many of these techniques are commonplace and currently exist as standards, while others require more activity on the part of the user.

In general, sniffer countermeasures address two facets of the attacker's approach: the ability to actually capture traffic, and the ability to use that information for dark purposes. Many countermeasures address the first approach, and attempt to prevent the sniffer from seeing traffic at all. Other countermeasures take steps to ensure that data extracted by the sniffer will not yield any useful information to the attacker. The following sub-sections discuss examples of both of these types of countermeasures.

Security Policy

Security policy defines the overall security posture of a network. Security policy is typically used to state an organization's position on particular network security issues. These policy statements are backed up by specific standards and guidelines that provide details on how an organization is to achieve its stated posture. Every organization should have a security policy that addresses its overall approach to security. A good security policy should address several areas that affect an attacker's ability to launch a sniffer-based attack.

Given physical access to the facility, it is easy to install a sniffer on most networks. Provisions in the security policy should limit the ability of an attacker to gain physical access to a facility. Denial of physical access to a network severely restricts an attacker's ability to install and operate a sniffer. Assuming an attacker does have physical access to a facility, provisions in the security policy should ensure that it is nontrivial to find an active but unused network port. A good security policy should also thoroughly address host security issues. Strong host security can prevent an attacker from installing sniffer software on a host already attached to the network. This closes down yet another avenue of approach for an attacker to install and operate a sniffer. Furthermore, policies that address the security of network devices help to deter gateway, LAN, and WAN attacks.

Policy should also clearly define the roles and responsibilities of the administrators who will have access to network sniffers. Because sniffing traffic for network analysis can easily lead to the compromise of confidential information, discretion should be exercised in granting access to sniffers and their output.

The following sections address point solutions that help to dilute the effectiveness of a sniffer-based attack. Security policy standards and guidelines should outline the specific use of these techniques.

Strong Authentication

It has been shown how password-based authentication can be exploited with the use of a sniffer. Stronger authentication schemes can be employed to render password-sniffing attacks useless. Password-sniffing attacks are successful, assuming that the attacker can use the sniffed password again to authenticate to a system. Strong authentication mechanisms ensure that the data seen on the network cannot be used again for later authentication. This defeats the password sniffer by rendering the data it captures useless.

Although certain strong authentication schemes can help to defeat password sniffers, they are not generally effective against all sniffer attacks. For example, an attacker sniffing the network to determine the version of Sendmail running on the mail server would not be deterred by a strong authentication scheme.

Encryption

Sniffer attacks are based on a fundamental security flaw in many types of electronic communications. The endpoints of a conversation may be extremely secure, but the communications channel itself is typically wide open, as many networks are not designed to protect information in transit. Encryption can be used to protect that information as it traverses various networks between the endpoints.

The process of encryption combines the original message, or plaintext, with a secret key to produce ciphertext. The definition of encryption provides that the ciphertext is not intelligible by an eavesdropper. Furthermore, without the secret key, it is not feasible for the eavesdropper to recover the plaintext from the ciphertext. These properties provide assurance that the ciphertext can be sent to the recipient without fear of compromise by an eavesdropper. Assuming the intended recipient also knows the secret key, she can decrypt the ciphertext to recover the plaintext and read the original message. Encryption is useful in protecting data in transit, because the ciphertext can be viewed by an eavesdropper, but is ultimately useless to an attacker.

Encryption protects the data in transit but does not restrict the attacker's ability to intercept the communication. Therefore, the cryptographic protocols and algorithms in use must themselves be resistant to attack. The encryption algorithm — the mathematical recipe for transforming plaintext into ciphertext — must be strong enough to prevent the attacker from decrypting the information without knowledge of the key. Weak encryption algorithms can be broken through a variety of cryptanalytic techniques. The cryptographic protocols — the rules that govern the use of cryptography in the communication process — must ensure that the attacker cannot deduce the encryption key from information made available during the conversation. Weak encryption provides no real security, only a false sense of confidence in the users of the system.

Switched Network Environments

Ethernet sniffers are by far the most commonly encountered sniffers in the wild. One of the reasons for this is that Ethernet is based on a shared segment. It is this shared-segment principle that allows a sniffer to be effective in an Ethernet environment; the sniffer can listen to all of the traffic within the collision domain.

Switches are used in many environments to control the flow of data through the network. This improves overall network performance through a virtual increase in bandwidth. Switches achieve this result by segmenting network traffic, which reduces the number of stations in an Ethernet collision domain. The fundamentals of Ethernet allow a sniffer to listen to traffic within a single collision domain. Therefore, by reducing the number of stations in a collision domain, switches also limit the amount of network traffic seen by the sniffer.

In most cases, servers reside on dedicated switched segments that are separate from the workstation switched networks. This will prevent a sniffer from seeing certain types of traffic. With the reduced cost of switches over the past few years, however, many organizations have implemented switches to provide a dedicated segment to each workstation. A sniffer in these totally switched environments can receive only broadcasts and information destined directly for it, missing out on all of the other network conversations taking place. Clearly, this is not a desirable situation for an attacker attempting to launch a sniffer-based attack.

Sniffers are usually deployed to improve network performance. The fact that sniffers heighten the security of the network is often a secondary consideration or may not have been considered at all. This is one of those rare cases in which the classic security/functionality paradox does not apply. In this case, an increase in functionality and performance on the network actually leads to improved security as a side effect.

Detecting Sniffers

The sniffer most commonly found in the wild is a software sniffer running on a workstation with a promiscuous Ethernet interface. Because sniffing is a passive activity, it is conceptually impossible for an administrator to directly detect such a sniffer on the network. It may be possible, however, to deduce the presence of a sniffer based on other information available within the environment. L0pht Heavy Industries has developed a tool that can deduce, with fairly high accuracy, when a machine on the network is operating its NIC in promiscuous mode. This tool is known as AntiSniff.

It is not generally possible to determine directly whether a machine is operating as a packet sniffer. AntiSniff uses deduction to form a conclusion about a particular machine and is quite accurate. Rather than querying directly to detect a sniffer, AntiSniff looks at various side effects exhibited by the operation of a sniffer. AntiSniff conducts three tests to gather information about the hosts on the network.

Most operating systems exhibit some unique quirks when operating an interface in promiscuous mode. For example, the TCP/IP stack in most early Linux kernels did not handle packets properly when operating in promiscuous mode. Under normal operation, the kernel behaves properly. When the stack receives a packet, it checks to see whether the destination MAC address is its own. If it is, the packet moves up to the next layer of the stack, which checks to see whether the destination IP address is its own. If it is, the packet is processed by the local system. However, in promiscuous mode, a small bug in the code produces abnormal results. In

promiscuous mode, when the packet is received, the MAC address is ignored and the packet is handed up the stack. The stack verifies the destination IP address and reacts accordingly. If the address is its own, it processes the packet. If not, the stack drops the packet. Either way, the packet is copied to the sniffer software.

There is a flaw, however, in this logic. Suppose station A is suspected of operating in promiscuous mode. AntiSniff crafts a packet, a ping for example, with a destination of station B's MAC address, but with station A's IP address. When station B receives the packet, it will drop it because the destination IP address does not match. When station A receives the packet, it will accept it because it is in promiscuous mode, so it will grab the packet regardless of the destination MAC address. Then, the IP stack checks the destination IP address. Because it matches its own, station A's IP stack processes the packet and responds to the ping. In nonpromiscuous mode, station A would have dropped the packet, because the destination MAC address was not its own. The only way the packet would have made it up the stack for processing is if the interface happened to be in promiscuous mode. When AntiSniff receives the ping reply, it can deduce that station A is operating in promiscuous mode.

This quirk is specific to early Linux kernels, but other operating systems exhibit their own quirks. The first AntiSniff test exercises the conditions that uncover those quirks in the various operating systems, with the intent to gain some insight as to whether the machine is operating in promiscuous mode.

Many sniffer-based attacks will perform a reverse-DNS query on the IP addresses it sees, in an attempt to maximize the amount of information it gleans from the network. The second AntiSniff test baits the alleged sniffer with packets destined for a nonexistent IP address and waits to see whether the machine does a reverse-DNS lookup on that address. If it does, chances are that it is operating as a sniffer.

A typical machine will take a substantial performance hit when operating its NIC in promiscuous mode. The final AntiSniff test floods the network with packets in an attempt to degrade the performance of a promiscuous machine. During this window of time, AntiSniff attempts to locate machines suffering from a significant performance hit and deduces that they are likely running in promiscuous mode.

AntiSniff is a powerful tool because it gives the network administrator the ability to detect machines that are operating as sniffers. This enables the administrator to disable the sniffer capability and examine the hosts for further evidence of compromise by an attacker. AntiSniff is the first tool of its kind, one that can be a powerful countermeasure for the network administrator.

Tools of the Trade

Sniffers and their ability to intercept network traffic make for an interesting conceptual discussion. However, the concept is not useful in the trenches of the internetworking battlefield until it is realized as a working tool. The power of a sniffer, in fact, has been incarnated in various hardware- and software-based tools. These tools can be organized into two general categories: those that provide a useful service to a legitimate network administrator, and those that provide an attacker with an easily operated, highly specialized attack tool. It should be noted that an operational tool that sees the entirety of network traffic can just as easily be used for dark purposes. The following sub-sections describe several examples of both the operational tools and the specialized attack tools.

Operational Tools

Sniffer operational tools are quite useful to the network administrator. By capturing traffic directly from the network, the tool provides the administrator with data that can be analyzed to discern valuable information about the network. Network administrators use operational tools to sniff traffic and learn more about how the network is behaving. Typically, an administrator is not interested in the contents of the traffic, only in the characteristics of the traffic that relate to network operation.

There are three primary types of operational tools, or utilities, that can be used for network monitoring or unauthorized activities. On the lower end of the scale are raw packet collectors — simple utilities that obtain various specifics about the communication but do not typically absorb the user data. These tools allow the user to see the communication characteristics for analysis, rather than providing the exact packet contents. For example, the operator can view the manner in which systems are communicating and the services being used throughout the network. Raw packet collectors are useful for determining basic communication properties, allowing the observer to draw certain deductions about the communication. The second, more common type of tool is the application sniffer. These are applications that can be loaded on a PC, providing several

layers of information to the operator. Everything from frame information to user data is collected and presented in a clear manner that facilitates easy interpretation. Typically, extended tools are provided for analyzing the data to determine trends in the communication. The last type of tool is dedicated sniffer equipment. Highly flexible and powerful, such equipment can be attached to many types of networks to collect data. Each topology and associated protocol that is supported by the device is augmented with analyzing functionality that assists in determining the status of the network. These tools provide powerful access at the most fundamental levels of communication. This blurs the line between network administration and unauthorized access to network traffic. Sniffers should be treated as powerful tools with tremendous potential for harm and good. Access to network sniffers should be tightly controlled to prevent individuals from crossing over that line.

Raw Packet Collectors

There are several variations of raw packet collectors, most of which are associated with UNIX systems. One example is tcpdump, a utility built into most variations of UNIX. It essentially makes a copy of everything seen by the kernel's TCP/IP protocol stack. It performs a basic level of packet decode, and displays key values from the packets in a tabular format. Included in the display is information such as the packet's timestamp, source host and port, destination host and port, protocol type, and packet size.

Snoop, similar to tcpdump, is another of the more popular utilities used in UNIX. These utilities do not wrap a graphical interface around their functionality, nor do they provide extended analytical information as part of their native feature set. The format used to store data is quite basic, however, and can be exported into other applications for trend analysis.

These tools can be very dangerous because they are easily operated, widely available, and can be started and stopped automatically. As with most advanced systems, separate processes can be started and placed in the background; they remain undetected while they collect vital information and statistics.

Application Sniffers

There are several commercial-grade products that are available to provide collection, analysis, and trend computations along with unprecedented access to user data. The most common operate on Microsoft platforms because of the market share Microsoft currently enjoys. Many examples exist, but Etherpeek, Sniffer, and Microsoft's own, NetMon are very common. Be assured there are hundreds of others, and some are even proprietary to certain organizations.

Each supports customizable filters, allowing the user to be selective about the type of packets saved by the application. With filters enabled, the promiscuous interface continues to capture every packet it sees, but the sniffer itself retains only those packets that match the filters. This allows a user to be selective and retain only those packets that meet certain criteria. This can be very helpful, both in network troubleshooting and launching attacks, as it significantly reduces the size of the packet capture while isolating specific communications that have known weaknesses or information. If either an administrator or an attacker is looking for something particular, having a much smaller data set is clearly an advantage.

By default, many application products display a summary listing of the packets as they are captured and retained. Typically, more information is available through packet decode capabilities, which allow the user to drill down into individual packets to see the contents of various protocol fields. The legitimate network administrator will typically stop at the protocol level, as this usually provides sufficient information to perform network troubleshooting and analysis. Packet decodes, however, also contain the packet's data payload, providing access to the contents of the communications. Access to this information might provide the attacker with the information he is looking for.

In addition to the ability to display vital detailed information about captured packets, many packages perform a variety of statistical analyses across the entire capture. This can be a powerful tool for the attacker to identify core systems and determine application flow. An example is an attacker who has enough access to get a sniffer on the network but is unaware of the location or applications that will allow him to obtain the desired information. By capturing data and applying statistical analysis, application servers can be identified, and their use, by volume or time of day, can be compared with typical business practices. The next time the sniffer is enabled, it can be focused on what appears to have the desired data to assist in expanding the attack.

Microsoft's NetMon runs as a service and can be configured to answer to polls from a central Microsoft server running the network monitor administrator. This allows an administrator to strategically place sniffers throughout the network environment and have them all report packet captures back to a central server. Although it is a powerful feature for the network administrator, the ability to query a remote NetMon sniffer also presents security concerns. For example, if an attacker cannot gain physical access to the network he wishes

to sniff but learns that NetMon is running, he may be able to attack the NetMon service itself, causing it to report its sniffer capture back to the attacker rather than to the legitimate administrative server. It is relatively simple to identify a machine running NetMon. An NBTSTAT –A/-a <IP/Name> command will provide a list of NetBIOS tags of the remote system. If a system tag of [BEh] is discovered, it indicates that the NetMon service is running on the remote system. Once this has been discovered, a sophisticated attacker can take advantage of the service and begin collecting information on a network that was previously inaccessible.

Dedicated Sniffer Equipment

Network General's Sniffer is the most recognized version of this type of tool; its existence is actually responsible for the term "sniffer." It is a portable device built to perform a single function: sniffing network traffic. Dedicated devices are quite powerful and have the ability to monitor larger traffic flows than could be accomplished with a PC-based sniffer. Additionally, dedicated devices have built-in interfaces for various media and topology types, making them flexible enough to be used in virtually any environment. This flexibility, while powerful, comes with a large price tag, so much so that dedicated equipment is not seen often in the wild.

Dedicated equipment supports advanced customizable filters, allowing the user to prune the traffic stream for particular types of packets. The sniffer is primarily geared toward capturing traffic, and allows the user to export the capture data to another machine for in-depth analysis.

Attack-Specific Tools

Staging a successful attack with an operational tool is often more an art than a science. Although there are many factors that determine the attacker's ability to capture network traffic, that is only half of the battle. The attacker must be able to find the proverbial needle in the haystack of packets provided by the sniffer. The attacker must understand internetworking protocols to decipher much of the information and must have a sound strategy for wading through the millions of packets that a sniffer might return.

Recent trends in computer hacking have seen the rise of scripted attacks and attacker toolkits. The Internet itself has facilitated the proliferation of hacking tools and techniques from the few to the many. Very few of the people who label themselves "hackers" actually understand the anatomy of the attacks they wage. Most simply download an exploit script, point it at a target, and pull the trigger. Simplifying the attack process into a suite of user-friendly software tools opens up the door to a whole new class of attacker.

Sniffer-based attacks have not escaped this trend. It can be argued that the information delivered by a sniffer does not provide any real value. It is what the attacker does with this information that ultimately determines the success of the sniffer-based attack. If this is true, then a sniffer in the hands of an unskilled attacker is probably of no use. Enter the attack-specific sniffer tool. Some of the talented few who understand how a sniffer's output can be used to launch attacks have bundled that knowledge and methodology into software packages.

These software packages are essentially all-in-one attack tools that leverage the information produced by a sniffer to automatically launch an attack. The following sub-sections present several examples of these attack-specific sniffer tools.

L0pht Crack Scanner

The L0pht Crack Scanner is produced by L0pht Heavy Industries, a talented group of programmers that specialize in security tools who operate on both sides of the network security battlefield. This tool is a password sniffer that exposes usernames and passwords in a Microsoft networking environment. L0pht Crack Scanner targets Microsoft's authentication processes, and uses mild algorithms to protect passwords from disclosure as they are sent across the network. This tool underscores the complementary role that a sniffer plays in many types of network attacks. The L0pht Crack Scanner combines a sniffer with a protocol vulnerability to attack the network, with drastic results.

The scanner capitalizes on several weaknesses in the authentication process to break the protection suites used, providing the attacker with usernames and passwords from the network. The individuals at L0pht have developed an algorithm to successfully perform cryptanalysis and recover the cleartext passwords associated with usernames.

The L0pht Crack Scanner uses a built-in sniffer to monitor the network, looking for authentication traffic. When the sniffer recognizes specific traffic, the packets are captured and the scanner applies L0pht's cryptanalysis routine and produces the password for the attacker.

PPTP Scanner

Microsoft's Point-to-Point Tunneling Protocol (PPTP) is a protocol designed to provide tunneled, encrypted communications. It has been proved that the encryption used in PPTP can be broken with simple cryptanalysis of the protocol. This cryptanalysis has been translated into a methodology for recovering traffic from a PPTP session.

PPTP Scanner combines a sniffer with a weakness in the design of PPTP, exposing and exercising a serious vulnerability. This vulnerability, when exercised, allows for the recovery of plaintext from an encrypted session.

PPTP Scanner is the incarnation of the PPTP cryptanalysis methodology. The Scanner uses built-in sniffer software to monitor network traffic, looking for a PPTP session. When PPTP traffic is recognized, the packets are captured and stored for analysis. The Scanner applies the cryptanalytic methodology, and recovers the plaintext traffic for the attacker.

Previously, we discussed the use of encryption to protect network traffic from sniffer-based attacks. The ease with which L0pht Crack Scanner and PPTP Scanner do their dirty work underscores an important point. Simply encrypting traffic before sending it across the network affords only limited protection. For this technique to provide any real security, the encryption algorithms and protocols chosen must be strong and resistant to attack.

Hunt

Hunt, an automated session hijack utility, is another example of a sniffer with a built-in attack capability. Hunt operates by examining network traffic flow for certain signatures — distinct traffic patterns that indicate a particular event or condition. When Hunt recognizes a signature for traffic it can work with, it springs into action. When Hunt goes active, it knocks one station offline, and assumes its identity in the ongoing TCP session. In this manner, Hunt hijacks the TCP session for itself, giving the operator access to an established connection that can be used to further explore the target system.

This capability can be quite useful to an attacker, especially if Hunt hijacks a privileged session. Consider the following example. If Hunt detects the traffic signature for a Telnet session that it can hijack, it will knock the originating station offline and resume the session itself. This gives the Hunt operator instant command-line access to the system. The attacker will be able to access the system as the original user, which could be anyone from a plain user to a system administrator.

Conclusion

Network sniffers exist primarily to assist network administrators in analyzing and troubleshooting their networks. These devices take advantage of certain characteristics of electronic communications to provide a window of observation into the network. This window provides the operator with a clear view into the details of network traffic flow.

In the hands of an attacker, a network sniffer can be used to learn many types of information. This information can range from basic operational characteristics of the network itself to highly sensitive information about the company or individuals who use the network. The amount and significance of the information learned through a sniffer-based attack are dependant on certain characteristics of the network and the attacker's ability to introduce a sniffer. The type of media employed, the topology of the network, and the location of the sniffer are key factors that combine to determine the amount and type of information seen by the sniffer.

Information security practitioners are committed to the pursuit of confidentiality, integrity, and availability of resources, as well as information in computing and electronic communications. Sniffers represent significant challenges in each of these arenas. As sniffer capabilities have progressed, so have the attacks that can be launched with a sniffer. The past few years have seen the evolution of easy-to-use sniffer tools that can be exercised by attackers of all skill levels to wage war against computing environments and electronic communications. As attackers have increased their capabilities, so have network administrators seeking to protect themselves against these attacks. The security community has responded with a myriad of techniques and technologies that can be employed to diminish the success of the sniffer-based attack.

As with most competitive environments, security professionals and system attackers continue to raise the bar for one another, constantly driving the other side to expand and improve its capabilities. This creates a seemingly endless chess match, in which both sides must constantly adjust their strategy to respond to the moves made by the other. As security professionals continue to improve the underlying security of computing and communications systems, attackers will respond by finding new ways to attack these systems. Similarly,

as attackers continue to find new vulnerabilities in computer systems, networks, and communications protocols, the security community will respond with countermeasures to combat these risks.

19

Secured Connections to External Networks

Steven F. Blanding

A private network that carries sensitive data between local computers requires proper security measures to protect the privacy and integrity of the traffic. When such a network is connected to other networks, or when telephone access is allowed into that network, the remote terminals, phone lines, and other connections become extensions of that private network and must be secured accordingly. In addition, the private network must be secured from outside attacks that could cause loss of information, breakdowns in network integrity, or breaches in security.

Many organizations have connected or want to connect their private local area networks (LANs) to the Internet so that their users can have convenient access to Internet services. Because the Internet as a whole is not trustworthy, their private systems are vulnerable to misuse and attack. Firewalls are typically used as a safeguard to control access between a trusted network and a less trusted network. A firewall is not a single component; it is a strategy for protecting an organization's resources from the Internet. A firewall serves as the gatekeeper between the untrusted Internet and the more trusted internal networks. Some organizations are also in the process of connecting their private networks to other organizations' private networks. Firewall security capabilities should also be used to provide protection for these types of connections.

This chapter identifies areas of security that should be considered with connections to external networks. Security policies must be developed for user identification and authorization, software import controls, encryption, and system architecture, which include the use of Internet firewall security capabilities. Chapter sections discuss security policy statements that address connections to external networks including the Internet. Each section contains multiple sample policies for use at the different risk profiles. Some areas provide multiple examples at the same risk level to show the different presentation methods that might be used to get the message across.

The first section discusses the risks and assumptions that should be acknowledged before a security analysis can be performed.

Risks and Assumptions

An understanding of the risks and assumptions is required before defining security policies for external connections. It is beyond the scope of this chapter to quantify the probability of the risks; however, the risks should cover a broad, comprehensive area. The following are the risks and assumptions:

- The data being protected, while not classified, is highly sensitive and would do damage to the organization and its mission if disclosed or captured.
- The integrity of the internal network directly affects the ability of the organization to accomplish its mission.
- The internal network is physically secure; the people using the internal network are trustworthy.
- PCs on the internal network are considered to be unsecured. Reliance is placed on the physical security of the location to protect them.

- Whenever possible, employees who are connected from remote sites should be treated as members of the internal network and have access to as many services as possible without compromising internal security.
- The Internet is assumed to be unsecured; the people using the Internet are assumed to be untrustworthy.
- Employees are targets for spying; information they carry or communicate is vulnerable to capture.
- Passwords transmitted over outside connections are vulnerable to capture.
- Any data transmitted over outside connections are vulnerable to capture.
- There is no control over e-mail once it leaves the internal network; e-mail can be read, tampered with, and spoofed.
- Any direct connection between a PC on the internal network and one on the outside can possibly be compromised and used for intrusion.
- Software bugs exist and may provide intrusion points from the outside into the internal network.
- Password protection on PCs directly reachable from the outside can be compromised and used for intrusion.
- Security through obscurity is counter-productive. Easy-to-understand measures are more likely to be sound, and are easier to administer.

Security Policies

Security policies fall into two broad categories: technical policies to be carried out by hardware or software, and administrative policy to be carried out by people using and managing the system. The final section of this chapter discusses Internet firewall security policies in more detail.

Identification and Authentication

Identification and authentication are the processes of recognizing and verifying valid users or processes. Identification and authentication information is generally then used to determine what system resources a user or process will be allowed to access. The determination of who can access what should coincide with a data categorization effort.

The assumption is that there is connectivity to internal systems from external networks or the Internet. If there is no connectivity, there is no need for identification and authentication controls. Many organizations separate Internet-accessible systems from internal systems through the use of firewalls and routers.

Authentication over the Internet presents several problems. It is relatively easy to capture identification and authentication data (or any data) and replay it in order to impersonate a user. As with other remote identification and authorization controls, and often with internal authorization systems, there can be a high level of user dissatisfaction and uncertainty, which can make this data obtainable via social engineering. Having additional authorization controls for use of the Internet may also contribute to authorization data proliferation, which is difficult for users to manage. Another problem is the ability to hijack a user session after identification and authorization have been performed.

There are three major types of authentication available: static, robust, and continuous. Static authentication includes passwords and other techniques that can be compromised through replay attacks. They are often called reusable passwords. Robust authentication involves the use of cryptography or other techniques to create one-time passwords that are used to create sessions. These can be compromised by session hijacking. Continuous authentication prevents session hijacking.

Static Authentication

Static authentication only provides protection against attacks in which an impostor cannot see, insert, or alter the information passed between the claimant and the verifier during an authentication exchange and subsequent session. In these cases, an impostor can only attempt to assume a claimant's identity by initiating an access control session as any valid user might do and trying to guess a legitimate user's authentication data. Traditional password schemes provide this level of protection, and the strength of the authentication process is highly dependent on the difficulty of guessing password values and how well they are protected.

Robust Authentication

This class of authentication mechanism relies on dynamic authentication data that changes with each authenticated session between a claimant and verifier. An impostor who can see information passed between the claimant and verifier may attempt to record this information, initiate a separate access control session with the verifier, and replay the recorded authentication data in an attempt to assume the claimant's identity. This type of authentication protects against such attacks, because authentication data recorded during a previous session will not be valid for any subsequent sessions.

However, robust authentication does not provide protection against active attacks in which the impostor is able to alter the content or flow of information between the claimant and verifier after they have established a legitimate session. Since the verifier binds the claimant's identity to the logical communications channel for the duration of the session, the verifier believes that the claimant is the source of all data received through this channel.

Traditional fixed passwords would fail to provide robust authentication because the password of a valid user could be viewed and used to assume that user's identity later. However, one-time passwords and digital signatures can provide this level of protection.

Continuous Authentication

This type of authentication provides protection against impostors who can see, alter, and insert information passed between the claimant and verifier even after the claimant/verifier authentication is complete. These are typically referred to as active attacks, since they assume that the impostor can actively influence the connection between claimant and verifier. One way to provide this form of authentication is to apply a digital signature algorithm to every bit of data that is sent from the claimant to the verifier. There are other combinations of cryptography that can provide this form of authentication, but current strategies rely on applying some type of cryptography to every bit of data sent. Otherwise, any unprotected bit would be suspect.

Applying Identification and Authorization Policies

Although passwords are easily compromised, an organization may find that a threat is not likely, would be fairly easy to recover from, or would not affect critical systems (which may have separate protection mechanisms). In low-risk connections, only static authentication may be required for access to corporate systems from external networks or the Internet.

In medium-risk connections, Internet access to information and processing (low impact if modified, unavailable, or disclosed) would require a password, and access to all other resources would require robust authentication. Telnet access to corporate resources from the Internet would also require the use of robust authentication.

Internet access to all systems behind the firewall would require robust authentication. Access to information and processing (high impact if modified, unavailable, or disclosed) would require continuous authentication.

Password Management Policies

The following are general password policies applicable for Internet use. These are considered to be the minimum standards for security control.

- Passwords and user log-on IDs will be unique to each authorized user.
- Passwords will consist of a minimum of 6 alphanumeric characters (no common names or phrases). There should be computer-controlled lists of proscribed password rules and periodic testing (e.g., letter and number sequences, character repetition, initials, common words, and standard names) to identify any password weaknesses.
- Passwords will be kept private i.e., not shared, coded into programs, or written down.
- Passwords will be changed every 90 days (or less). Most operating systems can enforce password change with an automatic expiration and prevent repeated or reused passwords.
- User accounts will be frozen after 3 failed log-on attempts. All erroneous password entries will be recorded in an audit log for later inspection and action, as necessary.
- Sessions will be suspended after 15 minutes (or other specified period) of inactivity and require the password to be reentered.

- Successful log-ons should display the date and time of the last log-on and log-off.
- Log-on IDs and passwords should be suspended after a specified period of non-use.
- For high-risk systems, after excessive violations, the system should generate an alarm and be able to simulate a continuing session (with dummy data, etc.) for the failed user (to keep this user connected while personnel attempt to investigate the incoming connection).

Robust Authentication Policy

The decision to use robust authentication requires an understanding of the risks, the security gained, and the cost of user acceptance and administration. User acceptance will be dramatically improved if users are appropriately trained in robust authentication and how it is used.

There are many technologies available that provide robust authentication including dynamic password generators, cryptography-based challenge/ response tokens and software, and digital signatures and certificates. If digital signatures and certificates are used, another policy area is opened up: the security requirements for the certificates.

Users of robust authentication must receive training prior to use of the authentication mechanism. Employees are responsible for safe handling and storage of all company authentication devices. Authentication tokens should not be stored with a computer that will be used to access corporate systems. If an authentication device is lost or stolen, the loss must be immediately reported to security so that the device can be disabled.

Digital Signatures and Certificates

If identification and authorization makes use of digital signatures, then certificates are required. They can be issued by the organization or by a trusted third party. Commercial public key infrastructures (PKI) are emerging within the Internet community. Users can obtain certificates with various levels of assurance. For example, level 1 certificates verify electronic mail addresses. This is done through the use of a personal information number that a user would supply when asked to register. This level of certificate may also provide a name as well as an electronic mail address; however, it may or may not be a genuine name (i.e., it could be an alias). Level 2 certificates verify a user's name, address, social security number, and other information against a credit bureau database. Level 3 certificates are available to companies. This level of certificate provides photo identification (e.g., for their employees) to accompany the other items of information provided by a Level 2 certificate.

Once obtained, digital certificate information may be loaded into an electronic mail application or a web browser application to be activated and provided whenever a web site or another user requests it for the purposes of verifying the identity of the person with whom they are communicating. Trusted certificate authorities are required to administer such systems with strict controls, otherwise fraudulent certificates could easily be issued.

Many of the latest web servers and web browsers incorporate the use of digital certificates. Secure Socket Layer (SSL) is the technology used in most Web-based applications. SSL version 2.0 supports strong authentication of the Web server, while SSL 3.0 adds client-side authentication. Once both sides are authenticated, the session is encrypted, providing protection against both eavesdropping and session hijacking. The digital certificates used are based on the X.509 standard and describe who issued the certificate, the validity period, and other information.

Oddly enough, passwords still play an important role even when using digital certificates. Since digital certificates are stored on a computer, they can only be used to authenticate the computer, rather than the user, unless the user provides some other form of authentication to the computer. Passwords or "passphrases" are generally used; smart cards and other hardware tokens will be used in the future.

Any company's systems making limited distribution data available over the Internet should use digital certificates to validate the identity of both the user and the server. Only Company-approved certificate authorities should issue certificates. Certificates at the user end should be used in conjunction with standard technologies such as Secure Sockets Layer to provide continuous authentication to eliminate the risk of session hijacking. Access to digital certificates stored on personal computers should be protected by passwords or passphrases. All policies for password management must be followed and enforced.

Software Import Control

Data on computers is rarely static. Mail arrives and is read. New applications are loaded from floppy, CD-ROM, or across a network. Web-based interactive software downloads executables that run on a computer. Each modification runs the risk of introducing viruses, damaging the configuration of the computer, or violating software-licensing agreements. Organizations need to protect themselves with different levels of control depending on the vulnerability to these risks. Software Import Control provides an organization with several different security challenges:

- Virus and Trojan horse prevention, detection, and removal
- Controlling Interactive Software (Java, ActiveX)
- Software licensing

Each challenge can be categorized according to the following criteria:

- Control: who initiates the activity, and how easily can it be determined that software has been imported
- Threat type: executable program, macro, applet, violation of licensing agreement
- Cleansing action: scanning, refusal of service, control of permissions, auditing, deletion

When importing software onto a computer, one runs the risk of getting additional or different functionality than one bargained for. The importation may occur as a direct action, or as a hidden side effect, which is not readily visible. Examples of direct action include:

- File transfer — utilizing FTP to transfer a file to a computer
- Reading e-mail — causing a message which has been transferred to a computer to be read, or using a tool (e.g., Microsoft Word) to read an attachment
- Downloading software from a floppy disk or over the network can spawn indirect action. Some examples include (1) reading a Web page which downloads a Java applet to your computer and (2) executing an application such as Microsoft Word and opening a file infected with a Word Macro Virus.

Virus Prevention, Detection, and Removal

A virus is a self-replicating program spread from executables, boot records, and macros. Executable viruses modify a program to do something other than the original intent. After replicating itself into other programs, the virus may do little more than print an annoying message, or it could do something as damaging as deleting all of the data on a disk. There are different levels of sophistication in how hard a virus may be to detect.

The most common "carrier" of viruses has been the floppy disk, since "sneaker net" was the most common means of transferring software between computers. As telephone-based bulletin boards became popular, viruses travelled more frequently via modem. The Internet provides yet another channel for virus infections, one that can often bypass traditional virus controls.

For organizations that allow downloading of software over the Internet (which can be via Internet e-mail attachments) virus scanning at the firewall can be an appropriate choice — but it does not eliminate the need for client and server based virus scanning, as well. For several years to come, viruses imported on floppy disks or infected vendor media will continue to be a major threat.

Simple viruses can be easily recognized by scanning for a signature of byte strings near the entry point of a program, once the virus has been identified. Polymorphic viruses modify themselves as they propagate. Therefore, they have no signature and can only be found (safely) by executing the program in a virtual processor environment. Boot record viruses modify the boot record such that the virus is executed when the system is booted.

Applications that support macros are at risk for macro viruses. Macro viruses are commands that are embedded in data. Vendor applications, such as Microsoft Word, Microsoft Excel, or printing standards such as Postscript are common targets. When the application opens the data file the infected macro virus is instantiated.

The security service policy for viruses has three aspects:

- Prevention — policies which prevent the introduction of viruses into a computing environment,
- Detection — determination that an executable, boot record, or data file is contaminated with a virus, and

- Removal — deletion of the virus from the infected computing system may require reinstallation of the operating system from the ground up, deleting files, or deleting the virus from an infected file.

There are various factors that are important in determining the level of security concern for virus infection of a computer. Viruses are most prevalent on DOS, Windows (3.x, 95), and NT operating systems. However some UNIX viruses have been identified.

The frequency that new applications or files are loaded on to the computer is proportional to the susceptibility of that computer to viruses. Configuration changes resulting from exposure to the Internet, exposure to mail, or receipt of files from external sources are more at risk for contamination.

The greater the value of the computer or data on the computer, the greater the concern should be for ensuring that virus policy as well as implementation procedures are in place. The cost of removal of the virus from the computing environment must be considered within your organization as well as from customers you may have infected. Cost may not always be identified as monetary; company reputation and other considerations are just as important.

It is important to note that viruses are normally introduced into a system by a voluntary act of a user (e.g., installation of an application, executing a file, etc.). Prevention policies can therefore focus on limiting the introduction of potentially infected software and files to a system. In a high-risk environment, virus-scanning efforts should be focused on when new software or files are introduced to maximize protection.

Controlling Interactive Software

A programming environment evolving as a result of Internet technology is Interactive Software, as exemplified by Java and ActiveX. In an Interactive Software environment, a user accesses a server across a network. The server downloads an application (applet) onto the user's computer that is then executed. There have been various claims that when utilizing languages such as Java, it is impossible to introduce a virus because of restrictions within the scripting language for file system access and process control. However, security risks using Java and ActiveX have been documented.

Therefore, there are several assumptions of trust that a user must make before employing this technology:

- The server can be trusted to download trustworthy applets.
- The applet will execute in a limited environment restricting disk reads and writes to functions that do not have security.
- The applet can be scanned to determine if it is safe.
- Scripts are interpreted, not precompiled.

Firewall Policy

Firewalls are critical to the success of secured connections to external networks as well as the Internet. The main function of a firewall is to centralize access control. If outsiders or remote users can access the internal networks without going through the firewall, its effectiveness is diluted. For example, if a traveling manager has a modem connected to his office PC that he or she can dial into while traveling, and that PC is also on the protected internal network, an attacker who can dial into that PC has circumvented the controls imposed by the firewall. If a user has a dial-up Internet account with a commercial Internet Service Provider (ISP), and sometimes connects to the Internet from his office PC via modem, he is opening an unsecured connection to the Internet that circumvents the firewall.

Firewalls can also be used to secure segments of an organization's intranet, but this document will concentrate on the Internet aspects of firewall policy.

Firewalls provide several types of protection, to include:

- They can block unwanted traffic.
- They can direct incoming traffic to more trustworthy internal systems.
- They hide vulnerable systems, which can't easily be secured from the Internet.
- They can log traffic to and from the private network.
- They can hide information like system names, network topology, network device types, and internal user IDs from the Internet.
- They can provide more robust authentication than standard applications might be able to do.

Each of these functions is described in more detail below.

As with any safeguard, there are trade-offs between convenience and security. Transparency is the visibility of the firewall to both inside users and outsiders going through a firewall. A firewall is transparent to users if they do not notice or stop at the firewall in order to access a network. Firewalls are typically configured to be transparent to internal network users (while going outside the firewall); on the other hand, firewalls are configured to be non-transparent for outside network coming through the firewall. This generally provides the highest level of security without placing an undue burden on internal users.

Firewall Authentication

Router-based firewalls don't provide user authentication. Host-based firewalls can provide various kinds of authentication. *Username/password authentication* is the least secure, because the information can be sniffed or shoulder-surfed. *One-time passwords* use software or hardware tokens and generate a new password for each session. This means that old passwords cannot be reused if they are sniffed or otherwise borrowed or stolen. Finally, *Digital Certificates* use a certificate generated using public key encryption.

Routing Versus Forwarding

A clearly defined policy should be written as to whether or not the firewall will act as a router or a forwarder of Internet packets. This is trivial in the case of a router that acts as a packet filtering gateway because the firewall (router in this case) has no option but to route packets. Applications gateway firewalls should generally not be configured to route any traffic between the external interface and the internal network interface, since this could bypass security controls. All external to internal connections should go through the application proxies.

Source Routing

Source routing is a routing mechanism whereby the path to a target machine is determined by the source, rather than by intermediate routers. Source routing is mostly used for debugging network problems but could also be used to attack a host. If an attacker has knowledge of some trust relationship between your hosts, source routing can be used to make it appear that the malicious packets are coming from a trusted host. Because of this security threat, a packet filtering router can easily be configured to reject packets containing source route option.

IP Spoofing

IP spoofing is when an attacker masquerades his machine as a host on the target's network (i.e., fooling a target machine that packets are coming from a trusted machine on the target's internal network). Policies regarding packet routing need to be clearly written so that they will be handled accordingly if there is a security problem. It is necessary that authentication based on source address be combined with other security schemes to protect against IP spoofing attacks.

Types of Firewalls

There are different implementations of firewalls, which can be arranged in different ways. These include packet filtering gateways, application gateways, and hybrid or complex gateways.

Packet Filtering Gateways

Packet filtering firewalls use routers with packet filtering rules to grant or deny access based on source address, destination address, and port. They offer minimum security but at a very low cost, and can be an appropriate choice for a low-risk environment. They are fast, flexible, and transparent. Filtering rules are not often easily maintained on a router, but there are tools available to simplify the tasks of creating and maintaining the rules.

Filtering gateways do have inherent risks, including:

- The source and destination addresses and ports contained in the IP packet header are the only information that is available to the router in making a decision whether or not to permit traffic access to an internal network.

- They don't protect against IP or DNS address spoofing.
- An attacker will have a direct access to any host on the internal network once access has been granted by the firewall.
- Strong user authentication isn't supported with packet filtering gateways.
- They provide little or no useful logging.

Application Gateways

An application gateway uses server programs called proxies that run on the firewall. These proxies take external requests, examine them, and forward legitimate requests to the internal host that provides the appropriate service. Application gateways can support functions such as user authentication and logging.

Because an application gateway is considered the most secure type of firewall, this configuration provides a number of advantages to the medium-high risk site:

- The firewall can be configured as the only host address that is visible to the outside network, requiring all connections to and from the internal network to go through the firewall.
- The use of proxies for different services prevents direct access to services on the internal network, protecting the enterprise against insecure or misconfigured internal hosts.
- Strong user authentication can be enforced with application gateways.
- Proxies can provide detailed logging at the application level. Application level firewalls shall be configured such that outbound network traffic appears as if the traffic had originated from the firewall (i.e., only the firewall is visible to outside networks). In this manner, direct access to network services on the internal network is not allowed. All incoming requests for different network services such as Telnet, FTP, HTTP, RLOGIN, etc., regardless of which host on the internal network will be the final destination, must go through the appropriate proxy on the firewall.

Applications gateways require a proxy for each service, such as FTP, HTTP, etc., to be supported through the firewall. When a service is required that is not supported by a proxy, an organization has three choices.

- Deny the service until the firewall vendor has developed a secure proxy. This is the preferred approach, as many newly introduced Internet services have unacceptable vulnerabilities.
- Develop a custom proxy — This is a fairly difficult task and should be undertaken only by very sophisticated technical organizations.
- Pass the service through the firewall — Using what are typically called "plugs," most application gateway firewalls allow services to be passed directly through the firewall with only a minimum of packet filtering. This can limit some of the vulnerability but can result in compromising the security of systems behind the firewall.

Hybrid or Complex Gateways

Hybrid gateways combine two or more of the above firewall types and implement them in series rather than in parallel. If they are connected in series, then the overall security is enhanced; on the other hand, if they are connected in parallel, then the network security perimeter will be only as secure as the least secure of all methods used. In medium- to high-risk environments, a hybrid gateway may be the ideal firewall implementation.

Suggested ratings are identified in Exhibit 19.1 for various firewall types.

EXHIBIT 19.1 Firewall Security Risk

Firewall Architecture	High-Risk Environment (e.g., hospital)	Medium-Risk Environment (e.g., university)	Low-Risk Environment (e.g., florist shop)
Packet filtering	Unacceptable	Minimal security	Recommended
Application gateways	Effective option	Recommended	Acceptable
Hybrid gateways	Recommended	Effective option	Acceptable

Firewall Architectures

Firewalls can be configured in a number of different architectures, providing various levels of security at different costs of installation and operation. Organizations should match their risk profile to the type of firewall architecture selected. The following describes typical firewall architectures and sample policy statements.

Multi-homed host

A multi-homed host is a host (a firewall in this case) that has more than one network interface, with each interface connected to logically and physically separate network segments. A dual-homed host (host with two interfaces) is the most common instance of a multi-homed host.

A dual-homed firewall is a firewall with two network interface cards (NICs) with each interface connected to different networks. For instance, one network interface is typically connected to the external or untrusted network, while the other interface is connected to the internal or trusted network. In this configuration, a key security tenet is not to allow traffic coming in from the untrusted network to be directly routed to the trusted network, that is, the firewall must always act as an intermediary. Routing by the firewall shall be disabled for a dual-homed firewall so that IP packets from one network are not directly routed from one network to the other.

Screened Host

A screened host firewall architecture uses a host (called a bastion host) to which all outside hosts connect, rather than allow direct connection to other, less secure internal hosts. To achieve this, a filtering router is configured so that all connections to the internal network from the outside network are directed towards the bastion host. If a packet filtering gateway is to be deployed, then a bastion host should be set up so that all connections from the outside network go through the bastion host to prevent direct Internet connection between the internal network and the outside world.

Screened Subnet

The screened subnet architecture is essentially the same as the screened host architecture, but adds an extra stratum of security by creating a network at which the bastion host resides (often call perimeter network) which is separated from the internal network. A screened subnet is deployed by adding a perimeter network in order to separate the internal network from the external. This assures that if there is a successful attack on the bastion host, the attacker is restricted to the perimeter network by the screening router that is connected between the internal and perimeter network.

Intranet

Although firewalls are usually placed between a network and the outside untrusted network, in large companies or organizations, firewalls are often used to create different subnets of the network, often called an intranet. Intranet firewalls are intended to isolate a particular subnet from the overall corporate network. The reason for the isolation of a network segment might be that certain employees can access subnets guarded by these firewalls only on a need-to-know basis. An example could be a firewall for the payroll or accounting department of an organization.

The decision to use an intranet firewall is generally based on the need to make certain information available to some but not all internal users, or to provide a high degree of accountability for the access and use of confidential or sensitive information.

For any systems hosting internal critical applications, or providing access to sensitive or confidential information, internal firewalls or filtering routers should be used to provide strong access control and support for auditing and logging. These controls should be used to segment the internal network to support the access policies developed by the designated owners of information.

Firewall Administration

A firewall, like any other network device, has to be managed by someone. Security policy should state who is responsible for managing the firewall.

Two firewall administrators (one primary and one secondary) shall be designated by the Chief Information Security Officer (or other manager) and shall be responsible for the upkeep of the firewall. The primary

administrator shall make changes to the firewall, and the secondary shall only do so in the absence of the former so that there is no simultaneous or contradictory access to the firewall. Each firewall administrator shall provide their home phone number, pager number, cellular phone number, and other numbers or codes in which they can be contacted when support is required.

Qualification of the Firewall Administrator

Two experienced people are generally recommended for the day-to-day administration of the firewall. In this manner availability of the firewall administrative function is largely ensured. It should be required that on-call information about each firewall administrator be written down so that one may be contacted in the event of a problem.

Security of a site is crucial to the day-to-day business activity of an organization. It is therefore required that the administrator of the firewall have a sound understanding of network concepts and implementation. For instance, since most firewalls are TCP/IP based, a thorough understanding of this protocol is compulsory. An individual that is assigned the task of firewall administration must have good hands-on experience with networking concepts, design, and implementation so that the firewall is configured correctly and administered properly. Firewall administrators should receive periodic training on the firewalls in use and in network security principles and practices.

Remote Firewall Administration

Firewalls are the first line of defense visible to an attacker. By design, firewalls are generally difficult to attack directly, causing attackers to often target the administrative accounts on a firewall. The username/password of administrative accounts must be strongly protected.

The most secure method of protecting against this form of attack is to have strong physical security around the firewall host and to only allow firewall administration from an attached terminal. However, operational concerns often dictate that some form of remote access for firewall administration be supported. In no case should remote access to the firewall be supported over untrusted networks without some form of strong authentication. In addition, to prevent eavesdropping, session encryption should be used for remote firewall connections.

User Accounts

Firewalls should never be used as general purpose servers. The only user accounts on the firewall should be those of the firewall administrator and any backup administrators. In addition, only these administrators should have privileges for updating system executables or other system software. Only the firewall administrator and backup administrators will be given user accounts on the COMPANY firewall. Any modification of the firewall system software must be done by the firewall administrator or backup administrator and requires approval of the cognizant Manager.

Firewall Backup

To support recovery after failure or natural disaster, a firewall, like any other network host, has to have some policy defining system backup. Data files as well as system configuration files need to be components of a backup and recovery plan in case of firewall failure.

The firewall (system software, configuration data, database files, etc.) must be backed up daily, weekly, and monthly so that in case of system failure, data and configuration files can be recovered. Backup files should be stored securely on read-only media so that data in storage is not over-written inadvertently, and locked up so that the media is only accessible to the appropriate personnel.

Another backup alternative would be to have another firewall configured as one already deployed and kept safely in case there is a failure of the current one. This backup firewall would simply be turned on and used as the firewall while the previous one is undergoing a repair. At least one firewall should be configured and reserved (not-in-use) so that in case of a firewall failure, this backup firewall can be switched in to protect the network.

Other Firewall Policy Considerations

Firewall technology has only been around for the last five years. In the past two years, however, firewall products have diversified considerably and now offer a variety of technical security controls that can be used in ever more complex network connections.

This section discusses some of the firewall policy considerations in the areas of network trust relationships, virtual private networks, DNS and mail resolution, system integrity, documentation, physical firewall security, firewall incident handling, service restoration, upgrades, and audit trail logging.

Network Trust Relationships

Business networks frequently require connections to other business networks. Such connections can occur over leased lines, proprietary Wide area networks, value added networks (VANs), or public networks such as the Internet. For instance, many local governments use leased lines or dedicated circuits to connect regional offices across the state. Many businesses use commercial VANs to connect business units across the country or the world.

The various network segments involved may be under control of different organizations and may operate under a variety of security policies. By their very nature, when networks are connected the security of the resulting overall network drops to the level of the weakest network. When decisions are made for connecting networks, trust relationships must be defined to avoid reducing the effective security of all networks involved.

Trusted networks are defined as networks that share the same security policy or implement security controls and procedures that provide an agreed upon set of common security services. Untrusted networks are those that do not implement such a common set of security controls, or where the level of security is unknown or unpredictable. The most secure policy is to only allow connection to trusted networks, as defined by an appropriate level of management. However, business needs may force temporary connections with business partners or remote sites that involve the use of untrusted networks.

Virtual Private Networks (VPN)

Virtual private networks allow a trusted network to communicate with another trusted network over untrusted networks such as the Internet. Because some firewalls provide VPN capability, it is necessary to define policy for establishing VPNs. The following are recommended policy statements:

- Any connection between firewalls over public networks shall use encrypted virtual private networks to ensure the privacy and integrity of the data passing over the public network.
- All VPN connections must be approved and managed by the Network Services Manager.
- Appropriate means for distributing and maintaining encryption keys must be established prior to operational use of VPNs.

DNS and Mail Resolution

On the Internet, the Domain Name Service provides the mapping and translation of domain names to IP addresses, such as "mapping server1. acme.com to 123.45.67.8". Some firewalls can be configured to run as a primary, secondary, or caching DNS server.

Deciding how to manage DNS services is generally not a security decision. Many organizations use a third party, such as an Internet Service Provider, to manage their DNS. In this case, the firewall can be used as a DNS caching server, improving performance but not requiring your organization to maintain its own DNS database.

If the organization decides to manage its own DNS database, the firewall can (but doesn't have to) act as the DNS server. If the firewall is to be configured as a DNS server (primary, secondary, or caching), it is necessary that other security precautions be in place. One advantage of implementing the firewall as a DNS server is that it can be configured to hide the internal host information of a site. In other words, with the firewall acting as a DNS server, internal hosts get an unrestricted view of both internal and external DNS data. External hosts, on the other hand, do not have access to information about internal host machines. To the outside world all connections to any host in the internal network will appear to have originated from the firewall. With the host information hidden from the outside, an attacker will not know the host names and addresses of internal hosts that offer service to the Internet. A security policy for DNS hiding might state: If the firewall is to run as a DNS server, then the firewall must be configured to hide information about the network so that internal host data is not advertised to the outside world.

System Integrity

To prevent unauthorized modifications of the firewall configuration, some form of integrity assurance process should be used. Typically, checksums, cyclic redundancy checks, or cryptographic hashes are made from the run-time image and saved on protected media. Each time the firewall configuration has been modified by an authorized individual (usually the firewall administrator), it is necessary that the system integrity online database be updated and saved onto a file system on the network or removable media. If the system integrity check shows that the firewall configuration files have been modified, it will be known that the system has been compromised.

The firewall's system integrity database shall be updated each time the firewall's configuration is modified. System integrity files must be stored on read only media or off-line storage. System integrity shall be checked on a regular basis on the firewall in order for the administrator to generate a listing of all files that may have been modified, replaced, or deleted.

Documentation

It is important that the operational procedures for a firewall and its configurable parameters be well documented, updated, and kept in a safe and secure place. This assures that if a firewall administrator resigns or is otherwise unavailable, an experienced individual can read the documentation and rapidly pick up the administration of the firewall. In the event of a break-in such documentation also supports trying to recreate the events that caused the security incident.

Physical Firewall Security

Physical access to the firewall must be tightly controlled to preclude any authorized changes to the firewall configuration or operational status, and to eliminate any potential for monitoring firewall activity. In addition, precautions should be taken to assure that proper environment alarms and backup systems are available to assure the firewall remains online.

The firewall should be located in a controlled environment, with access limited to the Network Services Manager, the firewall administrator, and the backup firewall administrator. The room in which the firewall is to be physically located must be equipped with heat, air-conditioner, and smoke alarms to assure the proper working order of the room. The placement and recharge status of the fire extinguishers shall be checked on a regular basis. If uninterruptible power service is available to any Internet-connected systems, such service should be provided to the firewall as well.

Firewall Incident Handling

Incident reporting is the process whereby certain anomalies are reported or logged on the firewall. A policy is required to determine what type of report to log and what to do with the generated log report. This should be consistent with Incident Handling policies detailed previously. The following policies are appropriate to all risk environments.

- The firewall shall be configured to log all reports on daily, weekly, and monthly bases so that the network activity can be analyzed when needed.
- Firewall logs should be examined on a weekly basis to determine if attacks have been detected.
- The firewall administrator shall be notified at anytime of any security alarm by e-mail, pager, or other means so that he may immediately respond to such alarm.
- The firewall shall reject any kind of probing or scanning tool that is directed to it so that information being protected is not leaked out by the firewall. In a similar fashion, the firewall shall block all software types that are known to present security threats to a network (such as ActiveX and Java) to better tighten the security of the network.

Restoration of Services

Once an incident has been detected, the firewall may need to be brought down and reconfigured. If it is necessary to bring down the firewall, Internet service should be disabled or a secondary firewall should be

made operational. Internal systems should not be connected to the Internet without a firewall. After being reconfigured, the firewall must be brought back into an operational and reliable state. Policies for restoring the firewall to a working state when a break-in occurs are needed.

In case of a firewall break-in, the firewall administrator(s) are responsible for reconfiguring the firewall to address any vulnerabilities that were exploited. The firewall shall be restored to the state it was before the break-in so that the network is not left wide open. While the restoration is going on, the backup firewall shall be deployed.

Upgrading the Firewall

It is often necessary that the firewall software and hardware components be upgraded with the necessary modules to assure optimal firewall performance. The firewall administrator should be aware of any hardware and software bugs, as well as firewall software upgrades that may be issued by the vendor. If an upgrade of any sort is necessary, certain precautions must be taken to continue to maintain a high level of operational security. Sample policies that should be written for upgrades may include the following:

- To optimize the performance of the firewall, all vendor recommendations for processor and memory capacities shall be followed.
- The firewall administrator must evaluate each new release of the firewall software to determine if an upgrade is required. All security patches recommended by the firewall vendor should be implemented in a timely manner.
- Hardware and software components shall be obtained from a list of vendor-recommended sources. Any firewall specific upgrades shall be obtained from the vendor. NFS shall not be used as a means of obtaining software components. The use of virus checked CD-ROM or FTP to a vendor's site is an appropriate method.
- The firewall administrator(s) shall monitor the vendor's firewall mailing list or maintain some other form of contact with the vendor to be aware of all required upgrades. Before an upgrade of any of the firewall components, the firewall administrator must verify with the vendor that an upgrade is required. After any upgrade the firewall shall be tested to verify proper operation prior to going operational.

Given the rapid introduction of new technologies and the tendency for organizations to continually introduce new services, firewall security policies should be reviewed on a regular basis. As network requirements change, so should security policy.

Logs and Audit Trails (Audit/Event Reporting and Summaries)

Most firewalls provide a wide range of capabilities for logging traffic and network events. Some security-relevant events that should be recorded on the firewall's audit trail logs are: hardware and disk media errors, login/logout activity, connect time, use of system administrator privileges, inbound and outbound e-mail traffic, TCP network connect attempts, inbound and outbound proxy traffic type.

Summary

Connections to external networks and to the Internet are rapidly becoming commonplace in today's business community. These connections must be effectively secured to protect internal trusted networks from misuse and attack. The security policies outlined above should provide an effective guideline for implementing the appropriate level of controls to protect internal networks from outside attack.

An Introduction to LAN/WAN Security

Steven F. Blanding

The purpose of this chapter is to provide a basic understanding of how to protect Local Area Networks (LANs) and Wide Area Networks (WANs). Connecting computers to networks significantly increases risk. Networks connect large numbers of users to share information and resources, but network security depends heavily on the cooperation of each user. Security is as strong as the weakest link. Studies have shown that most of the abuses and frauds are carried out by authorized users, not outsiders. As the number of LANs and WANs increase, cost-effective security becomes a much more significant issue to deter fraud, waste, and abuse and to avoid embarrassment.

This chapter is intended to help LAN managers understand why they should be concerned about security, what their security concerns should be, and how to resolve their concerns. We will begin by introducing the concept of risk management and touch on basic requirements for protecting LANs. This will be followed by a summary of LAN components and features that will serve as a foundation for determining security requirements. LAN security requirements will then be discussed in terms of the risk assessment process, followed by a detailed discussion of how to implement LAN security in a step-by-step approach. This should provide the necessary guidance in applying security procedures to specific LAN/WAN security risks and exposures.

DEFINITIONS

A LAN, or local area network, is a network of personal computers deployed in a small geographic area such as an office complex, building, or campus. A WAN, or wide area network, is an arrangement of data transmission facilities that provides communications capability across a broad geographic area. LANs and WANs can potentially contain and process sensitive data and, as a result, a plan should be prepared for the security and privacy of these networks. This plan should involve mandatory

periodic training in computer security awareness and accepted security practices for all individuals who are involved in the management, use, and operation of these networks and systems. Organizations should have a security program to assure that each automated system has a level of security that is commensurate with the risk and magnitude of the harm that could result from the loss, misuse, disclosure, or modification of the information contained in the system. Each system's level of security must protect the confidentiality, integrity, and availability of the information. Specifically, this would require that the organization has appropriate technical, personnel, administrative, environmental, and telecommunications safeguards; a cost-effective security approach; and adequate resources to support critical functions and provide continuity of operation in the event of a disaster.

Risk management is defined as a process for minimizing losses through the periodic assessment of potential hazards and the systematic application of corrective measures. Risk to information systems is generally expressed in terms of the potential for loss. The greater the value of the assets, the greater the potential loss. Threats can be people such as hackers, disgruntled employees, error-prone programmers, careless data entry operators, things such as unreliable hardware, or even nature itself such as earthquakes, floods, and lightning. Vulnerabilities are flaws in the protection of assets that can be exploited, partially or fully, by threats resulting in loss. Safeguards preclude or mitigate vulnerabilities.

Managing risks involves not only identifying threats but also determining their impact and severity. Some threats require extensive controls while others require few. Certain threats, such as viruses and other computer crimes, have been highlighted through extensive press coverage, while other threats such as repeated errors by employees generally receive no publicity. Yet, statistics reveal that errors and omissions generally cause more harm than virus attacks. Resources are often expended on threats not worth controlling, while other major threats receive little or no control. Until managers understand the magnitude of the problem and the areas in which threats are most likely to occur, protecting vital computer resources will continue to be an arbitrary and ineffective proposition. The added complexity of LAN/WAN environments creates greater challenges for understanding and managing risks.

LAN/WAN ENVIRONMENT

A brief overview of the highly complex LAN/WAN environment serves as a foundation for the understanding of network security issues and solutions. Many environments use a mix of personal computers (PCs), LANs/WANs, terminals, minicomputers, and mainframes to meet processing needs. LANs are primarily networks that come in many varieties and

provide connectivity, directly or indirectly, to many mini and mainframe computers.

A LAN is a group of computers and other devices dispersed over a relatively limited area and connected by a communications link that enables any device to interact with any other on the network. LANs commonly include PCs and shared resources such as laser printers and large hard disks. Although single LANs are typically limited geographically to a department or office building, separate LANs can be connected to form larger networks. Alternatively, LANs can be configured utilizing a client-server architecture which makes use of distributed intelligence by splitting the processing of an application between two distinct components: a front-end client and a back-end server. The client component, itself a complete, stand-alone PC, offers the user its full range of power and features for running applications. The server component, which can be another personal computer, minicomputer, or mainframe, enhances the client by providing the traditional strengths offered by minicomputers and mainframes in a time-shared environment. These strengths are data management, information sharing among clients, and sophisticated network administration and security features.

LAN/WAN Components

PCs are an integral part of the LAN, using an adaptor board, cabling, and software to access the data and devices on the network. PCs can also have dial-in access to a LAN via a modem and telephone line. The PC is the most vulnerable component of a LAN since a PC typically has weak security features, such as lack of memory protection.

LAN cabling, using twisted-pair cable, thin coaxial cable, standard coaxial cable, or optical fiber provides the physical connections. Of these, fiber optics provides the most security, as well as the highest capacity. Cabling is susceptible to tapping to gain unauthorized access to data, but this is considered unlikely due to the high cost of such action. A new alternative to cabling is a wireless LAN, which uses infrared light waves or various radio frequencies (RF) for transmission. Wireless LANs, like cellular telephones, are vulnerable to unauthorized interception.

Servers are dedicated computers that provide various support and resources to client workstations, including file storage, applications, data bases, and security services. In small peer-to-peer LANs, the server can function as one of the client PCs. In addition, minicomputers and mainframes can function in a true server mode. This shared processing feature is not to be confused with PCs that serve as dumb terminals to access minis and mainframes. Controlling physical access to the server is a basic LAN security issue.

A network operating system is installed on a LAN server to coordinate the activities of providing services to the computers and other devices attached to the network. Unlike a single-user operating system, which performs the basic tasks required to keep one computer running, a network operating system must acknowledge and respond to requests from many workstations, managing such details as network access and communications, resource allocation and sharing, data protection, and error control. The network operating system provides crucial security features for a LAN, and is discussed more fully in a separate section below.

Input/output devices (e.g., printers, scanners, faxes, etc.) are shared resources available to LAN users and are susceptible to security problems, such as sensitive output left unattended on a remote printer.

A backbone LAN interconnects the small LAN work groups. This can be accomplished through the use of copper or fiber-optic cabling for the backbone circuits. Fiber optics provides a high degree of security because light signals are difficult to tap or otherwise intercept. Internetworking devices include repeaters, bridges, routers, and gateways. These are communications devices for LANs/WANs that provide the connections, control, and management for efficient and reliable Internetwork access. These devices can also have built-in security control features for controlling access.

Dial-In Access

A PC dial-in connection can be made directly to a LAN server. This connection can occur when a server has been fitted with a dial-in port capability. The remote PC requires communications software, a modem, a telephone line, and the LAN dial-in number to complete the connection. This access procedure invokes the LAN access control measures such as log-on/password requirements. LANs usually have specific controls for remote dial-in procedures. The remote unit used to dial-in may be any computer, including a laptop PC.

A PC can remotely control a second PC via modems and commercially purchased software products such as *PC Anywhere* and *Carbon Copy*. When this second PC is cabled to a LAN, a remote connection can be made from the first PC through the second PC into the LAN. The result is access to the LAN within the limits of the user's access controls. One example of this remote control access is when an individual uses a home computer to dial in to their office PC and remotely control the office PC to access the LAN. The office PC is left running to facilitate this connection. It should be noted that the LAN may not have the capability to detect that a remote-control session is taking place.

Dial-in capabilities dramatically increase the risk of unauthorized access to the system, thereby requiring strong password protection and other safeguards, such as call-back devices, which are discussed later.

Topology

The topology of a network is the way in which the PCs on the network are physically interconnected. Network devices can be connected in specific patterns such as a bus, ring, or star or some combination of these. The name of the topology describes its physical layout.

PCs on a bus network send data to a head-end retransmitter that rebroadcasts the data back to the PCs. In a ring network, messages circulate the loop, passing from PC to PC in bucket-brigade fashion. An example is IBM's Token-Ring network, which uses a special data packet called a "token." Only one token exists on the network at any one time, and the station owning the token is granted the right to communicate with other stations on the network. A predefined token-holding time keeps one user from monopolizing the token indefinitely. When the token owner's work is completed or the token-holding time has run out, the token is passed to the next user on the ring.

In a star configuration, PCs communicate through a central hub device. Regarded as the first form of local area networking, the star network requires each node to have a direct line to the central or shared hub resource.

LAN topology has security implications. For example, in sending a data from one user to another, the star topology sends it directly through the hub to the receiver. In the ring and bus topologies, the message is routed past other users. As a result, sensitive data messages can be intercepted by these other uses in these types of topologies.

Protocols

A protocol is a formal set of rules that computers use to control the flow of messages between them. Networking involves such a complex variety of protocols that the International Standards Organization (ISO) defined the now-popular seven-layer communications model. The Open Systems Interconnection (OSI) model describes communication processes as a hierarchy of layers, each dependent on the layer beneath it. Each layer has a defined interface with the layer above and below. This interface is made flexible so that designers can implement various communications protocols with security features and still follow the standard. Below is a very brief summary of the layers, as depicted in the OSI model.

- The *application* layer is the highest level. It interfaces with users, gets information from data bases, and transfers whole files. E-mail is an application at this level.
- The *presentation* layer defines how applications can enter the network.
- The *session* layer makes the initial contact with other computers and sets up the lines of communication. This layer allows devices to be referenced by name rather than by network address.
- The *transport* layer defines how to address the physical locations/devices on the network, make connections between nodes, and handles the Internetworking of messages.
- The *network* layer defines how the small packets of data are routed and relayed between end systems on the same network or on interconnected networks.
- The *data-link* layer defines the protocol that computers must follow to access the network for transmitting and receiving messages. Token Ring and Ethernet operate within this layer and the physical layer, defined below.
- The *physical* layer defines the physical connection between the computer and the network and, for example, converts the bits into voltages or light impulses for transmission. Topology is defined here.

Bridges, routers, and gateways are "black boxes" that permit the use of different topologies and protocols within a single heterogeneous system. In general, two LANs that have the same physical layer protocol can be connected with a simple, low-cost repeater. Two LANs that speak the same data-link layer protocol can be connected with a bridge even if they differ at the physical layer. If the LANs have a common network layer protocol, they can be connected with a router. If two LANs have nothing in common they can be connected at the highest level, the application layer, with a gateway.

These black boxes have features and filters that can enhance network security under certain conditions, but the features must be understood and utilized. For example, an organization could elect to permit E-mail to pass bidirectionally by putting in place a mail gateway while preventing interactive log-in sessions and file sessions by not passing any other traffic than E-mail.

Companies should specify a set of OSI protocols for the computer network intended for acquisition and use by their organizations. This requirement should preclude the acquisition of their favorite computer networking products. Instead, when acquiring computer networking products, they are required to purchase OSI capabilities in addition to any other requirements so that multivendor interoperability becomes a built-in feature of the computing environment.

Security is of fundamental importance to the acceptance and use of open systems in a LAN/WAN environment. Part 2 of the Opens Systems Interconnection reference model (Security Architecture) is now an international standard. The standard describes a general architecture for security in OSI, defines a set of security services that may be supported within the OSI model, and outlines a number of mechanisms that can be used in providing the services. However, no protocols, formats, or minimum requirements are contained in the standard.

An organization desiring security in a product that is being purchased in accordance with this profile must specify the security services required, the placement of the services within the OSI architecture, the mechanisms to provide the services, and the management features required. Security services may be provided at one or more of the layers. The primary security services that are defined in the OSI security architecture are (1) data confidentially services to protect against unauthorized disclosure; (2) data integrity services to protect against unauthorized modification, insertion, and deletion; (3) authentication services to verify the identity of communication peer entities and the source of data; and (4) access control services to allow only authorized communication and system access.

Applications

Applications on a LAN can range from word processing to data base management systems. The most universally used application is E-mail. E-mail software provides a user interface to help construct the mail message and an engine to move the E-mail to its destination. Depending on the address, the E-mail may be routed across the office via the LAN or across the country via LAN/WAN bridges and gateways. E-mail may also be sent to other mail systems, both mainframe- and PC-based. An important security note is that on some systems it is also possible to restrict mail users from attaching files as a part of an antivirus program.

Many application systems have their own set of security features, in addition to the protection provided by the network operating system. Data base management systems, in particular, have comprehensive security controls built in to limit access to authorized users.

The WAN

A natural extension of the LAN is the wide area network or WAN. A WAN connects LANS, both locally and remotely, and thus connects remote computers together over long distances. The WAN provides the same functionality as the individual LAN, but on a larger scale where E-mail, applications, and files now move throughout an organization-wide Internet. WANs are, by default, heterogeneous networks that consist of a variety of computers, operating systems, topologies, and protocols. The most popular Internetworking

devices for WANs are bridges and routers. Hybrid units called *brouters* which provide both bridging and routing functions are also appearing. The decision to bridge or route depends on protocols, network topology, and security requirements. Internetworking schemes often include a combination of bridges and routers.

Many organizations today support a variety of networking capabilities for different groups or divisions within their companies. These include LAN to LAN interconnection, gateways to outside company networks, and E-mail backbone capabilities. Network management and security services typically include long-haul data encryption (DES) services.

Network Management

The overall management of a LAN/WAN is highly technical. The ISO's network management model divides network management functions into five subsystems: Fault Management, Performance Management, Configuration Management, Accounting Management, and Security Management. Security management includes controlling access to network resources.

Network management products, such as monitors, network analyzers, and integrated management systems, provide various network status and event history data. These and similar products are designed for troubleshooting and performance evaluation, but can also provide useful information, patterns, and trends for security purposes. For example, a typical LAN analyzer can help the technical staff troubleshoot LAN bugs, monitor network traffic, analyze network protocols, capture data packets for analysis, and assist with LAN expansion and planning. While LAN audit logs can record the user identification code of someone making excessive log-on errors which might not be the owner, it may require a network analyzer to determine the exact identity of the PC on which the log-on errors are occurring. As passive monitoring devices, network analyzers do not log on to a server and are not subject to server-software security. Therefore, analyzer operators should be appropriately screened.

Access Control Mechanisms

Network operating systems have access control mechanisms that are crucial for LAN/WAN security. For example, access controls can limit who can log on, what resources will be available, what each user can do with these resources, and when and from where access is available. Management, LAN, security, and key user personnel should cooperate closely to implement access controls. Security facilities typically included with network operating system software such as Novell NetWare and Banyan Vines include user security, network file access, console security, and network security. These are highlighted below to illustrate the range of security that a LAN can provide.

User security controls determine how, when, and where LAN users will gain access to the system. Setting up user security profiles generally includes the following tasks:

- Specify group security settings
- Specify settings for specific users
- Manage password security — length, expiration, etc., prevent user changes to settings
- Specify log-on settings
- Specify log-on times
- Specify log-out settings
- Specify, modify, and delete log-on locations (workstation, server, and link)
- Delete a user's security
- Specify user dial-in access lists for servers

Network file security is determined by the level of security that is imposed on the directory in which the file resides. Individual files can be secured by employing password protection or other security mechanisms allowed by the specific application software. Each directory has access rights defined to it that consist of an ordered series of user names and access levels.

The console security/selection function allows the system administrator to prevent unauthorized persons from using the operator console. This function allows the system administrator to assign a console password, lock and unlock the console, and change the console type (i.e., assign operator functions to a workstation).

Network security controls determine how outside users and servers can access the resources in the LAN over dial-up lines or intermediate networks or wide area networks. Network security tasks include specifying user dial-up access and specifying Internetwork access.

Future of LANS/WANS

The future direction of computing is increased information sharing across the organization. A host of technologies are evolving to assist companies in reaching this goal. These goals include powerful computers connected to large-bandwidth circuits to move huge amounts of information, open systems architectures to connect various hardware systems, portability of software across multiple systems, and desk-top multi-media capabilities, to name just a few. The center of these evolving technologies is the LAN/WAN. Office networks will continue to grow rapidly, becoming the lifeline of overall organization activity. The goal is to provide transparent access to local office data across mainframes, minicomputers, and PCs. Network security must be included commensurately. The key is to balance

information sharing with information security. The information systems security specialists for the LAN environment of tomorrow will, by necessity, require a high degree of technical hardware and software knowledge.

ASSESSING RISK

In general, risk analysis is used to determine the position an organization should take regarding the risk of loss of assets. Because LANs and WANs represent critical assets to the organization, assessing the risk of loss of these assets is an important management responsibility. The information security industry has used risk analysis techniques for many years. A risk analysis is a formalized exercise that includes:

- Identification, classification, and valuation of assets;
- Postulation and estimation of potential threats;
- Identification of vulnerabilities to threats; and
- Evaluation of the probable effectiveness of existing safeguards and the benefits of additional safeguards.

Protection Needed

The type and relative importance of protection needed for the LAN/WAN must be considered when assessing risk. LAN and WAN systems and their applications need protection in the form of administrative, physical, and technical safeguards for reasons of confidentiality, integrity, and availability.

Confidentiality — The system contains information that requires protection from unauthorized disclosure. Examples of confidentiality include the need for timed dissemination (e.g., the annual budget process), personal data covered by privacy laws, and proprietary business information.

Integrity — The system contains information that must be protected from unauthorized, unanticipated, or unintentional modification, including the detection of such activities. Examples include systems critical to safety or life support and financial transaction systems.

Availability — The system contains information or provides services that must be available on a timely basis to meet mission requirements or to avoid substantial losses. One way to estimate criticality of a system is in terms of downtime. If a system can be down for an extended period at any given time, without adverse impact, it is likely that it is not within the scope of the availability criteria.

For each of the three categories of confidentiality, integrity, and availability, it is necessary to determine the relative protection requirement. These may be defined as:

- **High —** a critical concern of the organization;
- **Medium —** an important concern, but not necessarily paramount in the organization's priorities; or
- **Low —** some minimal level of security is required, but not to the same degree as the previous two categories.

Asset Values

A valuation process is needed to establish the risk or potential for loss in terms of dollars. The greater the value of the assets, the greater the potential loss, and therefore, the greater the need for security. Asset values are useful indicators for evaluating appropriate safeguards for cost effectiveness, but they do not reflect the total tangible and intangible value of information systems. The cost of recreating the data or information could be more than the hardware costs. The violation of confidentiality, the unauthorized modification of important data, or the denial of services at a crucial time could result in substantial costs that are not measurable in monetary terms alone. For example, the accidental or intentional release of premature or partial information relating to investigations, budgets, or contracts could be highly embarrassing to company officials and cause loss of public confidence in the corporation.

Asset valuation should include all computing-associated tangible assets, including LAN/WAN computer hardware, special equipment, and furnishings. Software, data, and documentation are generally excluded since backup copies should be available.

The starting point for asset valuation is the LAN/WAN inventory. A composite summary of inventory items, acquisition value, current depreciated value, and replacement value is one way to provide a reasonable basis for estimating cost effectiveness for safeguards. It should be noted that if a catastrophic loss were to occur, it is unlikely that any organization would replace all hardware components with exact model equivalents. Instead, newer substitute items currently available would probably be chosen, due to the rapid pace of technological improvements.

THREATS TO LAN/WAN SECURITY

A threat is an identifiable risk that has some probability of occurring. Threats are grouped in three broad areas: people threats, virus threats, and physical threats. LANs and WANs are particularly susceptible to people and virus-related threats because of the large number of people who have access rights.

People Threats

The greatest threat posed to LANs and WANs are people — and this threat is primarily from insiders. These are employees who make errors

and omissions and employees who are disgruntled or dishonest. People threats are costly. Employee errors, accidents, and omissions cause some 50 to 60% of the annual dollar losses. Disgruntled employees and dishonest employees add another 20%. These insider threats are estimated to account for over 75% of the annual dollar loss experienced by organizations each year. Outsider threats such as hackers and viruses add another 5%. Physical threats, mainly fire and water damage, add another 20%. It should be noted that these figures were published in 1988, and since that time there has been a dramatic increase in virus incidents, which may significantly enlarge the dollar loss from outsider threats, particularly in the LAN/WAN environment. Some people threats include the following.

System administration error — This area includes all human errors occurring in the setup, administration, and operation of LAN systems, ranging from the failure to properly enable access controls and other security features to the lack of adequate backups. The possible consequences include loss of data confidentiality, integrity, and system availability, as well as possible embarrassment to the company or the individual.

PC operator error — This includes all human errors occurring in the operation of PC/LAN systems, including improper use of log-on/passwords, inadvertent deletion of files, and inadequate backups. Possible consequences include data privacy violations and loss of capabilities, such as the accidental erasure of critical programs or data.

Software/programming error — These errors include all the "bugs," incompatibility issues, and related problems that occur in developing, installing, and maintaining software on a LAN. Possible consequences include degradation, interruption, or loss of LAN capabilities.

Unauthorized disclosure — This is defined as any release of sensitive information on the LAN that is not sanctioned by proper authority, including those caused by carelessness and accidental release. Possible consequences are violations of law and policy, abridgement of rights of individuals, embarrassment to individuals and the company, and loss of shareholder confidence in the company.

Unauthorized use — Unauthorized use is the employment of company resources for purposes not authorized by the corporation and the use of noncompany resources on the network, such as using personally owned software at the office. Possible consequences include the introduction of viruses, and copyright violations for use of unlicensed software.

Fraud/embezzlement — This is the unlawful deletion of company recorded assets through the deceitful manipulation of internal controls, files, and data, often through the use of a LAN. Possible consequences include monetary loss and illegal payments to outside parties.

Modification of data — This is any unauthorized changing of data, which can be motivated by such things as personal gain, favoritism, a misguided sense of duty, or a malicious intent to sabotage. Possible consequences include the loss of data integrity and potentially flawed decision making. A high risk is the disgruntled employee.

Alteration of software — This is defined as any unauthorized changing of software, which can be motivated by such things as disgruntlement, personal gain, or a misguided sense of duty. Possible consequences include all kinds of processing errors and loss of quality in output products.

Theft of computer assets — Theft includes the unauthorized/unlawful removal of data, hardware, or software from company facilities. Possible consequences for the loss of hardware can include the loss of important data and programs resident on the hard disk or on diskettes stored in the immediate vicinity.

Viruses and Related Threats

Computer viruses are the most widely recognized example of a class of programs written to cause some form of intentional disruption or damage to computer systems or networks. A computer virus performs two basic functions: it copies itself to other programs, thereby infecting them, and it executes the instructions the author included in it. Depending on the author's motives, a program infected with a virus may cause damage immediately upon its execution, or it may wait until a certain event has occurred, such as a particular time or date. The damage can vary widely, and can be so extensive as to require the complete rebuilding of all system software and data. Because viruses can spread rapidly to other programs and systems, the damage can multiply geometrically.

Related threats include other forms of destructive programs such as Trojan horses and network worms. Collectively, they are known as malicious software. These programs are often written to masquerade as useful programs, so that users are induced into copying them and sharing them with their friends. The malicious software phenomenon is fundamentally a people problem, as it is frequently authored and often initially spread by individuals who use systems in an unauthorized manner. Thus, the threat of unauthorized use, by both unauthorized and authorized users, must be addressed as a part of virus prevention.

Physical Threats

Electrical power problems are the most frequent physical threat to LANs, but fire or water damage is the most serious. Physical threats generally include the following:

Electrical power failures/disturbances — This is any break or disturbance in LAN power continuity that is sufficient to cause operational interruption, ranging from high-voltage spikes to area "brownouts." Possible consequences range from minor loss of input data to temporary shutdown of systems.

Hardware failure — Hardware failures include any failure of LAN components (particularly disk crashes in PCs). Possible consequences include loss of data or data integrity, loss of processing time, and interruption of services, and may also include degradation or loss of software capabilities.

Fire/water damage — This could include a major catastrophic destruction of an entire building, partial destruction within an office area, LAN room fire, water damage from sprinkler system, and/or smoke damage. The possible consequences include loss of the entire system for extended periods of time.

Other physical threats — These include environmental failures/mishaps involving air conditioning, humidity, heating, liquid leakage, explosion, and contamination. Physical access threats include sabotage/terrorism, riot/civil disorders, bomb threats, and vandalism. Natural disasters include flood, earthquake, hurricane, snow/ice storm, windstorm, tornado, and lightning.

VULNERABILITIES

Vulnerabilities are flaws in the protection of LANs/WANs that can be exploited, partially or fully, by threats resulting in loss. Only a few generic vulnerabilities will be highlighted here, since vulnerabilities are specific weaknesses in a given LAN environment. Vulnerabilities are precluded by safeguards, and a comprehensive list of LAN safeguards is discussed later. Of paramount importance are the most basic safeguards, which are proper security awareness and training.

A LAN exists to provide designated users with shared access to hardware, software, and data. Unfortunately, the LAN's greatest vulnerability is access control. Significant areas of access vulnerability include the PC, passwords, LAN server, and Internetworking.

The Personal Computer

The PC is so vulnerable that user awareness and training are of paramount importance to assure even a minimum degree of protection. PC vulnerable areas include:

Access control — Considerable progress has been made in security management and technology for large-scale centralized data processing environments, but relatively little attention has been given to the protection

of small systems. Most PCs are single-user systems and lack built-in hardware mechanisms that would provide users with security-related systems functions. Without such hardware features (e.g., memory protection), it is virtually impossible to prevent user programs from accessing or modifying parts of the operating system and thereby circumventing any intended security mechanisms.

PC floppy disk drive — The floppy disk drive is a major asset of PC workstations, given its virtually unlimited storage capacity via the endless number of diskettes that can be used to store data. However, the disk drive also provides ample opportunity for sensitive government data to be stolen on floppy disks and for computer viruses to enter the network from literally hundreds of access points. This problem is severe in certain sensitive data environments, and the computer industry has responded with diskless workstations designed specifically for LAN operations. The advantage of diskless PCs is that they solve certain security problems, such as the introduction of unauthorized software (including viruses) and the unauthorized removal of sensitive data. The disadvantage is that the PC workstation becomes a limited, network-dependent unit, not unlike the old "dumb" mainframe terminals.

Hard disk — Most current PCs have internal hard disks ranging from 1 to 2 gigabytes of online storage capacity. Sensitive data residing on these hard disks are vulnerable to theft, modification, or destruction. Even if PC access and LAN access are both password protected, PCs with DOS-based operating systems may be booted from a floppy disk that bypasses the password, permitting access to unprotected programs and files on the hard disk. PC hardware and software security features and products are available to provide increasing degrees of security for data on hard disk drives, ranging from password protection for entering the system to data encryption. "Erasing" hard disks is another problem area. An "erase" or "delete" command does not actually delete a file from the hard disk. It only alters the disk directory or address codes so that it appears as if deletion or erasure of the data has taken place. The information is still there and will be electronically "erased" when DOS eventually writes new files over the old "deleted" files. This may take some time, depending on the available space on the hard disk. In the meantime, various file recovery programs can be used to magically restore the "deleted" file. There are special programs that really do erase a file and these should be used for the removal of sensitive files. A companion issue is that the server may have a copy of the sensitive file, and a user may or may not have erase privileges for the server files.

Repairs — Proper attention must be given to the repair and disposition of equipment. Outside commercial repair staff should be monitored by internal or company technical staff when service is being performed on

sensitive PC/LAN equipment. Excess or surplus hard disks should be properly erased prior to releasing the equipment.

PC Virus

PCs are especially vulnerable to viruses and related malicious software such as Trojan horses, logic bombs, and worms. An executing program, including a virus-infected program, has access to most things in memory or on disk. For example, when DOS activates an application program on a PC, it turns control over to the program for execution. There are virtually no areas of memory protected from access by application programs. There is no block between an application program and the direct usage of system input/output (disk drives, communications, ports, printers, screen displays, etc.). Once the application program is running, it has complete access to everything in the system.

Virus-infected software may have to be abandoned and replaced with uninfected earlier versions. Thus, an effective backup program is crucial in order to recover from a virus attack. Most important, it is essential to determine the source of the virus and the system's vulnerability and institute appropriate safeguards. A LAN/WAN is also highly vulnerable, because any PC can propagate an infected copy of a program to other PCs and possibly the server(s) on the network.

LAN Access

Access Control. A password system is the most basic and widely used method to control access to LANs/WANs. There may be multiple levels of password controls to the LAN and its services, to access to each major application on the LAN, and to other major systems interconnected to the LAN. Conversely, some system access controls depend heavily on the initial LAN log-on/password sequence. While passwords are the most common form of network protection, they are also the weakest from a human aspect. Studies by research groups have found that passwords have many weaknesses, including poor selection of passwords by users (e.g., middle names, birthdays, etc.), poor password administration (e.g., no password guidance, no requirement to change passwords regularly, etc.), and the recording of passwords in easily detected formats (e.g., on calendar pads, in DOS batch files, and even in log-on sequences). Group/multiuser passwords lack accountability and are also vulnerable to misuse.

Dial-In Access. Dial-in telephone access via modems provides a unique window to LANs and WANs, enabling anyone with a user ID, password, and a computer to log into the system. Hackers are noted for their use of dial-in capabilities for access, using commonly available user IDs and cleverly guessing passwords. Effective passwords and log-on procedures, dial-in

time limitations and locations, call-back devices, port protectors, and strong LAN/WAN administration are ways to provide dial-in access control.

UNIX. UNIX is a popular operating system that is often cited for its vulnerabilities, including its handling of "superusers." Whoever has access to the superuser password has access to everything on the system. UNIX was not really designed with security in mind. To complicate matters, new features have been added to UNIX over the years, making security even more difficult to control. Perhaps the most problematic features are those relating to networking, which include remote log-on, remote command execution, network file systems, diskless workstations, and E-mail. All of these features have increased the utility and usability of UNIX by untold amounts. However, these same features, along with the widespread connection of UNIX systems to the Internet and other networks, have opened up many new areas of vulnerabilities to unauthorized abuse of the system.

Internetworking

Internetworking is the connection of the local LAN server to other LAN/WAN servers via various connection devices which consist of routers and gateways. Virtually all organizations with multiple sites or locations use Internetworking technology within their computing environments. E-mail systems could not exist without this interconnectivity. Each additional LAN/WAN interconnection can add outside users and increase the risks to the system. LAN servers and network devices can function as "filters" to control traffic to and from external networks. For example, application gateways may be used to enforce access control policies at network boundaries. The important point is to balance connectivity requirements with security requirements.

The effective administration of LANs/WANs requires interorganizational coordination and teamwork. Since networks can cross so many organizational boundaries, integrated security requires the combined efforts of many personnel, including the administrators and technical staff (who support the local servers, networks, and Internetworks), security personnel, users, and management.

E-mail is the most popular application supported by Internetworking environments. E-mail messages are somewhat different from other computer applications in that they can involve "store and forward" communications. Messages travel from the sender to the recipient, often from one computer to another over a WAN. When messages are stored in one place and then forwarded to multiple locations, they become vulnerable to interception or can carry viruses and related malicious software.

SAFEGUARDS

Safeguards preclude or mitigate LAN vulnerabilities and threats, reducing the risk of loss. No set of safeguards can fully eliminate losses, but a well-planned set of cost-effective safeguards can reduce risks to a reasonable level as determined by management. Safeguards are divided into four major groups: general, technical, operational, and virus. Most of these safeguards also apply to applications as well as to LANs and WANs.

General Safeguards

General safeguards include a broad range of controls that serve to establish a firm foundation for technical and operational safeguards. Strong management commitment and support is required for these safeguards to be effective. General safeguards include, but are not necessarily limited to, the assignment of a LAN/WAN security officer, a security awareness and training program, personnel screening during hiring, separation of duties, and written procedures.

Assignment of LAN/WAN security officer — The first safeguard in any LAN/WAN security program is to assign the security responsibility to a specific, technically knowledgeable person. This person must then take the necessary steps to assure a viable LAN security program, as outlined in a company policy statement. Also, this policy should require that a responsible owner/security individual be assigned to each application, including E-mail and other LAN applications.

Security awareness and training — All employees involved with the management, use, design, acquisition, maintenance, or operation of a LAN must be aware of their security responsibilities and trained in how to fulfil them. Technical training is the foundation of security training. These two categories of training are so interrelated that training in security should be a component of each computer systems training class. Proper technical training is considered to be perhaps the single most important safeguard in reducing human errors.

Personnel screening — Personnel security policies and procedures should be in place and working as part of the process of controlling access to LANs and WANs. Specifically, LAN/WAN management must designate sensitive positions and screen incumbents, which should be described in a company human resource policy manual, for individuals involved in the management, operation, security, programming, or maintenance of systems. Computer security studies have shown that fraud and abuse are often committed by authorized employees. The personnel screening process should also address LAN/WAN repair and maintenance activities, as well as janitorial and building repair crews that may have unattended access to LAN/WAN facilities.

Separation of duties — People within the organization are the largest category of risk to the LAN and WAN. Separation of duties is a key to internal control and should be, designed to make fraud or abuse difficult without collusion. For example, setting up the LAN security controls, auditing the controls, and management review of the results should be performed by different persons.

Written procedures — It is human nature for people to perform tasks differently and inconsistently, even if the same person performs the same task. An inconsistent procedure increases the potential for an unauthorized action (accidental or intentional) to take place on a LAN. Written procedures help to establish and enforce consistency in LAN/WAN operations. Procedures should be tailored to specific LANs and addressed to the actual users, to include the "do's" and "don't's" of the main elements of safe computing practices such as access control (e.g., password content), handling of removable disks and CDs, copyright and license restrictions, remote access restrictions, input/output controls, checks for pirated software, courier procedures, and use of laptop computers. Written procedures are also an important element in the training of new employees.

Technical Safeguards

These are the hardware and software controls to protect the LAN and WAN from unauthorized access or misuse, help detect abuse and security violations, and provide security for LAN applications. Technical safeguards include user identification and authentication, authorization and access controls, integrity controls, audit trail mechanisms, confidentiality controls, and preventive hardware maintenance controls.

User Identification and Authentication. User identification and authentication controls are used to verify the identity of a station, originator, or individual prior to allowing access to the system or to specific categories of information within the system. Identification involves the identifier or name by which the user is known to the system (e.g., a user identification code). This identifying name or number is unique, is unlikely to change, and need not be kept secret. When authenticated, it is used to provide authorization/access and to hold individuals responsible for their subsequent actions.

Authentication is the process of "proving" that the individual is actually the person associated with the identifier. Authentication is crucial for proper security; it is the basis for control and accountability in a system. Following are three basic authentication methods for establishing identity.

Something Known by the Individual. Passwords are presently the most commonly used method of controlling access to systems. Passwords are a combination of letters and numbers (or symbols), preferably comprised of six

or more characters, that should be known only to the accessor. Passwords and log-on codes should have an automated expiration feature, should not be reusable, should provide for secrecy (e.g., nonprint, nondisplay feature, encryption), and should limit the number of unsuccessful access attempts. Passwords should conform to a set of rules established by management.

In addition to the password weaknesses, passwords can be misused. For example, someone who can electronically monitor the channel may also be able to "read" or identify a password and later impersonate the sender. Popular computer network media such as Ethernet or token rings are vulnerable to such abuses. Encryption authentication schemes can mitigate these exposures. Also, the use of one-time passwords has proven effective.

Something Possessed by an Individual. Several techniques can be used in this method. One technique would include a magnetically encoded card (e.g., smart cards) or a key for a lock. Techniques such as encryption may be used in connection with card devices to further enhance their security.

Dial-back is a combination method where users dial in and identify themselves in a prearranged method. The system then breaks the connection and dials the users back at a predetermined number. There are also devices to determine, without the call back, that a remote device hooked to the computer is actually an authorized device.

Other security devices used at the point of log-on and as validation devices on the LAN server include port-protection devices and random number generators.

Something About the Individual. These would include biometric techniques that measure some physical attribute of a person such as a fingerprint, voiceprint, signature, or retinal pattern and transmits the information to the system that is authenticating the person. Implementation of these techniques can be very expensive.

Authorization and Access Controls. These are hardware or software features used to detect and/or permit only authorized access to or within the system. An example of this control would be the use of access lists or tables. Authorization/access controls include controls to restrict access to the operating system and programming resources, limits on access to associated applications, and controls to support security policies on network and Internetwork access.

In general, authorization/access controls are the means whereby management or users determine *who* will have *what* modes of access to *which* objects and resources. The *who* may include not only people and groups, but also individual PCs and even modules within an application. The modes of access typically include read, write, and execute access to data, programs, servers, and Internetwork devices. The objects that are candidates

for authorization control include data objects (directories, files, libraries, etc.), executable objects (commands, programs, etc.), input/output devices (printers, tape backups), transactions, control data within the applications, named groups of any of the foregoing elements, and the servers and Internetwork devices.

Integrity Controls. Integrity controls are used to protect the operating system, applications, and information in the system from accidental or malicious alteration or destruction, and provide assurance to users that data have not been altered (e.g., message authentication). Integrity starts with the identification of those elements that require specific integrity controls. The foundations of integrity controls are the identification/authentication and authorization/access controls. These controls include careful selection of and adherence to vendor-supplied LAN administrative and security controls. Additionally, the use of software packages to automatically check for viruses is effective for integrity control.

Data integrity includes two control mechanisms that must work together and are essential to reducing fraud and error control. These are (1) the well-formed transaction, and (2) segregation of duties among employees. A well-formed transaction has a specific, constrained, and validated set of steps and programs for handling data, with automatic logging of all data modifications so that actions can be audited later. The most basic segregation of duty rule is that a person creating or certifying a well-formed transaction may not be permitted to execute it.

Two cryptographic techniques provide integrity controls for highly sensitive information. Message Authentication Codes (MACs) are a type of cryptographic checksum that can protect against unauthorized data modification, both accidental and intentional. Digital signatures authenticate the integrity of the data and the identity of the author. Digital signature standards are used in E-mail, electronic funds transfer, electronic data interchange, software distribution, data storage, and other applications that require data integrity assurance and sender authentication.

Audit Trail Mechanisms. Audit controls provide a system monitoring and recording capability to retain or reconstruct a chronological record of system activities. An example would be system log files. These audit records help to establish accountability when something happens or is discovered. Audit controls should be implemented as part of a planned LAN security program. LANs have varying audit capabilities, which include exception logging and event recording. Exception logs record information relating to system anomalies such as unsuccessful password or log-on attempts, unauthorized transaction attempts, PC/remote dial-in lockouts, and related matters. Exception logs should be reviewed and retained for specified periods.

Event records identify transactions entering or exiting the system, and journal tapes are a backup of the daily activities.

Confidentiality Controls. These controls provide protection for data that must be held in confidence and protected from unauthorized disclosure. The controls may provide data protection at the user site, at a computer facility, in transit, or some combination of these. Confidentiality relies on comprehensive LAN/WAN security controls which may be complemented by encryption controls.

Encryption is a means of encoding or scrambling data so that they are unreadable. When the data are received, the reverse scrambling takes place. The scrambling and descrambling requires an encryption capability at either end and a specific key, either hardware or software, to code and decode the data. Encryption allows only authorized users to have access to applications and data.

The use of cryptography to protect user data from source to destination, which is called *end-to-end encryption*, is a powerful tool for providing network security. This form of encryption is typically applied at the transport layer of the network (layer 4). End-to-end encryption cannot be employed to maximum effectiveness if application gateways are used along the path between communicating entities. These gateways must, by definition, be able to access protocols at the application layer (layer 7), above the layer at which the encryption is employed. Hence, the user data must be decrypted for processing at the application gateway and then reencrypted for transmission to the destination (or another gateway). In such an event the encryption being performed is not really end-to-end. There are a variety of low-cost, commercial security/encryption products available that may provide adequate protection for unclassified use, some with little or no maintenance of keys. Many commercial software products have security features that may include encryption capabilities, but do not meet the requirements of the DES.

Preventive Maintenance. Hardware failure is an ever-present threat, since LAN and WAN physical components wear out and break down. Preventive maintenance identifies components nearing the point at which they could fail, allowing for the necessary repair or replacement before operations are affected.

Operational Safeguards

Operation safeguards are the day-to-day procedures and mechanisms to protect LANs. These safeguards include backup and contingency planning, physical and environmental protection, production and input/output controls, audit and variance detection, hardware and system software maintenance controls, and documentation.

Backup and Contingency Planning. The goal of an effective backup strategy is to minimize the number of workdays that can be lost in the event of a disaster (e.g., disk crash, virus, fire). A backup strategy should indicate the type and scope of backup, the frequency of backups, and the backup retention cycle. The type/scope of backup can range from complete system backups, to incremental system backups, to file/data backups, or even dual backup disks (disk "mirroring"). The frequency of the backups can be daily, weekly, or monthly. The backup retention cycle could be defined as daily backups kept for a week, weekly backups kept for a month, or monthly backups kept for a year.

Contingency planning consists of workable procedures for continuing to perform essential functions in the event that information technology support is interrupted. Application plans should be coordinated with the backup and recovery plans of any installations and networks used by the application. Appropriate emergency, backup, and contingency plans and procedures should be in place and tested regularly to assure the continuity of support in the event of system failure. These plans should be known to users and coordinated with them. Offsite storage of critical data, programs, and documentation is important. In the event of a major disaster such as fire, or even extensive water damage, backups at offsite storage facilities may be the only way to recover important data, software, and documentation.

Physical and Environmental Protection. These are controls used to protect against a wide variety of physical and environmental threats and hazards, including deliberate intrusion, fire, natural hazards, and utility outages or breakdowns. Several areas come within the direct responsibility of the LAN/WAN personnel and security staff including adequate surge protection, battery backup power, room and cabinet locks, and possibly additional air-conditioning sources. Surge protection and backup power will be discussed in more detail.

Surge suppressors that protect stand-alone equipment may actually cause damage to computers and other peripherals in a network. Ordinary surge protectors and uninterruptible power supplies (UPS) can actually divert dangerous electrical surges into network data lines and damage equipment connected to that network. Power surges are momentary increases in voltage of up to 6,000 volts in 110-volt power systems, making them dangerous to delicate electronic components and data as they search for paths to ground. Ordinary surge protectors simply divert surges from the hot line to the neutral and ground wires, where they are assumed to flow harmlessly to earth. The extract below summarizes this surge protection problem for networks.

> Computers interconnected by data lines present a whole new problem because network data lines use the powerline ground circuit for signal voltage reference. When a conventional surge protector diverts a surge

to ground, the surge directly enters the data lines through the ground reference. This causes high surge voltages to appear across data lines between computers, and dangerous surge currents to flow in these data lines. TVSSs (Transient Voltage Surge Suppressors) based on conventional diversion designs should not be used for networked equipment. Surge protectors may contribute to LAN crashes by diverting surge pulses to ground, thereby contaminating the reference used by data cabling. To avoid having the ground wire act as a "back door" entry for surges to harm a computer's low-voltage circuitry, network managers should consider powerline protection that (1) provides low let-through voltage, (2) does not use the safety ground as a surge sink and preserves it for its role as voltage reference, (3) attenuates the fast rise times of all surges, to avoid stray coupling into computer circuitry, and (4) intercepts all surge frequencies, including internally generated high-frequency surges.

The use of an UPS for battery/backup power can make the difference between a "hard or soft crash." Hard crashes are the sudden loss of power and the concurrent loss of the system, including all data and work in progress in the servers' random access memory (RAM). An UPS provides immediate backup power to permit an orderly shutdown or "soft crash" of the LAN, thus saving the data and work in progress. The UPS protecting the server should include software to alert the entire network of an imminent shutdown, permitting users to save their data. LAN servers should be protected by UPSs, and UPS surge protectors should avoid the "back door" entry problems described above.

Production and Input/Output Controls. These are controls over the proper handling, processing, storage, and disposal of input and output data and media, including locked storage of sensitive paper and electronic media, and proper disposal of materials (i.e., erasing/degaussing diskettes/tape and shredding sensitive paper material).

Audit and Variance Detection. These controls allow management to conduct an independent review of system records and activities in order to test for adequacy of system controls, and to detect and react to departures from established policies, rules, and procedures. Variance detection includes the use of system logs and audit trails to check for anomalies in the number of system accesses, types of accesses, or files accessed by users.

Hardware and System Software Maintenance Controls. These controls are used to monitor the installation of and updates to hardware and operating system and other system software to ensure that the software functions as expected and that an historical record is maintained of system changes. They may also be used to ensure that only authorized software is allowed on the system. These controls may include a hardware and system software

configuration policy that grants managerial approval to modifications, then documents the changes. They may also include virus protection products.

Documentation. Documentation controls are in the form of descriptions of the hardware, software, and policies, standards, and procedures related to LAN security, and include vendor manuals, LAN procedural guidance, and contingency plans for emergency situations. They may also include network diagrams to depict all interconnected LANs/WANs and the safeguards in effect on the network devices.

Virus Safeguards

Virus safeguards include the good security practices cited above which include backup procedures, the use of only company approved software, and procedures for testing new software. All organizations should require a virus prevention and protection program, including the designation and training of a computer virus specialist and backup. Each LAN should be part of this program. More stringent policies should be considered as needed, such as:

- Use of antivirus software to prevent, detect, and eradicate viruses;
- Use of access controls to more carefully limit users;
- Review of the security of other LANs before connecting;
- Limiting of E-mail to nonexecutable files; and,
- Use of call-back systems for dial-in lines.

Additionally, there are several other common-sense tips which reduce the exposure to computer viruses. If the software allows it, apply write-protect tabs to all program disks before installing new software. If it does not, write protect the disks immediately after installation. Also, do not install software without knowing where it has been. Where applicable, make executable files read-only. It won't prevent virus infections, but it can help contain those that attack executable files (e.g., files that end in ".exe" or ".com"). Designating executable files as read-only is easier and more effective on a network, where system managers control read/write access to files. Finally, back up the files regularly. The only way to be sure the files will be around tomorrow is to back them up today.

METHOD OF ANALYSIS

Analysis methodologies may range from informal reviews of small office automation installations through formal risk assessments at major data centers. An informal security review can be used for systems with low-level risk designations. Formal security assessments should be required for high-level risk environments. Below is a further discussion of levels of protection.

Automated Risk Assessment

There are a considerable number of automated risk assessment packages, of varying capabilities and costs, available in the marketplace. These automated packages address large and medium facilities, applications, office automation, and LAN/WAN environments. Several packages contain general analyses of network vulnerabilities applicable to LANs. These packages have been found to have adequate coverage of LAN administration, protection of file servers, and PC/LAN backup practices and procedures.

Questionnaires and Checklists

The key to good security management is measurement — knowing where one is in relation to what needs to be done. Questionnaires are one way to gather relevant information from the user community. A PC/LAN questionnaire can be a simple, quick, and effective tool to support informal and formal risk assessments. For small, informal risk assessments, the PC/LAN questionnaire can be the main assessment tool. A checklist is another valuable tool for helping to evaluate the status of security.

A customized version of an automated questionnaire and assessment can be developed by security consultants as well. With this approach, the user is prompted to respond to a series of PC and LAN questions which are tailored online to the user's environment, and then provides recommendations to improve the user's security practices and safeguards. Typically designed for the average PC user, this approach functions as a risk assessment tool. A questionnaire/checklist may be a useful first step in determining if a more formal/extensive risk assessment needs to be done, as well as to guide the direction of the risk assessment.

LAN/WAN SECURITY IMPLEMENTATION

This section provides a step by step approach for implementing cost-effective LAN/WAN security. A simple example is used to illustrate this approach. The steps performed in the implementation process include determining and reviewing responsibilities, determining required procedures, determining security level requirements, and determining detailed security procedures.

Determine/Review Responsibilities

The first step in LAN/WAN security implementation is to know who is responsible for doing what. LAN/WAN security is a complex undertaking, requiring an integrated team effort. Responsibilities must be defined for managers of facilities, information technology operations personnel, and managers of application systems which run on LANs.

In addition, every area network should require a LAN/WAN administrator and an information systems security officer whose specific duties include the implementation of appropriate general, technical (e.g., access controls and Internetwork security), and operational controls (e.g., backups and contingency planning). In general, the security officer is responsible for the development and coordination of LAN and WAN security requirements, including the "Computer Systems Security Plan". The LAN/WAN administrator is responsible for the proper implementation and operation of security features on the LAN/WAN.

Determine Required Procedures

The second step is to understand the type and relative importance of protection needed for a LAN. As stated above, a LAN may need protection for reasons of confidentiality, integrity, and availability. For each of the three categories there are three subcategories to determine the level of security needed: High, Medium, or Low. A matrix approach can be used to document the conclusions for needed security. This involves ranking the security objectives for the LAN being reviewed, using the following simple matrix.

Typical Security Matrix

Security Objectives	Level of Protection Needed		
	High (Level 3)	Medium (Level 2)	Low (Level 1)
Confidentiality			
Integrity			
Availability			
Overall			

The result is an overall security designation of low (Level 1), medium (Level 2), or high (Level 3). In all instances, the security level designation of a LAN should be equal to or higher than the highest security level designation of any data it processes or systems it runs. This security level designation determines the minimum security safeguards required to protect sensitive data files and to ensure the operational continuity of critical processing capabilities.

This matrix analysis approach to documenting security designations can be expanded and refined into more complex models with security objective subcategories and possibly the use of weighted value assignments for categories. Most automated packages are based on more complex measurement models.

Determine Security Level Requirements

Once the level of protection has been determined, the next step is to determine the security level requirements. Using the simple model that has been created to illustrate this approach, the following is a suggested definition of the minimum security requirements for each level of protection.

Level 1 Requirements. The suggested controls required to adequately safeguard a Level 1 system are considered good management practices. These include, but are not limited, to the following.

1. Information systems security awareness and training.
2. Position sensitivity designations.
3. Physical access controls.
4. A complete set of information systems and operations documentation.

Level 2 Requirements. The suggested controls required to adequately safeguard a Level 2 system include all of the requirements for Level 1, plus the following requirements.

1. A detailed risk management program.
2. Record retention procedures.
3. A list of authorized users.
4. Security review and certification procedures.
5. Clearance (i.e., appropriate background checks) for persons in sensitive positions.
6. A detailed fire/catastrophe plan.
7. A formal written contingency plan.
8. A formal risk analysis.
9. An automated audit trail.
10. Authorized access and control procedures.
11. Secure physical transportation procedures.
12. Secure telecommunications.
13. An emergency power program.

Level 3 Requirements. The suggested controls required to adequately safeguard a Level 3 system include all of the requirements for Levels 1 and 2, plus the following.

1. More secure data transfer, maybe including encryption.
2. Additional audit controls.
3. Additional fire prevention requirements.
4. Provision of waterproof covers for computer equipment.
5. Maintenance of a listing of critical-sensitive clearances.

Determine Detailed Security Procedures

The matrix model and suggested security requirements described above illustrate a very general simple approach for documenting the security implementation requirements. To proceed with the implementation, specific, detailed security protections must be determined, starting with who gets what access, and when. Management, LAN personnel, and security officials, working with key users, must determine the detailed security protections. Procedures for maintaining these protections must be formalized (e.g., who reviews audit logs; who notifies the LAN administrator of departed personnel) to complete the security implementation requirements phase.

DEVELOP AN INTEGRATED SECURITY APPROACH

The final step is the development of an integrated security approach for a LAN/WAN environment. The approach involves the culmination of areas described above into one integrated comprehensive approach. Areas discussed below that are included within the integrated approach are: the use of PC/LAN questionnaires, the role of the Computer System Security Plan, risk assessment, annual review and training, and annual management reporting and budgeting.

Role of the PC/LAN Questionnaire

Security programs require the gathering of a considerable amount of information from managers, technical staff, and users. Interviews are one way, and these are often used with the technical staff. Another way to obtain information is with a PC questionnaire, which is a particularly good method for reaching a reasonable segment of the user community, quickly and efficiently. With minor updating, these surveys can be used periodically to provide a current picture of the security environment.

A PC/LAN questionnaire is suggested for Level 1 reviews and to support Level 2 and 3 risk assessments. In other words, a questionnaire can be the focus of an informal risk assessment and can be a major element in a formal risk assessment. A PC/LAN questionnaire, for example, can collect the information to help identify applications and general purpose systems, identify sensitivity and criticality, and determine specific additional security needs relating to security training, access controls, backup and recovery requirements, input/output controls, and many other aspects of security. This questionnaire can be passed out to a representative sampling of PC users, from novices to experienced users, asking them to take 15 to 20 minutes to fill out the form. The aggregated results of this questionnaire should provide a reasonable number of indicators to assess the general status of PC computing practices within the LAN/WAN environment.

Role of the Computer System Security Plan

A Computer Systems Security Plan (CSSP) is suggested for development of Level 2 and Level 3 LANs and WANs. CSSPs are an effective tool for organizing LAN security. The CSSP format provides simplicity, uniformity, consistency, and scalability. The CSSP is to be used as the risk management plan for controlling all recurring requirements, including risk updates, personnel screening, training, etc.

Risk Assessment

Risk assessment includes the identification of informational and other assets of the system, threats that could affect the confidentiality, integrity, or availability of the system, system vulnerabilities/susceptibility to the threats, potential impacts from threat activity, identification of protection requirements to control the risks, and selection of appropriate security measures. Risk assessment for general purpose systems, including LANs/WANs, are suggested for use at least every five years, or more often when there are major operational, software, hardware, or configuration changes.

Annual Review and Training Session

An ideal approach would be to conduct a yearly LAN/WAN meeting where LAN/WAN management, security, and end-user personnel can get together and review the security of the system. LAN/WAN meetings are an ideal way to satisfy both the security needs/updates of the system and the training/orientation needs of the individuals who are associated with the system. The process can be as simple as reviewing the CSSP, item by item, for additions, changes, and deletions. General discussion on special security topics such as planned network changes and management concerns can round out the agenda. A summary of the meeting is useful for personnel who were unable to attend, for managers, and for updating the management plan.

An often overlooked fact is that LAN/WAN security is only as good as the security being practiced. Information and system security is dependent on each user. Users need to be sensitized, trained, and monitored to ensure good security practices.

Update Management/Budget Plan

The management/budget plan is the mechanism for getting review and approval of security requirements in terms of specific projects, descriptions, responsibilities, schedule, and costs. This plan should be updated yearly to reflect the annual review findings.

20

Security and Network Technologies

Chris Hare, CISSP, CISA

While it is common for security people to examine issues regarding network connectivity, there can be some level of mysticism associated with the methods and technologies that are used to actually construct the network. This chapter addresses what a network is, and the different methods that can be used to build one. It also introduces issues surrounding the security of the network.

People send voice, video, audio, and data through networks. People use the Internet for bank transactions. People look up information in encyclopedias online. People keep in touch with friends and family using e-mail and video. As so much information is now conveyed in today's world through electronic means, it is essential that the security practitioner understands the basics of the network hardware used in today's computer networks.

What Is a Network?

A network is two or more devices connected together in such a way as to allow them to exchange information. When most people think of a network, they associate it with a computer network — ergo, the ability of two or more computers to share information among them. In fact, there are other forms of networks. Networks that carry voice, radio, or television signals. Even people establish networks of contacts — those people with whom they meet and interact.

In the context of this chapter, the definition is actually the first one: two or more devices that exchange information over some form of communication system.

Network Devices

Network devices are computer or topology-specific devices used to connect the various network segments together to allow for data communication between different systems. Such devices include repeaters, bridges, routers, and switches.

Hubs

Hubs are used to concentrate a series of computer connections into one location. They are used with twisted-pair wiring systems to interconnect the systems. Consider the traditional Ethernet network where each station is connected to a single network cable. The twisted-pair network is unlike this; it is physically a star network. Each cable from a station is electrically connected to the others through a hub.

Hubs can be passive or active. A passive hub simply splits the incoming signal among all of the ports in the device. Active hubs retransmit the received signal into the other access ports. Active hubs support remote monitoring and support, while passive hubs do not.

The term "hub" is often extended to bridges, repeaters, routers, switches, or any combination of these.

Repeaters

A repeater retransmits the signal on one network segment to another segment with the original signal strength. This allows for very long networks when the actual maximum distance associated with a particular medium is not. For example, the 10Base5 network standard allows for a maximum of four repeaters between two network stations. Because a coaxial segment can be up to 1500 meters, the use of the repeater significantly increases the length of the network.

Bridges

Bridges work by reading information in the physical data frames and determining if the traffic is for the network on the other side of the bridge. They are used in both Token Ring and Ethernet networks. Bridges filter the data they transmit from one network to another by only copying the frames that they should, based upon the destination address of the frame.

Routers

Routers are more sophisticated tools for routing data between networks. They use the information in the network protocol (e.g., IP) packet to determine where the packet is to be routed. They are capable of collecting and storing information on where to send packets, based on defined configurations or information that they receive through routing protocols. Many routers are only capable of two network connections, while larger-scale routers can handle hundreds of connections to different media types.

Switches

A switch is essentially a multi-port bridge, although the term is now becoming more confusing. Switches have traditionally allowed for the connection of multiple networks for a certain length of time, much like a rotary switch. Two, and only two, networks are connected together for the required time period. However, today's switches not only incorporate this functionality, but also include routing intelligence to enhance their capability.Network Types

Networks can be large or small. Many computer hobbyists operate small, local area networks (LANs) within their own home. Small businesses also operate small LANs. Exactly when a LAN becomes something other than a LAN can be an issue for debate; however, a simpler explanation exists.

A LAN, as illustrated in Exhibit 20.1, connects two or more computers together, regardless of whether those computers are in the same room or on the same floor of a building. However, a LAN is no longer a LAN when it begins to expand into other areas of the local geography. For example, the organization that has two offices at opposite ends of a city and operates two LANs, one in each location. When they extend those two LANs to connect to each other, they have created a metropolitan area network (MAN); this is illustrated in Exhibit 20.2.

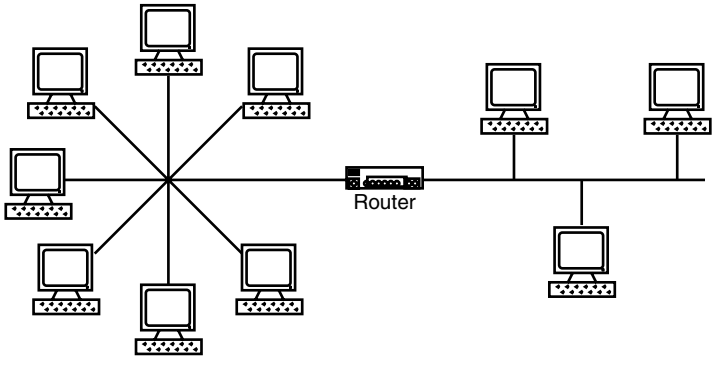


EXHIBIT 20.1 Sample local area network.

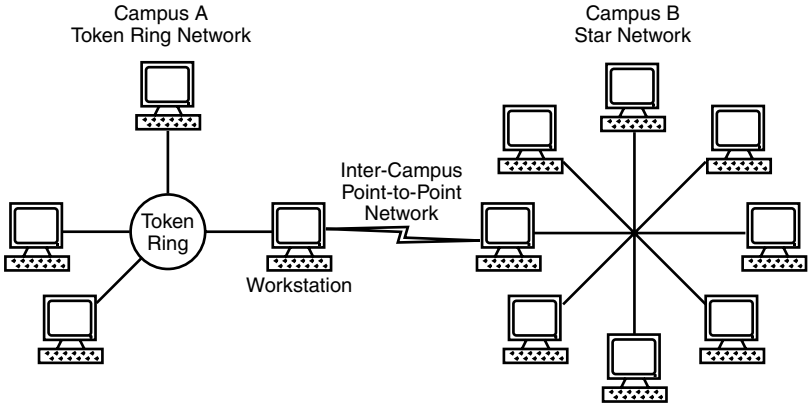


EXHIBIT 20.2 Sample metropolitan area network.

Note that a MAN is only applicable if two or more sites are within the same geographical location. For example, if the organization has two offices in New York City as illustrated in Exhibit 20.2, they operate a MAN. However, if one office is in New York and the other is in San Francisco (as shown in Exhibit 20.3), they no longer operate a MAN, but rather a WAN (i.e., wide area network).

These network layouts are combined to form inter-network organizations and establish a large collection of networks for information sharing. In fact, this is what the Internet is: a collection of local, metropolitan, and wide area networks connected together.

However, while networks offer a lot to the individual and the organization with regard to putting information into the hands of those who need it regardless of where they are, they offer some significant disadvantages.

It used to be that if people wanted to steal something, they had to break into a building, find the right desk or filing cabinet, and then physically remove something. Because information is now stored online, people have more information to lose, and more ways to lose it.

No longer do "burglars" need to break into the physical premises; they only have to find a way onto a network and achieve the same purpose. However, the properly designed and secured network offers more advantages to today's organizations than disadvantages.

However, a network must have a structure. That structure (or topology) can be as simple as a point-to-point connection, or as complicated as a multi-computer, multi-segment network.Network Topologies

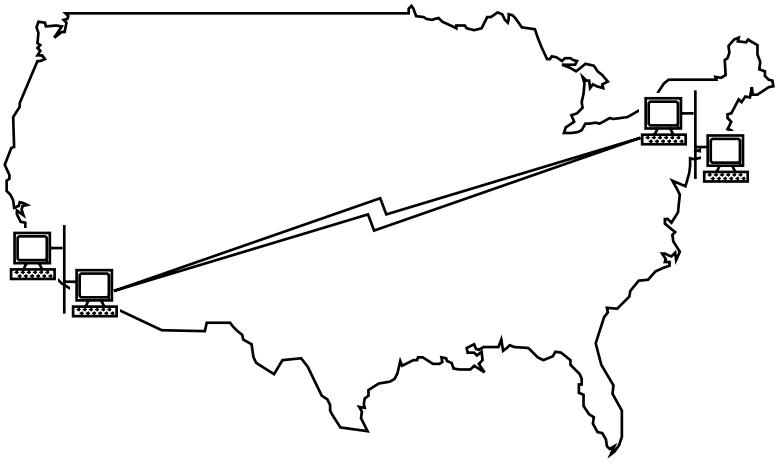


EXHIBIT 20.3 Sample wide area network.

EXHIBIT 20.4 Point-to-point network.

A network consists of segments. Each segment can have a specific number of computers, depending on the cable type used in the design. These networks can be assembled in different ways.

Point-to-Point

A point-to-point network consists of exactly two network devices, as seen in Exhibit 20.4. In this network layout, the two devices are typically connected via modems and a telephone line. Other physical media may be used, for example twisted pair, but the applications outside the phone line are quite specific. In this type of network, the attacks are based at either the two computers themselves, or at the physical level of the connection. Because the connection itself can be carried by an analog modem, it is possible to eavesdrop on the sound and create a data stream that another computer can understand.

Bus

The bus network (see Exhibit 20.5) is generally thought of when using either 10Base2 or 10Base5 coaxial cabling. This is because the electrical architecture of this cabling causes it to form a bus or electrical length. The computers are generally attached to the cable using a connector that is dependent on cable type.

Bus networks can have a computer or network sniffer added on to them without anyone's knowledge as long as the physical limitations of the cabling have not been exceeded. If there is a spare, unused connector, then it is not difficult to add a network sniffer to capture network traffic.

Daisy Chain

The daisy-chain network as seen in Exhibit 20.6 is used in the thin-client or 10Base2 coaxial network. When connecting stations in this environment, one can either create a point-to-point connection where systems are linked together using multiple dialup or point-to-point links, or connect station to station.

The illustration suggests that the middle station has two network cards. This is not the case, however; it was drawn in this exaggerated fashion to illustrate that the systems are *chained* together. In the case of the thin-client network, the connections are made using two pieces of cable and a T-connector, which is then attached directly to the workstation, as shown in Exhibit 20.7.

This example illustrates how systems are daisy-chained, and specifically how it is accomplished with the 10Base2 or thin-client network.

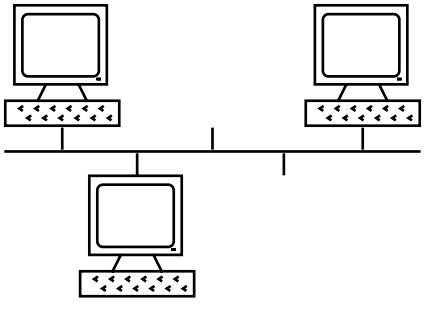


EXHIBIT 20.5 Sample bus network.

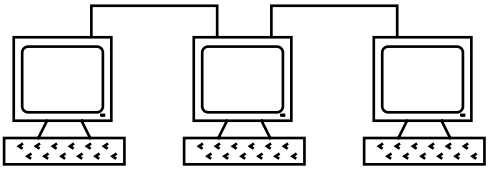


EXHIBIT 20.6 Sample daisy chain network.

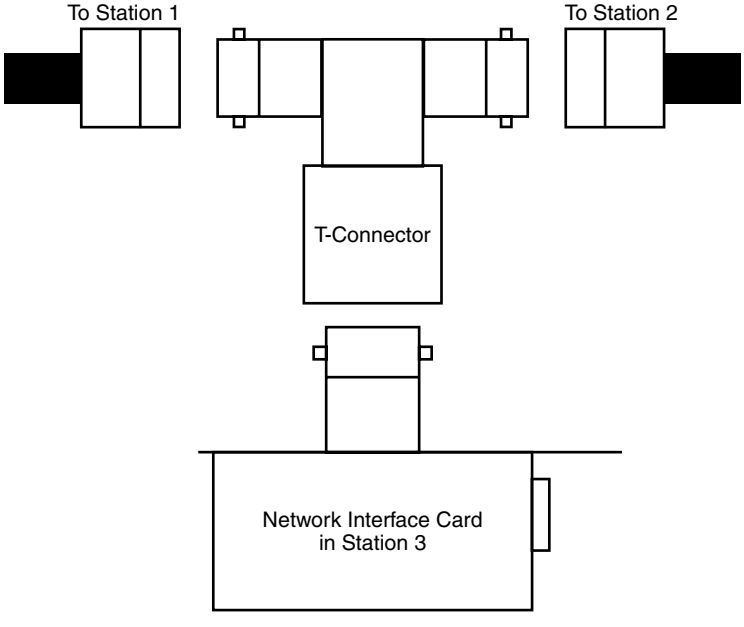


EXHIBIT 20.7 Thin-client connections.

Star

Star networks (Exhibit 20.8) are generally seen in twisted-pair type environments, in which each computer has its own connection or segment between it and the concentrator device in the middle of the star. All the connections are terminated on the concentrator that electrically links the cables together to form the network. This concentrator is generally called a hub.

This network layout has the same issues as the bus. It is easy for someone to replace an authorized computer or add a sniffer at an endpoint of the star or at the concentrator in the middle.

Ring

The ring network (Exhibit 20.9) is most commonly seen in IBM Token Ring networks. In this network, a token is passed from computer to computer. No computer can broadcast a packet unless it has the token. In this way, the token is used to control when stations are allowed to transmit on the network.

However, while a Token Ring network is the most popular place to "see" a ring, a Token Ring network as illustrated in Exhibit 20.9 is electrically a star. A ring network is also achieved when each system only knows how to communicate with two other stations, but are linked together to form a ring, as illustrated in Exhibit 20.10. This means that it is dependent on those two other systems to know how to communicate with other systems that may be reachable.

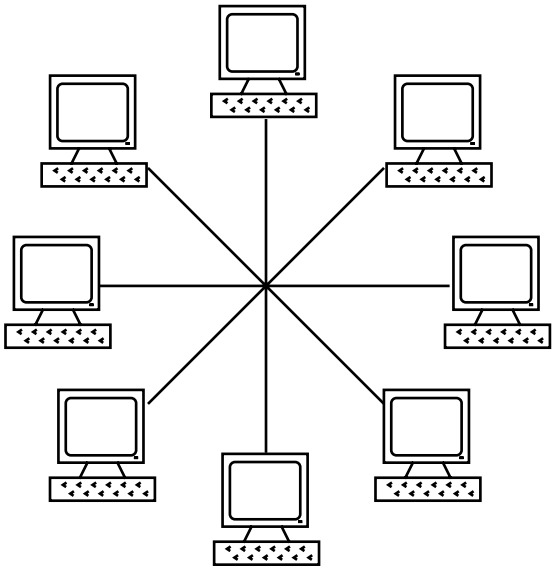


EXHIBIT 20.8 Sample star network.

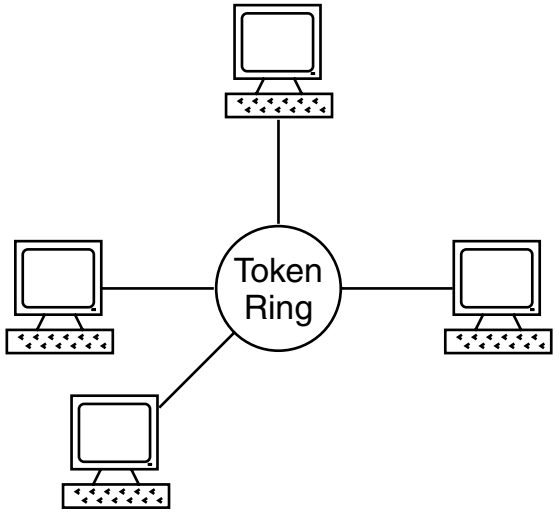


EXHIBIT 20.9 Token Ring network.

Web

The Web network (Exhibit 20.11) is complex and difficult to maintain on a large scale. It requires that each and every system on the network knows how to contact any other system. The more systems in use, the larger and more difficult the configuration files. However, the Web network has several distinct advantages over any of the previous networks.

It is highly robust, in that multiple failures will still allow the computer to communicate with other systems. Using the example shown in Exhibit 20.11, a single system can experience up to four failures. Even at four failures, the system still maintains communication within the Web. The system must experience total communication loss or be removed from the network for data to not move between the systems.

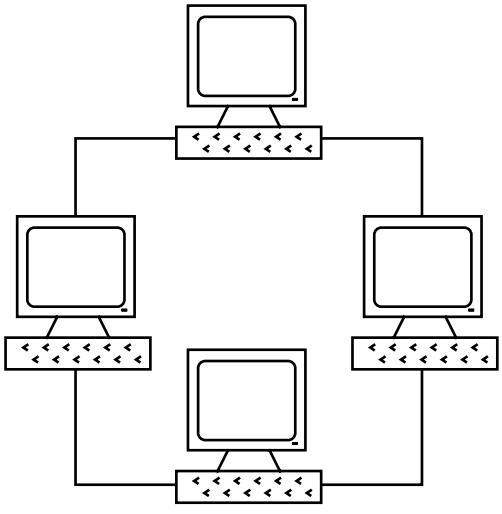


EXHIBIT 20.10 Ring network.

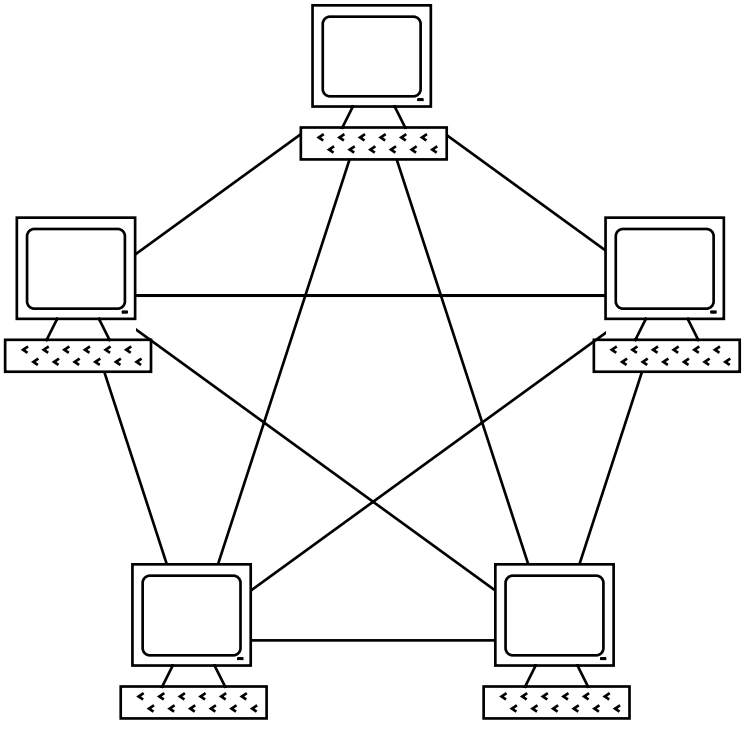


EXHIBIT 20.11 Web network.

This makes the Web network extremely resilient to network failures and allows data movement even in high failure conditions. Organizations will choose this network type for these features, despite the increased network cost in circuits and management.

Each of the networks described previously relies on specific network hardware and topologies to exchange information. To most people, the exact nature of the technology used and the operation is completely transparent; and for the most part, it is intended to be that way.

Network Formats

Network devices must be connected using some form of physical medium. Most commonly, this is done through cabling. However, today's networks also include wireless, which can be extended to desktop computers, or to laptop or palmtop devices connected to a cellular phone. There are several different connection methods; however, the most popular today are Ethernet and Token Ring.

Serious discussions about both of these networks, their associated cabling, devices, and communications methods can easily fill large books. Consequently, this chapter only provides a brief discussion of the history and different media types available.

Ethernet

Ethernet is, without a doubt, the most widely used local area network (LAN) technology. While the original and most popular version of Ethernet supported a data transmission speed of 10 Mbps, newer versions have evolved, called Fast Ethernet and Gigabit Ethernet, that support speeds of 100 Mbps and 1000 Mbps.

Ethernet LANs are constructed using coaxial cable, special grades of twisted-pair wiring, or fiber-optic cable. Bus and star wiring configurations are the most popular by virtue of the connection methods to attach devices to the network. Ethernet devices compete for access to the network using a protocol called Carrier Sense Multiple Access with Collision Detection (CSMA/CD).

Bob Metcalfe and David Boggs of the Xerox Palo Alto Research Center (PARC) developed the first experimental Ethernet system in the early 1970s. It was used to connect the lab's Xerox Alto computers and laser printers at a (modest, but slow by today's standards) data transmission rate of 2.94 Mbps. This data rate was chosen because it was derived from the system clock of the Alto computer. The Ethernet technologies are all based on a 10 Mbps CSMA/CD protocol.

10Base5

This is often considered the grandfather of networking technology, as this is the original Ethernet system that supports a 10-Mbps transmission rate over "thick" (10 mm) coaxial cable. The "10Base5" identifier is shorthand for **10**-Mbps transmission rate, the baseband form of transmission, and the **5**00-meter maximum supported segment length. In a practical sense, this cable is no longer used in many situations. However, a brief description of its capabilities and uses is warranted.

In September 1980, Digital Equipment Corp., Intel, and Xerox released Version 1.0 of the first Ethernet specification, called the DIX standard (after the initials of the three companies). It defined the "thick" Ethernet system (10Base5), "thick" because of the thick coaxial cable used to connect devices on the network.

To identify where workstations can be attached, 10Base5 thick Ethernet coaxial cabling includes a mark every 2.5 meters to mark where the transceivers (multiple access units, or MAUs) can be attached. By placing the transceiver at multiples of 2.5 meters, signal reflections that may degrade the transmission quality are minimized.

10Base5 transceiver taps are attached through a clamp that makes physical and electrical contact with the cable that drills a hole in the cable to allow electrical contact to be made (see Exhibit 20.12). The transceivers are called non-intrusive taps because the connection can be made on an active network without disrupting traffic flow.

Stations attach to the transceiver through a transceiver cable, also called an attachment unit interface, or AUI. Typically, computer stations that attach to 10Base5 include an Ethernet network interface card (NIC) or adapter card with a 15-pin AUI connector. This is why many network cards even today still have a 15-pin AUI port.

A 10Base5 coaxial cable segment can be up to 500 meters in length, and up to 100 transceivers can be connected to a single segment at any multiple of 2.5 meters apart. A 10Base5 segment may consist of a single continuous section of cable or be assembled from multiple cable sections that are attached end to end.

10Base5 installations are very reliable when properly installed, and new stations are easily added by tapping into an existing cable segment. However, the cable itself is thick, heavy, and inflexible, making installation a

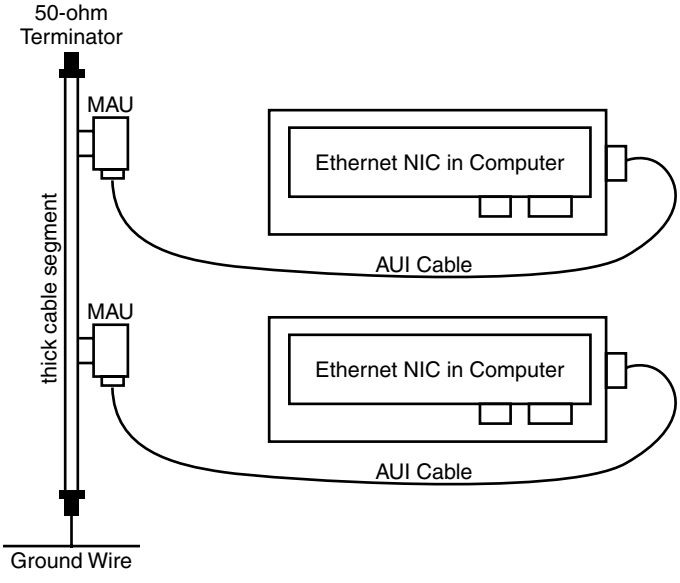


EXHIBIT 20.12 10Base5 station connections.

challenge. In addition, the bus topology makes problem isolation difficult, and the coaxial cable does not support higher-speed networks that have since evolved.

10Base2

A second version of Ethernet called "thin" Ethernet, "cheapernet," or 10Base2 became available in 1985. It used a thinner, cheaper coaxial cable that simplified the cabling of the network. Although both the thick and thin systems provided a network with excellent performance, they utilized a bus topology that made implementing changes in the network difficult and also left much to be desired with regard to reliability. It was the first new variety of physical medium adopted after the original thick Ethernet standard.

While both the thin and thick versions of Ethernet have the same network properties, the thinner cable used by 10Base2 has the advantages of being cheaper, lighter, more flexible, and easier to install than the thick cable used by 10Base5. However, the thin cable has the disadvantage that its transmission characteristics are not as good. It supports only a 185-meter maximum segment length (versus 500 meters for 10Base5) and a maximum of 30 stations per cable segment (versus 100 for 10Base5).

Transceivers are connected to the cable segment through a BNC Tee connector and not through tapping as with 10Base5. As the name implies, the BNC Tee connector is shaped like the letter "T." Unlike 10Base5, where one can add a new station without affecting data transmission on the cable, one must "break" the network to install a new station with 10Base2, as illustrated in Exhibit 20.13. This method of adding or removing stations is due to the connectors used, as one must cut the cable and insert the BNC Tee connector to allow a new station to be connected. If care is not taken, it is possible to interrupt the flow of network traffic due to an improperly assembled connector.

The BNC Tee connector either plugs directly into the Ethernet network interface card (NIC) in the computer station or to an external thin Ethernet transceiver that is then attached to the NIC through a standard AUI cable. If stations are removed from the network, the BNC Tee connector is removed and replaced with a BNC Barrel connector that provides a straight-through connection.

The thin coaxial cable used in the 10Base2 installation is much easier to work with than the thick cable used in 10Base5, and the cost of implementing the network is lower due to the elimination of the external transceiver. However, the typical installation is based on the daisy-chain model illustrated in Exhibit 20.6 which results in lower reliability and increased difficulty in troubleshooting. Furthermore, in some office environments, daisy-chain segments can be difficult to deploy, and like 10Base5, thin-client networks do not support the higher network speeds.

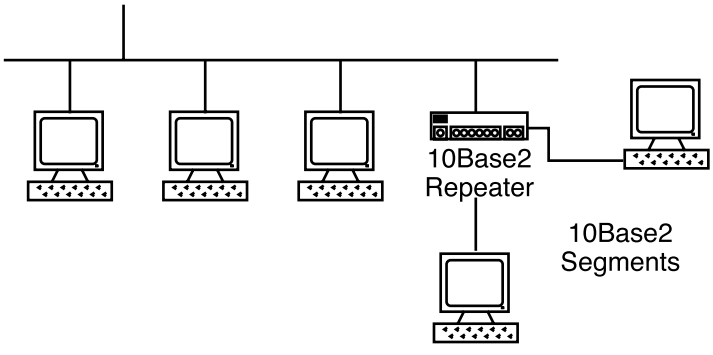


EXHIBIT 20.13 10Base2 network.

10Base-T

Like 10Base2 and 10Base5 networks, 10Base-T also supports only a 10-Mbps transmission rate. Unlike those technologies, however, 10Base-T is based on voice-grade or Category 3 or better telephone wiring. This type of wiring is commonly known as twisted pair, of which one pair of wires is used for transmitting data, and another pair is used for receiving data. Both ends of the cable are terminated on an RJ-45 eight-position jack. The widespread use of twisted pair wiring has made 10Base-T the most popular version of Ethernet today.

All 10Base-T connections are point-to-point. This implies that a 10Base-T cable can have a maximum of two Ethernet transceivers (or MAUs), with one at each end of the cable. One end of the cable is typically attached to a 10Base-T repeating hub. The other end is attached directly to a computer station's network interface card (NIC) or to an external 10Base-T transceiver. Today's NICs have the transceiver integrated into the card, meaning that the cable can now be plugged in directly, without the need for an external transceiver. If one is unfortunate enough to have an older card with an AUI port but no RJ-45 jack, the connection can be achieved through the use of an inexpensive external transceiver.

It is not a requirement that 10Base-T wiring be used only within a star configuration. This method is often used to connect two network devices together in a point-to-point link. In establishing this type of connection, a crossover cable must be used to link the receive and transmit pairs together to allow for data flow. In all other situations, a straight-through or normal cable is used.

The target segment length for 10Base-T with Category 3 wiring is 100 meters. Longer segments can be accommodated as long as signal quality specifications are met. Higher quality cabling such as Category 5 wiring may be able to achieve longer segment lengths, on the order of 150 meters, while still maintaining the signal quality required by the standard.

The point-to-point cable connections of 10Base-T result in a star topology for the network, as illustrated in Exhibit 20.14. In a star layout, the center of the star holds a hub with point-to-point links that appear to radiate out from the center like light from a star. The star topology simplifies maintenance, allows for faster troubleshooting, and isolates cable problems to a single link.

The independent transmit and receive paths of the 10Base-T media allow the full-duplex mode of operation to be optionally supported. To support full-duplex mode, both the NIC and the hub must be capable of, and be configured for, full-duplex operation.

10Broad36

10Broad36 is not widely used in a LAN environment. However, because it can be used in a MAN or WAN situation, it is briefly discussed. 10Broad36 supports a 10-Mbps transmission rate over a broadband cable system. The "36" in the name refers to the 3600-meter total span supported between any two stations, and this type of network is based on the same inexpensive coaxial cable used in cable TV (CATV) transmission systems.

Baseband network technology uses the entire bandwidth of the transmission medium to transmit a single electrical signal. The signal is placed on the medium by the transmitter with no modulation. This makes baseband technology cheaper to produce and maintain and is the technology of choice for all of the Ethernet systems discussed, except for 10Broad36.

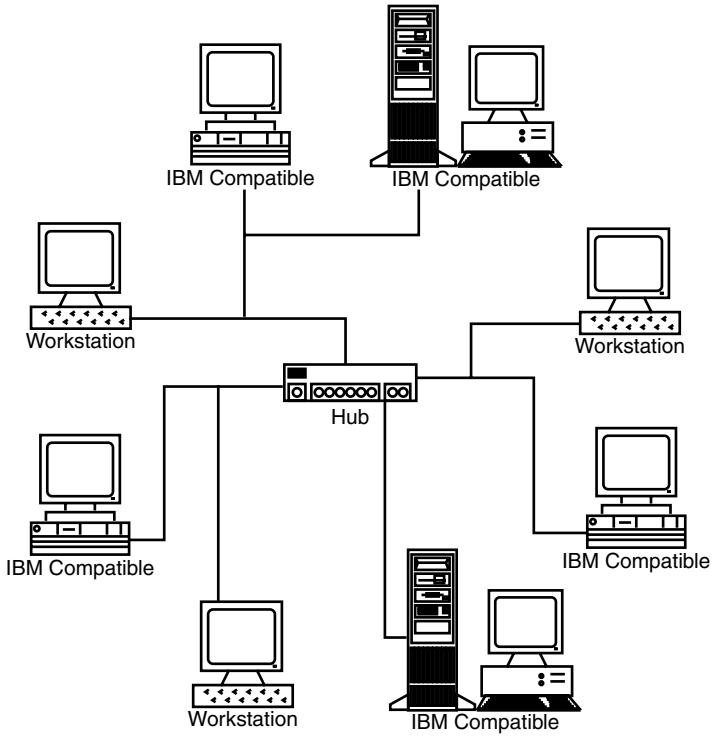


EXHIBIT 20.14 10Base-T star network.

Broadband has sufficient bandwidth to carry multiple signals across the medium. These signals can be voice, video, and data. The transmission medium is split into multiple channels, with a guard channel separating each channel. The guard channels are empty frequency space that separates the different channels to prevent interference.

Broadband cable has the advantage of being able to support transmission of signals over longer distances than the baseband coaxial cable used with 10Base5 and 10Base2. Single 10Broad36 segments can be as long as 1800 meters. 10Broad36 supports attachment of stations through transceivers that are physically and electrically attached to the broadband cable. Computers attach to the transceivers through an AUI cable as in 10Base5 installations.

When introduced, 10Broad36 offered the advantage of supporting much longer segment lengths than 10Base5 and 10Base2. But this advantage was diminished with introduction of the fiber-based services. Like 10Base2 and 10Base5, 10Broad36 is not capable of the higher network speeds, nor does it support the full-duplex mode of operation.

Fiber-Optic Inter-repeater Link

The fiber-optic inter-repeater link (FOIRL) was developed to provide a 10-Mbps point-to-point link over two fiber-optic cables. As defined in the standard, FOIRL is restricted to links between two repeaters. However, vendors have adapted the technology to also support long-distance links between a computer and a repeater.

10Base-FL

Like the Ethernet networks discussed thus far, the 10Base-FL (fiber link) supports a 10-Mbps transmission rate. It uses two fiber-optic cables to provide full-duplex transmit and receive capabilities. All 10Base-FL segments are point-to-point with one transceiver on each end of the segment. This means that it would most commonly be used to connect two router or network devices together. A computer typically attaches through an external 10Base-FL transceiver.

10Base-FL is widely used in providing network connectivity between buildings. Its ability to support longer segment lengths, and its immunity to electrical hazards such as lightning strikes and ground currents, make it ideal to prevent network damage in those situations. Fiber is also immune to the electrical noise caused by generators and other electrical equipment.

10Base-FB

Unlike 10Base-FL, which is generally used to link a router to a computer, 10Base-FB (fiber backbone) supports a 10-Mbps transmission rate over a special synchronous signaling link that is optimized for interconnecting repeaters.

While 10Base-FL can be used to link a computer to a repeater, 10Base-FB is restricted to use as a point-to-point link between repeaters. The repeaters used to terminate both ends of the 10Base-FB connection must specifically support this medium due to the unique signaling properties and method used. Consequently, one cannot terminate a 10Base-FB link on a 10Base-FL repeater; the 10Base-FL repeater does not support the 10Base-FB signaling.

10Base-FP

The 10Base-FP (fiber passive) network supports a 10-Mbps transmission rate over a fiber-optic passive star system. However, it cannot support full-duplex operations. The 10Base-FP star is a passive device, meaning that it requires no power directly, and is useful for locations where there is no direct power source available. The star unit itself can provide connectivity for up to 33 workstations. The star acts as a passive hub that receives optical signals from special 10Base-FP transceivers (and passively distributes the signal uniformly to all the other 10Base-FP transceivers connected to the star, including the one from which the transmission originated).

100Base-T

The 100Base-T identifier does not refer to a network type itself, but to a series of network types, including 100Base-TX, 100Base-FX, 100Base-T4, and 100Base-T2. These are collectively referred to as Fast Ethernet.

The 100Base-T systems generally support speeds of 10 or 100 Mbps using a process called auto-negotiation. This process allows the connected device to determine at what speed it will operate. Connections to the 100Base-T network is done through an NIC that has a built-in media-independent interface (MII), or by using an external MII much like the MAU used in the previously described networks.

100Base-TX

100Base-TX supports a 100-Mbps transmission rate over two pairs of twisted-pair cabling, using one pair of wires for transmitting data and the other pair for receiving data. The two pairs of wires are bundled into a single cable that often includes two additional pairs of wires. If present, the two additional pairs of wires must remain unused because 100Base-TX is not designed to tolerate the "crosstalk" that can occur when the cable is shared with other signals. Each end of the cable is terminated with an eight-position RJ-45 connector, or jack.

100Base-TX supports transmission over up to 100 meters of 100-ohm Category 5 unshielded twisted pair (UTP) cabling. Category 5 cabling is a higher grade wiring than the Category 3 cabling used with 10Base-T. It is rated for transmission at frequencies up to 100 MHz. The different categories of twisted pair cabling are discussed in Exhibit 20.15.

All 100Base-TX segments are point-to-point with one transceiver at each end of the cable. Most 100Base-TX connections link a computer station to a repeating hub. 100Base-TX repeating hubs typically have the transceiver function integrated internally; thus, the Category 5 cable plugs directly into an RJ-45 connector on the hub. Computer stations attach through an NIC. The transceiver function can be integrated into the NIC, allowing the Category 5 twisted-pair cable to be plugged directly into an RJ-45 connector on the NIC. Alternatively, an MII can be used to connect the cabling to the computer.

100Base-FX

100Base-FX supports a 100-Mbps transmission rate over two fiber-optic cables and supports both half- and full-duplex operation. It is essentially a fiber-based version of 100Base-TX. All of the twisted pair components are replaced with fiber components.

EXHIBIT 20.15 Twisted Pair Category Ratings

The following is a summary of the UTP cable categories:

Category 1 & Category 2: Not suitable for use with Ethernet.

Category 3: Unshielded twisted pair with 100-ohm impedance and electrical characteristics supporting transmission at frequencies up to 16 MHz. Defined by the TIA/EIA 568-A specification. May be used with 10Base-T, 100Base-T4, and 100Base-T2.

Category 4: Unshielded twisted pair with 100-ohm impedance and electrical characteristics supporting transmission at frequencies up to 20 MHz. Defined by the TIA/EIA 568-A specification. May be used with 10Base-T, 100Base-T4, and 100Base-T2.

Category 5: Unshielded twisted pair with 100 ohm impedance and electrical characteristics supporting transmission at frequencies up to 100 MHz. Defined by the TIA/EIA 568-A specification. May be used with 10Base-T, 100Base-T4, 100Base-T2, and 100Base-TX. May support 1000Base-T, but cable should be tested to make sure it meets 100Base-T specifications.

Category 5e: Category 5e (or "Enhanced Cat 5") is a new standard that will specify transmission performance that exceeds Cat 5. Like Cat 5, it consists of unshielded twisted pair with 100-ohm impedance and electrical characteristics supporting transmission at frequencies up to 100 MHz. However, it has improved specifications for NEXT (Near End Cross Talk), PSELFEXT (Power Sum Equal Level Far End Cross Talk), and Attenuation. To be defined in an update to the TIA/EIA 568-A standard. Targeted for 1000Base-T, but also supports 10Base-T, 100Base-T4, 100Base-T2, and 100Base-TX.

Category 6: Category 6 is a proposed standard that aims to support transmission at frequencies up to 250 MHz over 100-ohm twisted pair.

Category 7: Category 7 is a proposed standard that aims to support transmission at frequencies up to 600 MHz over 100-ohm twisted pair.

100Base-T4

100Base-T4 supports a 100-Mbps transmission rate over four pairs of Category 3 or better twisted-pair cabling. It allows 100-Mbps Ethernet to be carried over inexpensive Category 3 cabling, as opposed to the Category 5 cabling required by 100Base-TX.

Of the four pairs of wire used by 100Base-T4, one pair is dedicated to transmit data, one pair is dedicated to receive data, and two bi-directional pairs are used to either transmit or receive data. This scheme ensures that one dedicated pair is always available to allow collisions to be detected on the link, while the three remaining pairs are available to carry the data transfer.

100Base-T4 does not support the full-duplex mode of operation because it cannot support simultaneous transmit and receive at 100 Mbps.

1000Base-X

The identifier "1000Base-X" refers to the standards that make up Gigabit networking. These include 1000Base-LX, 1000Base-SX, 1000Base-CX, and 1000Base-T. These technologies all use a Gigabit Media-Independent Interface (GMII) that attaches the Media Access Control and Physical Layer functions of a Gigabit Ethernet device. GMII is analogous to the Attachment Unit Interface (AUI) in 10-Mbps Ethernet, and the Media-Independent Interface (MII) in 100-Mbps Ethernet. However, unlike AUI and MII, no connector is defined for GMII to allow a transceiver to be attached externally via a cable. All functions are built directly into the Gigabit Ethernet device, and the GMII mentioned previously exists only as an internal component.

1000Base-LX

This cabling format uses long-wavelength lasers to transmit data over fiber-optic cable. Both single-mode and multi-mode optical fibers (explained later) are supported. Long-wavelength lasers are more expensive than short-wavelength lasers but have the advantage of being able to drive longer distances.

1000Base-SX

This cabling format uses short-wavelength lasers to transmit data over fiber-optic cable. Only multi-mode optical fiber is supported. Short-wavelength lasers have the advantage of being less expensive than long-wavelength lasers.

1000Base-CX

This cabling format uses specially shielded balanced copper jumper cables, also called "twinax" or "short haul copper." Segment lengths are limited to only 25 meters, which restricts 1000Base-CX to connecting equipment in small areas such as wiring closets.

1000Base-T

This format supports Gigabit Ethernet over 100 meters of Category 5 balanced copper cabling. It employs full-duplex transmission over four pairs of Category 5 cabling. The aggregate data rate of 1000 Mbps is achieved by transmission at a data rate of 250 Mbps over each wire pair.

Token Ring

Token Ring is the second most widely used local area network (LAN) technology after Ethernet. Stations on a Token Ring LAN are organized in a ring topology, with data being transmitted sequentially from one ring station to the next. Circulating a token initializes the ring. To transmit data on the ring, a station must capture the token. When a station transmits information, the token is replaced with a frame that carries the information to the stations. The frame circulates the ring and can be copied by one or more destination stations. When the frame returns to the transmitting station, it is removed from the ring and a new token is transmitted.

IBM initially defined Token Ring at its research facility in Zurich, Switzerland, in the early 1980s. IBM pursued standardization of Token Ring and subsequently introduced its first Token Ring product, an adapter for the original IBM personal computer, in 1985. The initial Token Ring products operated at 4 Mbps. IBM collaborated with Texas Instruments to develop a chipset that would allow non-IBM companies to develop their own Token Ring-compatible devices. In 1989, IBM improved the speed of Token Ring by a factor of four when it introduced the first 16-Mbps Token Ring products.

In 1997, Dedicated Token Ring (DTR) was introduced that provided dedicated, or full-duplex operation. Dedicated Token Ring bypasses the normal token passing protocol to allow two stations to communicate over a point-to-point link. This doubles the transfer rate by allowing each station to concurrently transmit and receive separate data streams. This provides an overall data transfer rate of 32 Mbps. In 1998, a new 100 Mbps Token Ring product was developed that provided dedicated operation at this extended speed.

The Ring

The ring in a Token Ring network consists of the transmission medium or cabling and the ring station. While most people consider that Token Ring is a ring network-based topology, it is not. Token Ring uses a star-wired ring topology as illustrated in Exhibit 20.9.

Each station must have a Token Ring adapter card and connects to the concentrator using a lobe cable. Concentrators can be connected to other concentrators through a patch or trunk cable using the ring-in and ring-out ports on the concentrator. The concentrator itself is commonly known as a Multi-Station Access Unit (MSAU).

Each station in the ring receives its data from one neighbor, the nearest upstream neighbor, and then transmits the data to a downstream neighbor. This means that data in the Token Ring network moves sequentially from one station to another, while checking the data for errors. The station that is the intended recipient of the data copies the information as it passes. When the information reaches the originating station again, it is stripped, or removed from the ring.

A station gains the right to transmit data, commonly referred to as frames, onto the network when it detects the token passing it. The token is itself a frame that contains a unique signaling sequence that circulates on the network following each frame transfer.

Upon detecting a valid token, any station can itself modify the data contained in the token. The token data includes:

- Control and status fields
- Address fields
- Routing information fields

- Information field
- Checksum

After completing the transmission of its data, the station transmits a new token, thus allowing other stations on the ring to gain access to the ring and transmitting data of their own.

Like some Ethernet-type networks, Token Ring networks have an insertion and bypass mechanism that allows stations to enter and leave the network. When the station is in bypass mode, the lobe cable is "wrapped" back to the station, allowing it to perform diagnostic and self-tests on a single node network. In this mode, the station cannot participate in the ring to which it is connected. When the concentrators receive a "phantom drive" signal, it is inserted into the ring.

Token Ring operates at either 4 or 16 Mbps and is known as Classic Token Ring. There are Token Ring implementations that operate at higher speeds, known as Dedicated Token Ring. Today's Token Ring adapters include circuitry to allow them to detect and adjust to the current ring speed when inserting into the network.

Cabling Types

This section introduces several of the more commonly used cable types and their uses (see also Exhibit 20.16).

Twisted-Pair

Twisted-pair cabling is so named because pairs of wires are twisted around each other. Each pair of wires consists of two insulated copper wires that are twisted together. By twisting the wire pairs together, it is possible to reduce crosstalk and decrease noise on the circuit.

Unshielded Twisted-Pair Cabling (UTP)

Unshielded twisted pair cabling is in popular use today. This cable, also known as UTP, contains no shielding, and like all twisted-pair formats is graded based upon "category" level. This category level determines what the acceptable cable limits are and the implementations in which it is used.

UTP is a 100-ohm cable, with multiple pairs, but most commonly contains four pairs of wires enclosed in a common sheath. 10Base-T, 100Base-TX, and 100Base-T2 use only two of the twisted-pairs, while 100Base-T4 and 1000Base-T require all four twisted-pairs.

Screened Twisted-Pair (ScTP)

Screened twisted pair (ScTP) is four-pair 100-ohm UTP, with a single foil or braided screen surrounding all four pairs. This foil or braided screen minimizes EMI radiation and susceptibility to outside noise. This type of cable is also known as foil twisted pair (FTP), or screened UTP (sUTP). Technically, screened twisted pair is the same as unshielded twisted pair with the foil shielding. It is used in Ethernet applications in the same manner as the equivalent category of UTP cabling.

Shielded Twisted-Pair Cabling (STP)

This form of cable is technically a form of shielded twisted-pair and is the term most commonly used to describe the cabling used in Token Ring networks. Each twisted-pair is individually wrapped in a foil shield and enclosed in an overall out-braided wire shield. This level of shielding both minimizes EMI radiation and crosstalk. While this cable is not generally used with Ethernet, it can be adapted for such use with the use of "baluns" or impedance-matching transformers.

Optical Fiber

Unlike other cable systems in which the data is transmitted using an electrical signal, optical fiber uses light. This system converts the electrical signals into light, which is transmitted through a thin glass fiber, where the receiving station converts it back into electrical signals. It is used as the transmission medium for the FOIRL, 10Base-FL, 10Base-FB, 10Base-FP, 100Base-FX, 1000Base-LX, and 1000Base-SX communications standards.

Fiber-optic cabling is manufactured in three concentric layers. The central-most layer (or core) is the region where light is actually transmitted through the fiber. The "cladding" forms the second or middle layer. This layer has a lower refraction index, meaning that light does not travel through it as well as in the core. This serves to keep the light signal confined to the core. The outer layer serves to provide a "buffer" and protection for the inner two layers.

EXHIBIT 20.16 Cable Types and Properties

Standard Rate	Data Nodes per Segment	Topology	Medium	Maximum Cable Segment Length (meters)	Half-duplex	Full-duplex
10Base5	10 Mbps	100	Bus	Single 50-ohm coaxial cable (thick Ethernet) (10-mm thick)	500	n/a
10Base2	10 Mbps	30	Bus	Single 50-ohm RG 58 coaxial cable (thin Ethernet) (5-mm thick)	185	n/a
10Broad36	10 Mbps	2	Bus	Single 75-ohm CATV broadband cable	1800	n/a
FOIRL	10 Mbps	2	Star	Two optical fibers	1000	>1000
1Base5	1 Mbps		Star	Two pairs of twisted telephone cable	250	n/a
10Base-T	10 Mbps	2	Star	Two pairs of 100-ohm Category 3 or better UTP cable	100	100
10Base-FL	10 Mbps	2	Star	Two optical fibers	2000	>2000
10Base-FB	10 Mbps	2	Star	Two optical fibers	2000	n/a
10Base-FP	10 Mbps	2	Star	Two optical fibers	1000	n/a
100Base-TX	100 Mbps	2	Star	Two pairs of 100-ohm Category 5 UTP cable	100	100
100Base-FX	100 Mbps	2	Star	Two optical fibers	412	2000
100Base-T4	100 Mbps	2	Star	Four pairs of 100-ohm Category 3 or better UTP cable	100	n/a
100Base-T2	100 Mbps	2	Star	Two pairs of 100-ohm Category 3 or better UTP cable	100	100
1000Base-LX	1 Gbps	2	Star	Long-wavelength laser		
1000Base-SX	1 Gbps	2	Star	Short-wavelength laser		
1000Base-CX	1 Gbps	2	Star	Specialty shielded balanced copper jumper cable assemblies (twinax or short haul copper)	25	25
1000Base-T	1 Gbps	2	Star	Four pairs of 100-ohm Category 5 or better cable	100	100

There are two primary types of fiber-optic cable: multi-mode fiber and single-mode fiber.

Multi-Mode Fiber (MMF)

Multi-mode fiber (MMF) allows many different modes or light paths to flow through the fiber-optic path. The MMF core is relatively large, which allows for good transmission from inexpensive LED light sources.

MMF has two types: graded or stepped. Graded index fiber has a lower refraction index toward the outside of the core and progressively increases toward the center of the core. This index reduces signal dispersion in the fiber. Stepped index fiber has a uniform refraction index in the core, with a sharp decrease in the index of refraction at the core/cladding interface. Stepped index multi-mode fibers generally have lower bandwidths than graded index multi-mode fibers.

The primary advantage of multi-mode fiber over twisted-pair cabling is that it supports longer segment lengths. From a security perspective, it is much more difficult to obtain access to the information carried on the fiber than on twisted-pair cabling.

Single-Mode Fiber (SMF)

Single-mode fiber (SMF) has a small core diameter that supports only a single mode of light. This eliminates dispersion, which is the major factor in limiting bandwidth. However, the small core of a single-mode fiber makes coupling light into the fiber more difficult, and thus the use of expensive lasers as light sources is required. Laser sources are used to attain high bandwidth in SMF because LEDs emit a large range of frequencies, and thus dispersion becomes a significant problem. This makes use of SMFs in networks more expensive to implement and maintain.

SMF is capable of supporting much longer segment lengths than MMF. Segment lengths of 5000 meters and beyond are supported at all Ethernet data rates through 1 Gbps. However, SMF has the disadvantage of being significantly more expensive to deploy than MMF.

Token Ring

As mentioned, Token Ring systems were originally implemented using shielded twisted-pair cabling. It was later adapted to use the conventional unshielded twisted-pair wiring. Token Ring uses two pairs of wires to connect each workstation to the concentrator. One pair of wires is used for transmitting data and the other for receiving data.

Shielded twisted-pair cabling contains two wire pairs for the Token Ring network connection and may include additional pairs for carrying telephone transmission. This allows a Token Ring environment to use the same cabling to carry both voice and data. UTP cabling typically includes four wire pairs, of which only two are used for Token Ring.

Token Ring installations generally use a nine-pin D-shell connector as the media interface. With the adaptation of unshielded twisted-pair cabling, it is now possible to use either the D-shell or the more predominant RJ-45 data jack. Modern Token Ring cards have support for both interfaces.

Older Token Ring cards that do not have the RJ-45 jack can still be connected to the unshielded twisted-pair network through the use of an impedance matching transformer, or balun. This transformer converts from the 100-ohm impedance of the cable to the 150-ohm impedance that the card is expecting.

Cabling Vulnerabilities

There are only a few direct vulnerabilities to cabling, because this is primarily a physical medium and, as a result, direct interference or damage to the cabling is required. However, with the advent of wireless communications, it has become possible for data on the network to be eavesdropped without anyone's knowledge.

Interference

Interference occurs when a device is placed intentionally or unintentionally in a location to disrupt or interfere with the flow of electrical signals across the cable. Data flows along the cable using electrical properties and can be altered by magnetic or other electrical fields. This can result in total signal loss or in the modification of data on the cable. The modification of the data generally results in data loss.

Interference can be caused by machinery, microwave devices, and even by fluorescent light fixtures. To address situations such as these, alternate cabling routing systems (including conduit) have been deployed and

specific installations arranged to accommodate the location of the cabling. Additionally, cabling has been developed that reduces the risk of such signal loss by including a shield or metal covering to protect the cabling. Because fiber-optic cable uses light to transmit the signals, it does not suffer from this problem.

Cable Cutting

This is likely the cause of more network outages than any other. In this case, the signal path is broken as a result of physically cutting the cable. This can happen when the equipment is moved or when digging in the vicinity of the cable cuts through it. Communications companies that offer public switched services generally address this by installing network-redundant circuits when the cable is first installed. Additionally, they design their network to include fault tolerance to reduce the chance of total communications loss.

Generally, the LAN manager does not have the same concerns. His concerns focus on the protection of the desktop computers from viruses and from being handled incorrectly resulting in lost information. The LAN managers must remember that the office environment is also subject to cable cuts from accidental damage and from service or construction personnel. Failure to have a contingency and recovery plan could jeopardize their position.

Cable Damage

Damage to cables can result from normal wear and tear. The act of attaching a cable over time damages the connectors on the cable plug and the jack. The cable itself can also become damaged due to excessive bending or stretching. This can cause intermittent communications in the network, leading to unreliable communications.

Cable damage can be reduced through proper installation techniques and by regularly performing checks on exposed cabling to validate proper operation to specifications.

Eavesdropping

Eavesdropping occurs when a device is placed near the cabling to intercept the electronic signals and then reconvert them into similar signals on an external transmission medium. This provides unauthorized users with the ability to see the information without the original sender and receiver being aware of the interception. This can be easily accomplished with Ethernet and serial cables, but it is much more difficult with fiber-optic cables because the cable fibers must be exposed. Damage to the outer sheath of the fiber cables modifies their properties, producing noticeable signal loss.

Physical Attack

Most network devices are susceptible to attack from the physical side. This is why any serious network designer will take appropriate care in protecting the physical security of the devices using wiring closets, cable conduits, and other physical protection devices. It is understood that with physical access, the attacker can do almost anything. However, in most cases, the attacker does not have the luxury of time. If attackers need time to launch their attack and gain access, then they will use a logical or network-based approach.

Logical Attack

Many of these network elements are accessible via the network. Consequently, all of these devices must be appropriately configured to deny unauthorized access. Additional preventive, detective, and reactive controls must be installed to identify intrusions or attacks against these devices and report them to the appropriate monitoring agency within the organization.

Summary

In conclusion, there is much about today's networking environments for the information security specialist to understand. However, being successful in assisting the network engineers in designing a secure solution does not mean understanding all of the components of the stack, or of the physical transport method involved. It

does, however, require knowledge of what they are talking about and the differences in how the network is built with the different media options and what the inherent risks are.

However, despite the different network media and topologies available, there is a significant level of commonality between them as far as risks go. If one is not building network-level protection into the network design (i.e., network-level encryption), then it needs to be included somewhere else in the security infrastructure.

The network designer and the security professional must have a strong relationship to ensure that the concerns for data protection and integrity are maintained throughout the network.

21

Wired and Wireless Physical Layer Security Issues

James Trulove

Network security considerations normally concentrate on the higher layers of the OSI seven-layer model. However, significant issues exist in protecting physical security of the network, in addition to the routine protection of data message content that crosses the Internet. Even inside the firewall, an enterprise network may be vulnerable to unauthorized access.

Conventional wired networks are subject to being tapped by a variety of means, whether copper or fiber connections are used. In addition, methods of network snooping exist that make such eavesdropping minimally invasive, but no less significant. Wireless networking has additional characteristics that also decrease physical network security. As new technologies emerge, the potential for loss of company information through lax physical security must be carefully evaluated and steps taken to mitigate the risk.

In addition to automated security measures, such as intrusion detection and direct wiring monitoring, careful network management procedures can enhance physical security. Proper network design is critical to maintaining the desired level of security. In addition to the measures used on wired networks, wireless networks should be protected with encryption.

Wired Network Topology Basics

Everyone involved with local area networking has a basic understanding of network wiring and cabling. Modern LANs are almost exclusively Ethernet hub-and-spoke topologies (also called star topologies). Individual cable runs are made from centralized active hubs to each workstation, network printer, server, or router. At today's level of technology, these active hubs may perform additional functions, including switching, VLAN (virtual LAN) filtering, and simple layer 3 routing. In some cases, relatively innocuous decisions in configuring and interconnecting these devices can make a world of difference in a network's physical security.

An illustration of network topology elements is shown in Exhibit 21.1. The exhibit shows the typical user-to-hub and hub-to-hub connections, as well as the presence of switching hubs in the core of the network. Three VLANs are shown that can theoretically separate users in different departments. The general purpose of a VLAN is to isolate groups of users so they cannot access certain applications or see each other's data. VLANs are inherently difficult to diagram and consequently introduce a somewhat unwelcome complexity in dealing with physical layer security. Typically, a stand-alone router is used to interconnect data paths between the VLANs and to connect to the outside world, including the Internet, through a firewall. A so-called layer 3 switch could actually perform the non-WAN functions of this router, but some sort of WAN router would still be needed to make off-site data connections, such as to the Internet.

This chapter discusses the physical layer security issues of each component in this network design as well as the physical security of the actual interconnecting wiring links between the devices.

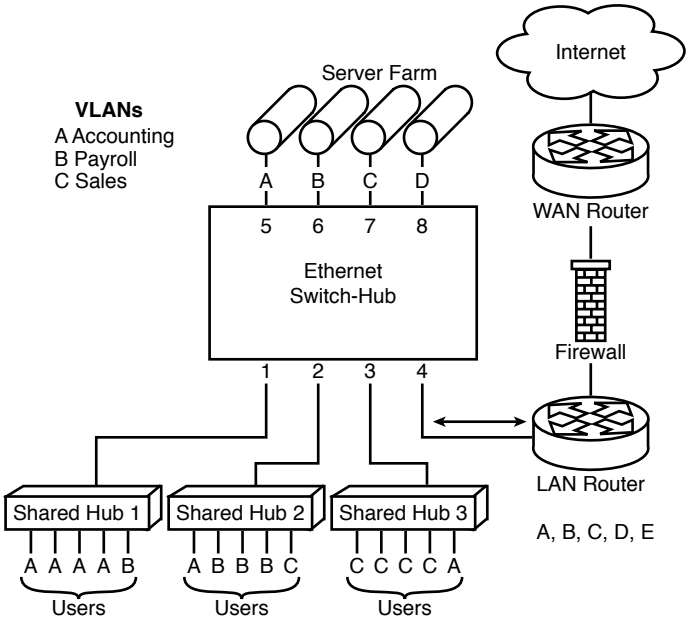


Exhibit 21.1 Topology of a network with shared, switched, and routed connections.

Shared Hubs

The original concept of the Ethernet network topology was that of a shared coaxial media with periodic taps for the connection of workstations. Each length of this media was called a segment and was potentially interconnected to other segments with a repeater or a bridge. Stations on a segment listened for absence of signal before beginning a transmission and then monitored the media for indication of a collision (two stations transmitting at about the same time). This single segment (or group of segments linked by repeaters) is considered a collision domain, as a collision anywhere in the domain affects the entire domain. Unfortunately, virtually any defect in the main coax or in any of the connecting transceivers, cables, connectors, or network interface cards (NICs) would disrupt the entire segment.

One way to minimize the effects of a single defect failure is to increase the number of repeaters or bridges. The shared hub can decrease the network failures that are a result of physical cable faults. In the coaxial-Ethernet world, these shared hubs were called multiport repeaters, which closely described their function. Additional link protection was provided by the evolution to twisted-pair Ethernet, commonly known as 10BaseT. This link topology recognizes defective connections and dutifully isolates the offending link from the rest of the hub, which consequently protects the rest of the collision domain. The same type of shared network environment is available to 10BaseF; 100BaseT, FX, and SX (Fast Ethernet); and 1000BaseT, TX, FX, SX (Gigabit Ethernet).

Shared hubs, unfortunately, are essentially a party line for data exchange. Privacy is assured only by the courtesy and cooperation of the other stations in the shared network. Data packets are sent out on the shared network with a destination and source address, and the protocol custom dictates that each workstation node "listens" only to those packets that have its supposedly unique address as the destination. Conversely, a courteous workstation would listen exclusively to traffic addressed to itself and would submit a data packet only to the shared network with its own uniquely assigned address as the source address. Right!?

In practice, it is possible to connect sophisticated network monitoring devices, generically called network sniffers to any shared network and see each and every packet transmitted. These monitoring devices are very expensive (U.S.$10,000 to $25,000) and high-performance, specialized test equipment, which would

theoretically limit intrusion into networks. However, much lower-performance, less-sophisticated packet-snooping software is readily available and can run on any workstation (including PDAs). This greatly complicates the physical security problem, as any connected network device, whether authorized or not, can snoop virtually all of the traffic on a shared LAN.

In addition to the brute-force sniffing devices, a workstation may simply attempt to access network resources for which it has no inherent authorization. For example, in many types of network operating system (NOS) environments, one may easily access network resources that are available to any authorized user. Microsoft's security shortcomings are well documented, from password profiles to NetBIOS and from active control structures to the infamous e-mail and browser problems. A number of programs are available to assist the casual intruder in unauthorized information mining.

In a shared hub environment, physical layer security must be concerned with limiting physical access to workstations that are connected to network resources. For the most part, these workstation considerations are limited to the use of boot-up, screen saver, and log-in passwords; the physical securing of computer equipment; and the physical media security described later. Most computer boot routines, network logins, and screen savers provide a method of limiting access and protecting the workstation when not in use. These password schemes should be individualized and changed often.

Procedures for adding workstations to the network and for interconnecting hubs to other network devices should be well documented and their implementation limited to staff members with appropriate authorization. Adds, moves, and changes should also be well documented. In addition, the physical network connections and wiring should be periodically audited by an outside organization to ensure the integrity of the network. This audit can be supplemented by network tools and scripts that self-police workstations to determine that all of the connected devices are known, authorized, and free of inappropriate software that might be used to intrude within the network.

Switched Hubs Extend Physical Security

The basic security fault of a shared network is the fact that all packets that traverse the network are accessible to all workstations within the collision domain. In practice, this may include hundreds of workstations. A simple change to a specialized type of hub, called a switched hub, can provide an additional measure of security, in addition to effectively multiplying data throughput of the hub.

A switched hub is an OSI layer 2 device, which inspects the destination media access layer (MAC) address of a packet and selectively repeats the packet only to the appropriate switch port segment on which that MAC address device resides. In other words, if a packet comes in from any port, destined for a known MAC address X_1 on port 3, that packet would be switched directly to port 3, and would not appear on any other outbound port. This is illustrated in Exhibit 21.2. The switch essentially is a multi-port layer 2 bridge that learns the relative locations of all MAC addresses of devices that are attached and forms a temporary path to the appropriate destination port (based on the destination MAC address) for each packet that is processed. This processing is normally accomplished at "wire speed." Simultaneous connection paths may be present between sets of ports, thus increasing the effective throughput beyond the shared hub.

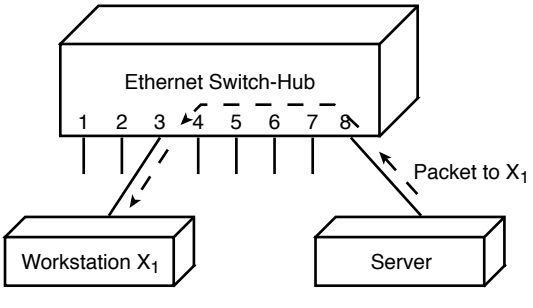


Exhibit 21.2 Switched Ethernet hub operation.

Switched hubs are often used as simple physical security devices, because they isolate the ports that are not involved in a packet transmission. This type of security is good if the entire network uses switched connections. However, switched hubs are still more expensive than shared hubs, and many networks are implemented using the switch-to-shared hub topology illustrated in Exhibit 21.1. While this may still provide a measure of isolation between groups of users and between certain network resources, it certainly allows any user on a shared hub to view all the packets to any other user on that hub.

Legitimate testing and monitoring on a switched hub is much more difficult than on a shared hub. A sniffing device connected to port 7 (Exhibit 21.2), for example, could not see the packet sent from port 8 to port 3! The sniffer would have its own MAC address, which the switch would recognize, and none of the packets between these two other nodes would be sent. To alleviate this problem somewhat, a feature called port mirroring is available on some switches. Port mirroring can enable a user to temporarily create a shared-style listening port on the switch that duplicates all the traffic on a selected port. Alternatively, one could temporarily insert a shared hub on port 3 or port 8 to see each port's respective traffic. An inadvertent mirror to a port that is part of a shared-hub network can pose a security risk to the network. This is particularly serious if the mirrored port happens to be used for a server or a router connection, because these devices see data from many users.

To minimize the security risk in a switched network, it is advisable to use port mirroring only as a temporary troubleshooting technique and regularly monitor the operation of switched hubs to disable any port mirroring. In mixed shared/switched networks, layer 2 VLANs may offer some relief (the cautions of the next section notwithstanding). It may also be possible to physically restrict users to hubs that are exclusively used by the same department, thus minimizing anyone's ability to snoop on other departments' data. This assumes that each department-level shared hub has an uplink to a switched hub, perhaps with VLAN segregation.

In addition, administrators should tightly manage the passwords and access to the switch management interface. One of the most insidious breaches in network security is the failure to modify default passwords and to systematically update control passwords on a regular basis.

VLANS Offer Deceptive Security

One of the most often used network capabilities for enhancing security is the virtual LAN (VLAN) architecture. VLANs can be implemented at either layer 2 or layer 3.

A layer 2 VLAN consists of a list of MAC addresses that are allowed to exchange data and is rather difficult to administer. An alternative style of layer 1/layer 2 VLAN assigns physical ports of the switch to different VLANs. The only caveat here is that all of the devices connected to a particular switch port are restricted to that VLAN. Thus, all of the users of shared hub 1 (Exhibit 21.1) would be assigned to switch hub port 1's VLAN. This may be an advantage in many network designs and can actually enhance security.

Here is the deception for layer 2. A layer 2 VLAN fails to isolate packets from all of the other users in either a hierarchical (stacked) switch network or in a hybrid shared/switched network. In the hybrid network, all VLANs may exist on any shared hub, as shown in Exhibit 21.3. Therefore, any user on shared hub 2 can snoop

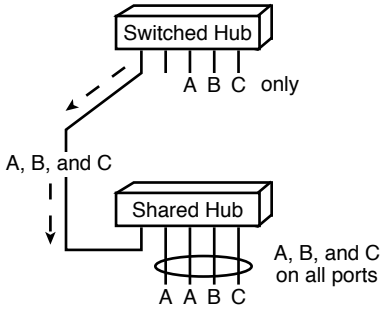


Exhibit 21.3 VLANs A, B, and C behavior across both switched and shared Ethernet hubs.

on any traffic on that hub, regardless of VLAN. In a port-based layer 2 VLAN, the administrator must be certain that all users that are connected to each port of the VLAN are entitled to see any of the data that passes to or from that port. Sadly, the only way to do that is to connect every user to his own switch port, which takes away the convenience of the VLAN and additionally adds layers of complexity to setup. A MAC-based VLAN can still allow others to snoop packets on shared hubs or on mirrored switch hubs.

A layer 3 VLAN is really a higher-level protocol subnet. In addition to the MAC address, packets that bear Internet Protocol (IP) data possess a source and destination address. A subset of IP addresses, called a subnet, consists of a contiguous range of addresses. Typically, IP devices recognize subnets through a base address and a subnet mask that "sizes" the address range of the subnet. The IP protocol stack screens out all data interchanges that do not bear addresses within the same subnet. A layer 3 router allows connection between subnets. Technically, then, two devices must have IP addresses in the same subnet to "talk," or they must connect through a router (or series of routers) that recognizes both subnets.

The problem is that IP data packets of different subnets may coexist within any collision domain — that is, on the same shared hub or switched link. The TCP/IP protocol stack simply ignores any packet that is not addressed to the local device. As long as everybody is a good neighbor, packets go only where they are intended. Right?

In reality, any sniffer or snooping program on any workstation can see all data traffic that is present within its collision domain, regardless of IP address. The same was true of non-IP traffic, as was established previously. This means that protecting data transmission by putting devices in different subnets is a joke, unless care is taken to limit physical access to the resources so that no unauthorized station can snoop the traffic.

VLAN/Subnets Plus Switching

A significant measure of security can be provided within a totally switched network with VLANs and subnets. In fact, this is exactly the scheme that is used in many core networks to restrict traffic and resources to specific, protected paths. For the case of direct access to a data connection, physical security of the site is the only area of risk. As long as the physical connections are limited to authorized devices, port mirroring is off, and no remote snooping (often called Trojan horse) programs are running surreptitiously and firewalling measures are effective, then the protected network will be reasonably secure, from the physical layer standpoint.

Reducing the risk of unauthorized access is very dependent on physical security. Wiring physical security is another issue that is quite important, as is shown in the following section.

Wiring Physical Security

Physical wiring security has essentially three aspects: authorized connections, incidental signal radiation, and physical integrity of connections. The first requirement is to inspect existing cabling and verify that every connection to the network goes to a known location. Organized, systematic marking of every station cable, patch cord, patch panel, and hub is a must to ensure that all connections to the network are known and authorized.

Where does every cable go? Is that connection actually needed? When moves are made, are the old data connections disabled? Nothing could be worse than having extra data jacks in unoccupied locations that are still connected to the network. The EIA/TIA 569 A *Commercial Building Standard for Telecommunications Pathways and Spaces* and EIA/TIA 606 *The Administration Standard for the Telecommunications Infrastructure of Commercial Buildings* give extensive guidelines for locating, sizing, and marking network wiring and spaces.

In addition, the cable performance measurements that are recommended by ANSI/TIA/EIA-568-B *Commercial Building Telecommunications Cabling Standard* should be kept on file and periodically repeated. The reason is simple. Most of the techniques that could be used to tap into a data path will drastically change the performance graph of a cable run. For example, an innocuous shared hub could be inserted into a cable path, perhaps hidden in a wall or ceiling, to listen in to a data link. However, this action would change the reported cable length, as well as other parameters reported by a cable scanner.

Network cabling consists of two types: four-pair copper cables and one-pair fiber-optic cables. Both are subject to clandestine monitoring. Copper cabling presents the greater risk, as no physical connection may be

required. As is well known, high-speed data networking sends electrical signals along two or more twisted pairs of insulated copper wire. A 10BaseT Ethernet connection has a fundamental at 10 MHz and signal components above that. A 100BaseT Fast Ethernet connection uses an encoding technique to keep most of the signal component frequencies below 100 MHz. Both generate electromagnetic fields, although most of the field stays between the two conductors of the wire pair. However, a certain amount of energy is actually radiated into the space surrounding the cable.

The major regulatory concern with this type of cabling is that this radiated signal should be small so it does not interfere with conventional radio reception. However, that does not mean that it cannot be received! In fact, one can pick up the electromagnetic signals from Category 3 cabling anywhere in proximity to the cable. Category 5 and above cabling is better only by degree. Otherwise, the cable acts like an electronic leaky hose, spewing tiny amounts of signal all along its length.

A sensor can be placed anywhere along the cable run to pick up the data signal. In practice, it is (fortunately) a little more difficult than this, simply because this would be a very sophisticated technique and because access, power, and an appropriate listening point would also be required. In addition, bidirectional (full-duplex) transmission masks the data in both directions, as do multiple cables. This probably presents less of a threat to the average data network than direct physical connection, but the possibility should not be ignored.

Fiber cable tapping is a much subtler problem. Unlike that on its copper equivalent, the signal is in the form of light and is carried within a glass fiber. However, there are means to tap into the signal if one has access to the bare fiber or to interconnect points. It is true that most of the light passes longitudinally down the glass fiber. However, a tiny amount may be available through the sidewall of the fiber, if one has the means to detect it. Presumably, this light leakage would be more evident in a multi-mode fiber, where the light is not restricted to so narrow a core as with single-mode fiber. In addition, anyone with access to one of the many interconnection points of a fiber run could tap the link and monitor the data.

Fiber-optic cable runs consist of patch and horizontal fiber cable pairs that are connectorized at the patch panel and at each leg of the horizontal run. Each connectorized cable segment is interconnected to the next leg by a passive coupler (also called an adapter). For example, a typical fiber link is run through the wall to the workstation outlet. The two fibers are usually terminated in an ordinary fiber connector, such as an SC or one of the new small-form factor connectors. The pair of connectors is then inserted into the inside portion of the fiber adapter in the wall plate, and the plate is attached to the outlet box. A user cable or patch cord is then plugged into the outside portion of the same fiber adapter to connect the equipment. If some person were to have access to removing the outlet plate, it would take a few seconds to insert a device to tap into the fiber line, since it is conveniently connectorized with a standard connector, such as the SC connector.

Modern progress has lessened this potential risk somewhat, as some of the new small-form factor connector systems use an incompatible type of fiber termination in the wall plate. However, this could certainly be overcome with a little ingenuity.

Most of the techniques that involve a direct connection or tap into a network cable require that the cable's connection be temporarily interrupted. Cable-monitoring equipment is available that can detect any momentary break in a cable, to make the reconnection of a cable through an unauthorized hub, or to make a new connection into the network. This full-time cable-monitoring equipment can report and log all occurrences, so that an administrator can be alerted to any unusual activities on the cabling system.

Security breaches happen and, indeed, should be anticipated. An intrusion detection system should be employed inside the firewall to guard against external and internal security problems. It may be the most effective means of detecting unauthorized access to an internal network. An intrusion detection capability can include physical layer alarms and reporting, in addition to the monitoring of higher layers of protocol.

Wireless Physical Layer Security

Wireless networking devices, by their very nature, purposely send radio signals out into the surrounding area. Of course, it is assumed that only the authorized device receives the wireless signal, but it is impossible to limit potential eavesdropping. Network addressing and wireless network "naming" cannot really help, although they are effective in keeping the casual user out of a wireless network.

The only technique that can ensure that someone cannot easily monitor wireless data transmissions is data encryption. Many of the wireless LAN devices on the market now offer Wired Equivalent Privacy (WEP) as a standard feature. This is a 64-bit encryption standard that uses manual key exchange to privatize the signal between a wireless network interface card (WNIC) and an access point bridge (which connects to the wired

network). As the name implies, this is not expected to be a high level of security; it is expected only to give one approximately the same level of privacy that would exist if the connection were made over a LAN cable.

Some WNICs use a longer encryption algorithm, such as 128-bit encryption, that may provide an additional measure of security. However, there is an administration issue with these encryption systems, and keys must be scrupulously maintained to ensure integrity of the presumed level of privacy.

Wireless WAN connections, such as the popular cellular-radio systems, present another potential security problem. At the present time, few of these systems use any effective encryption whatsoever and thus are accessible to anyone with enough reception and decoding equipment. Strong-encryption levels of SSL should certainly be used with any private or proprietary communications over these systems.

Conclusion

A complete program of network security should include considerations for the physical layer of the network. Proper network design is essential in creating a strong basis for physical security. The network practices should include the use of switching hubs and careful planning of data paths to avoid unnecessary exposure of sensitive data. The network manager should ensure that accurate network cabling system records are maintained and updated constantly to document authorized access and to reflect all moves and adds. Active network and cable monitoring may be installed to enhance security. Network cable should be periodically inspected to ensure integrity and authorization of all connections. Links should be rescanned periodically and discrepancies investigated. Wireless LAN connections should be encrypted at least to WEP standards, and strong encryption should be considered. Finally, the information security officer should consider the value of periodic security audits at all layers to cross-check the internal security monitoring efforts.

22

Network Router Security

Steven F. Blanding

Routers are a critical component in the operation of a data communications network. This chapter describes network router capabilities and the security features available to manage the network. Routers are used in local area networks, wide area networks, and for external connections, either to service providers or to the Internet.

Router Hardware and Software Components

Routers contain a core set of hardware and software components, although the router itself provides different capabilities and has different interfaces. The core hardware components include the central processing unit (CPU), random access memory (RAM), nonvolatile RAM, read-only memory (ROM), flash memory, and input/output (I/O) ports. These are outlined in Exhibit 22.1. While these components may be configured differently, depending on the type of router, they remain critical to the proper overall operation of the device and support for the router's security features.

- *Central processing unit*. Typically known as a critical component in PCs and larger computer systems, the CPU is also a critical component found in network routers. The CPU, or microprocessor, is directly related to the processing power of the router, executing instructions that make up the router's operating system (OS). User commands entered via the console or Telnet connection are also handled by the CPU.
- *Random access memory*. RAM is used within the router to perform a number of different functions. RAM is also used to perform packet buffering, provide memory for the router's configuration file (when the device is operational), hold routing tables, and provide an area for the queuing of packets when they cannot be directly output due to traffic congestion at the common interface. During operation, RAM provides space for caching Address Resolution Protocol (ARP) information that enhances the transmission capability of local area networks connected to the router.
- *Nonvolatile RAM*. When the router is powered off, the contents of RAM are cleared. Nonvolatile RAM (NVRAM) retains its contents when the router is powered off. Recovery from power failures is performed much more quickly where a copy of the router's configuration file is stored in NVRAM. As a result, the need to maintain a separate hard disk or floppy device to store the configuration file is eliminated. The wear-and-tear or moving components such as hard drives is the primary source of router hardware failures. As a result, the absence of these moving components provides for a much longer life span.
- *Read-only memory*. Code contained on read-only memory (ROM) chips on the system board in routers performs power-on diagnostics. This function is similar to the power-on self-test that PCs perform. In network routers, OS software is also loaded by a bootstrap program in ROM. Software upgrades are performed by removing and replacing ROM chips on some types of routers, while others may use different techniques to store and manage the operating system.
- *Flash memory*. An erasable and reprogrammable type of ROM is referred to as flash memory. The router's microcode and an image of the OS can be held in flash memory on most routers. The cost of

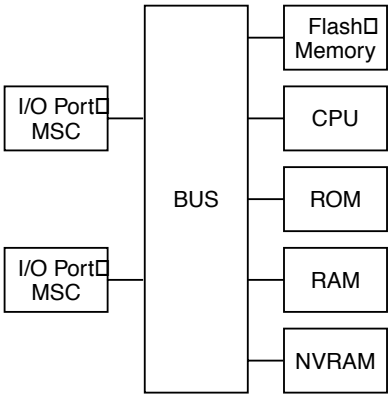


EXHIBIT 22.1 Basic router hardware components.

flash memory can easily be absorbed through savings achieved on chip upgrades over time because it can be updated without having to remove and replace chips. Depending on the memory capacity, more than one OS image can be stored in flash memory. A router's flash memory can also be used to Trival File Transfer Protocol (TFTP) an OS image to another router.

- *Input/output ports.* The connection through which packets enter and exit a router is the I/O port. Media-specific converters, which provide the physical interface to specific types of media, are connected to each I/O port. The types of media include Ethernet LAN, Token Ring LAN, RS-232, and V.35 WAN. As data packets pass through the ports and converters, each packet must be processed by the CPU to consult the routing table and determine where to send the packet. This process is called process switching mode. Layer 2 headers are removed as the packet is moved into RAM as data is received from the LAN. The packet's output port and manner of encapsulation are determined by this process.

A variation of process switching mode is called fast switching, in which the router maintains a memory cache containing information about destination IP addresses and next-hop interfaces. In fast switching, the router builds the cache by saving information previously obtained from the routing table. In this scheme, the first packet to a specific destination causes the CPU to consult the routing table. After information is obtained regarding the next-hop interface for that particular destination and that information is inserted into the fast switching cache, the routing table is no longer consulted for new packets sent to this destination. As a result, a substantial reduction in the load on the router's CPU occurs and the router's capacity to switch packets takes place at a much faster rate. Some of the higher-end router models are special hardware features that allow for advanced variations of fast switching. Regardless of the type of router, cache is used to capture and store the destination address to interface mapping. Some advanced-feature routers also capture the source IP address and the upper layer TCP ports. This type of switching mode is called netflow switching.

Initializing Routers

The router executes a series of predefined operations when the device is powered on. Depending on the previous configuration of the router, additional operations can be performed. These operations contribute to the stability of the router, and are necessary to its proper and secure performance.

The first function performed by the router is a series of diagnostic tests called power-on tests or POST. These tests validate the operation of the router's processor, memory, and interface circuitry. This function, as well as all of the other major functions performed during power-on time, is illustrated in Exhibit 22.2.

According to the flowchart, upon completion of the POST process, the bootstrap loader is to initialize the operating system (OS) into main memory. The first step in this process is to determine the location of the OS image by checking the router's configuration register. The image could be located in either ROM, flash memory, or possibly on the network. The register settings not only indicate the location of the OS, but they also define other key functions, including whether the console terminal displays diagnostic messages and how the router reacts to the entry to the entry of a break key on the console keyboard. Typically, the configuration register is a 16-bit value with the last four bits indicating the boot field. The location of the router's configuration file is

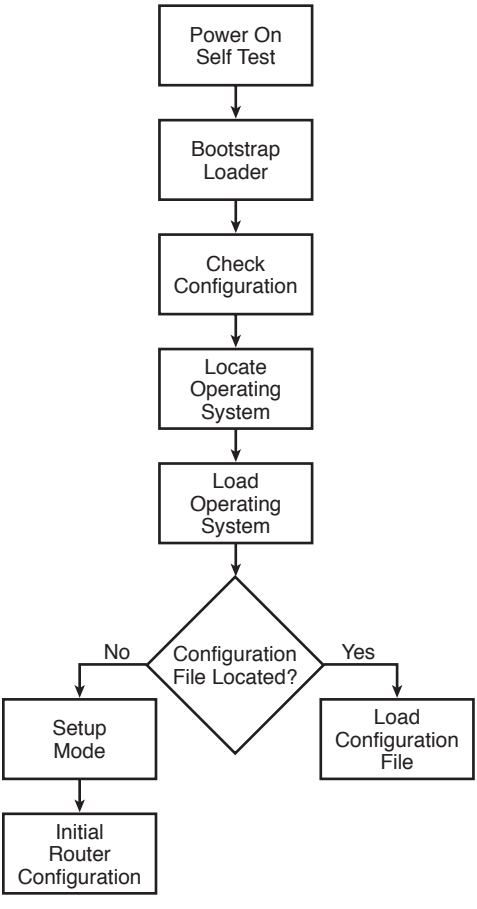


EXHIBIT 22.2 Router initialization.

identified by the boot field. The router will search the configuration file for boot commands if the boot register is set to 2, which is the most common setting. The router will load the OS image from flash memory if this setting is not found. The router will send a TFTP request to the broadcast address requesting an OS image if no image exists in flash memory. The image will then be loaded from the TFTP server.

The bootstrap loader loads the OS image into the router's RAM once the configuration register process is complete. With the OS image now loaded, NVRAM is examined by the bootstrap loader to determine if a previous version of the configuration file had been saved. This file is then loaded into RAM and executed, at which point the router becomes operational. If the file is not stored in NVRAM, a Setup dialog is established by the operating system. The Setup dialog is a predefined sequence of questions posed to the console operator that must be completed to establish the configuration information that is then stored in NVRAM.

During subsequent initialization procedures, this version of the configuration file will be copied from NVRAM and loaded into RAM. To bypass the contents of the configuration file during password recovery of the router, the configuration register can be instructed to ignore the contents of NVRAM.

Operating System Image

As mentioned, the bootstrap loader locates the OS image based on the setting of the configuration register. The OS image consists of several routines that perform the following functions:

- Executing user commands
- Supporting different network functions

- Updating routing tables
- Supporting data transfer through the router, including managing buffer space

The OS image is stored in low-address memory.

Configuration File

The role of the configuration file was discussed briefly in the router initialization process. The router administrator is responsible for establishing this file, which contains information interpreted by the OS. The configuration file is a key software component responsible for performing different functions built into the OS. One of the most important functions is the definition of access lists and how they are applied by the OS to different interfaces. This is a critical security control function that establishes the degree of control concerning packet flow through the router. In other words, the OS interprets and executes the access control list statements stored in the configuration file to establish security control. The configuration file is stored in the upper-address memory of the NVRAM when the console operator saves it. The OS then accesses it, which is stored in the lower-address memory of NVRAM.

Controlling Router Data Flow

Understanding how the router controls data flow is key to the overall operation of this network device. The information stored in the configuration file determines how the data will flow through the router.

To begin, the types of frames to be processed are determined at the media interface — either Ethernet, Token Ring, FDDI, etc. — by previously entered configuration commands. These commands consist of one or more operating rates and other parameters that fully define the interface. The router verifies the frame format of arriving data and develops frames for output after it knows the type of interface it must support. The frames for output could be formed via that interface or through a different interface. An important control feature provided by the router is its ability to use an appropriate cyclic redundancy check (CRC). The CRC feature checks data integrity on received frames because the interface is known to the router. The appropriate CRC is also computed and appended to frames placed onto media by the router.

The method by which routing table entries occur is controlled by configuration commands within NVRAM. These entries include static routing, traffic prioritization routing, address association, and packet destination interface routing. When static routing is configured, the router does not exchange routing table entries with other routers. Prioritization routing allows data to flow into one or more priority queues where higher-priority packets pass ahead of lower-priority packets. The area within memory that stores associations between IP addresses and their corresponding MAC layer 2 addresses is represented by ARP cache. The destination interfaces through which the packet will be routed are also defined by entries in the routing table.

As data flows into a router, several decision operations take place. For example, if the data packet destination is a LAN and address resolution is required, the router will use the ARP cache to determine the MAC delivery address and outgoing frame definition. The router will form and issue an ARP packet to determine the necessary layer 2 address if the appropriate address is not in cache. The packet is ready for delivery to an outgoing interface port once the destination address and method of encapsulation are determined. Depending on priority definitions, the packet could be placed into a priority queue prior to delivery into the transmit buffer.

Configuring Routers

Before addressing the security management areas associated with routers, the router configuration process must first be understood. This process includes a basic understanding of setup considerations, the Command Interpreter, the user mode of operation, the privileged mode of operation, and various types of configuration commands. Once these areas are understood, the access security list and the password control functions of security management are described.

Router Setup Facility

The router setup facility is used to assign the name to the router and to assign both a direct connect and virtual terminal password. The operator is prompted to accept the configuration once the setup is complete. During

the setup configuration process, the operator must be prepared to enter several specific parameters for each protocol and interface. In preparation, the operator must be familiar with the types of interfaces installed and the list of protocols that can be used.

The router setup command can be used to not only review previously established configuration entries, but also to modify them. For example, the operator could modify the enable password using the enable command. The enable password must be specified by the operator upon entering the enable command on the router console port. This command allows access to privileged execute commands that alter a router's operating environment. Another password, called the enable secret password, can also be used to provide access security. This password serves the same purpose as the enable password; however, the enable secret password is encrypted in the configuration file. As a result, only the encrypted version of the enable secret password is available when the configuration is displayed on the console. Therefore, the enable secret password cannot be disclosed by obtaining a copy of the router configuration. To encrypt the enable password — as well as the virtual terminal, auxiliary, and console ports — the service password-encryption command can be used. This encryption technique is not very powerful and can be easily compromised through commonly available password-cracking software. As a result, the enable secret password should be used to provide adequate security to the configuration file.

Command Interpreter

The command interpreter is used by the router to interpret router commands entered by the operator. The interpreter checks the command syntax and executes the operation requested. To obtain access to the command interpreter, the operator must log on to the router using the correct password, which was established during the setup process. There are two separate command interpreter levels or access levels available to the operator. These are referred to as user and privileged commands, each of which is equipped with a separate password.

- *User mode of operation*. The user mode of operation is obtained by simply logging into the router. This level of access allows the operator to perform such functions as displaying open connections, changing the terminal parameters, establishing a logical connection name, and connecting to another host. These are all considered noncritical functions.
- *Privileged mode of operation*. The privileged commands are used to execute sensitive, critical operations. For example, the privileged command interpreter allows the operator to lock the terminal, turn privileged commands off or on, and enter configuration information. Exhibit 22.3 contains a list of some of the privileged mode commands. All commands available to the user mode are also available to the privileged mode. User mode commands are not included in the list.

The privileged mode of operation must be used to configure the router. A password is not required the first time one enters this mode. The enable-password command would then be used to assign a password for subsequent access to privileged mode.

EXHIBIT 22.3 Prigileged Mode Commands

Command	Function
Clear	Reset functions
Configure	Enter configuration mode
Connect	Open a terminal connection
Disable	Turn off privileged commands
Erase	Erase flash or configuration memory
Lock	Lock the terminal
Reload	Halt and perform cold restart
Setup	Run the SETUP command facility
Telnet	Open a telnet session
Tunnel	Open a tunnel connection
Write	Write running configuration to memory

Configuration Commands

Configuration commands are used to configure the router. These commands are grouped into four general categories: global, interface, line, and router subcommands. Exhibit 22.4 contains a list of router configuration commands.

Global configuration commands define systemwide parameters, to include access lists. Interface commands define the characteristics of a LAN or WAN interface and are preceded by an interface command. These commands are used to assign a network to a particular port and configure specific parameters required for the interface. Line commands are used to modify the operation of a serial terminal line. Finally, router subcommands are used to configure IP routing protocol parameters and follow the use of the router command.

Router Access Control

As mentioned previously, access control to the router and to the use of privileged commands is established through the use of passwords. These commands are included in Exhibit 22.5.

Router Access Lists

The use of router access lists plays a key role in the administration of access security control. One of the most critical security features of routers is the capability to control the flow of data packets within the network. This feature is called packet filtering, which allows for the control of data flow in the network based on source and destination IP addresses and the type of application used. This filtering is performed through the use of access lists.

An ordered list of statements permitting or denying data packets to flow through a router based on matching criteria contained in the packet is defined as an access list. Two important aspects of access lists are the sequence or order of access list statements and the use of an implicit deny statement at the end of the access list. Statements

EXHIBIT 22.4 Router Configuration Commands

Command	Use
Write terminal	Display the current configuration in RAM
Write network	Share the current configuration in RAM with a network server via TFTP
Write erase	Erase the contents of NVRAM
Configure network	Load a previously created configuration from a network server
Configure memory	Load a previously created configuration from NVRAM
Configure terminal	Configure router manually from the console

EXHIBIT 22.5 Router Access Control Commands

Command	Function
Enable password	Privileged EXE mode access is established with this password
Enable secret	Enable secret access using MD5 encryption is established with this password
Line console 0	Console terminal access is established with this password
Line vty 0 4	Telnet connection access is established with this password
Service password encryption	When using the Display command, this command protects the display of the password

must be entered in the correct sequence in the access list for the filtering to operate correctly. Also, explicit permit statements must be used to ensure that data is not rejected by the implicit deny statement. A packet that is not explicitly permitted will be rejected by the implicit "deny all" statement at the end of the access list.

Routers can be programmed to perform packet filtering to address many different kinds of security issues. For example, packet filtering can be used to prevent Telnet session packets from entering the network originating from specified address ranges. The criteria used to permit or deny packets depend on the information contained within the packet's layer 3 or layer 4 header. While access lists cannot use information above layer 4 to filter packets, context-based access control (CBAC) can be used. CBAC provides for filtering capability at the application layer.

Administrative Domains

An administrative domain is a general grouping of network devices such as workstations, servers, network links, and routers that are maintained by a single administrative group. Routers are used as a boundary between administrative domains. Each administrative domain typically has its own security policy and, as a result, there is limited access between data networks in separate domains. Most organizations would typically need only one administrative domain; however, separate domains can be created if different security policies are required.

While routers are used as boundaries between domains, they also serve to connect separate administrative domains. Routers can be used to connect two or more administrative domains of corporate networks or to connect the corporate administrative domain to the Internet. Because all data packets must flow through the router and because routers must be used to connect separate geographic sites, packet-filtering functionality can be provided by the router without the need for additional equipment or software. All of the functionality for establishing an adequate security policy with sophisticated complex security can be provided by network routers.

The operating system used by Cisco Corporation to create security policies as well as all other router functions is called the internetwork operating system (IOS). The commands entered by the console operator interface with the IOS. These commands are used by the IOS to manage the router's configuration, to control system hardware such as memory and interfaces, and to execute system tasks such as moving packets and building dynamic information like routing and ARP tables. In addition, the IOS has many of the same features as other operating systems such as Windows, Linux, and UNIX.

Access lists also provide functions other than packet filtering. These functions include router access control, router update filtering, packet queuing, and dial-on-demand control. Access lists are used to control access to the router through mechanisms such as SNMP and Telnet. Access lists can also be used to prevent a network from being known to routing protocols through router update filtering. Classes of packets can be given priority over other classes of packets by using access lists to specify these packet types to different outgoing queues. Finally, access lists can be used to trigger a dial connection to occur by defining packets to permit this function.

Packet Filtering

As described previously, a primary function performed by access lists is packet filtering. Filtering is an important function in securing many networks. Many devices can be used to implement packet filters. Packet filtering is also a common feature within firewalls where network security exists to control access between internal trusted systems and external, untrusted systems. The specification of which packets are permitted access through a router and which packets are denied access through a router, as determined by the information contained within the packet, is called a packet filter.

Packet filters allow administrators to specify certain criteria that a packet must meet in order to be permitted through a router. If the designated criteria are not met, the packet is denied. If the packet is not explicitly denied or permitted, then the packet will be denied by default. This is called an implicit deny, which is a common and important security feature used in the industry today. As mentioned, the implicit deny, although it operates by default, can be overridden by explicit permits. Other security features available through packet filtering are subject to limitations. These limitations include stateless packet inspection, information examination limitations, and IP address spoofing.

Stateless Packet Inspection

Access control lists cannot determine if a packet is part of a TCP/UDP conversation because each packet is examined as if it is a stand-alone entity. No mechanism exists to determine that an inbound TCP packet with the ACK bit set is actually part of an existing conversation. This is called stateless packet filtering (e.g., the router does not maintain information on the status or state of existing conversations). Stateless packet inspection is performed by non-context-based access control lists.

State tables are used to record the source and destination addresses and ports from which the router places the entries. While incoming packets are checked to ensure they are part of the existing session, the traditional access list is not capable of detecting whether a packet is actually part of an existing upper-layer conversation. Access lists can be used to examine individual packets to determine if it is part of an existing conversation, but only through the use of an established keyword. This check, however, is limited to TCP conversations because UDP is a connectionless protocol and no flags exist in the protocol header to indicate an existing connection. Furthermore, in TCP conversations, this control can easily be compromised through spoofing.

Information Examination Limits

Traditional access lists have a limited capability to examine packet information above the IP layer, no way of examining information above layer 4, and are incapable of securely handling layer 4 information. Extended access lists can examine a limited amount of information in layer 4 headers. There are, however, enhancements that exist in more recent access list technology; these are described later in this chapter.

IP Address Spoofing

IP address spoofing is a common network attack technique used by computer hackers to disrupt network systems. Address filtering is used to combat IP address spoofing, which is the impersonation of a network address so that the packets sent from the impersonator's PC appear to have originated from a trusted PC. For the spoof to work successfully, the impersonator's PC instead of the legitimate PC whose network address the impersonator is impersonating. To achieve this, the impersonator would need to guess the initial sequence number sent in reply to the SYN request from the attacker's PC during the initial TCP three-way handshake. The destination PC, upon receiving a SYN request, returns a SYN-ACK response to the legitimate owner of the spoofed IP address. As a result, the impersonator never receives the response, therefore necessitating guessing the initial sequence number contained in the SYN-ACK packet so that the ACK sent from the attacker's PC would contain the correct information to complete the handshake. At this point, the attacker or hacker has successfully gained entry into the network.

Attackers need not gain entry into a network to cause damage. For example, an attacker could send malicious packets to a host system for purposes of disrupting the host's capability to function. This type of attack is commonly known as a denial-of-service attack. The attacker only needs to spoof the originating address, never needing to actually complete the connection with the attacked host.

Standard Access Lists

Standard access lists are very limited functionally because they allow filtering only by source IP address. Typically, this does not provide the level of granularity needed to provide adequate security. They are defined within a range of 1 to 99; however, named access lists can also be used to define the list. By using names in the access list, the administrator avoids the need to recreate the entire access list after specific entries in the list are deleted.

In standard access lists, each entry in the list is read sequentially from beginning to end as each packet is processed. Any remaining access list statements are ignored once an entry or statement is reached in the list that applies to that packet. As a result, the sequence or order of the access list statements is critical to the intended processing/routing of a packet. If no match is made between the access list statement and the packet, the packet continues to be examined by subsequent statements until the end of the list is reached and it becomes subject to the implicit "deny all" feature. The implicit deny all can be overridden by an explicit permit all statement at the end of the list, allowing any packet that has not been previously explicitly denied to be passed through the router. This is not a recommended or sound security practice. The best practice is to use explicit

permit statements in the access list for those packets that are allowed and utilize the implied deny all to deny all other packets. This is a much safer practice simply because of the length and complexity of standard access lists.

Standard access lists are best used where there is a requirement to limit virtual terminal access, limit Simple Network Management Protocol (SNMP) access, and filter network ranges. Virtual terminal access is the ability to Telnet into a router from an external device. To limit remote access to routers within the network, an extended access list could be applied to every interface. To avoid this, a standard access list can be applied to restrict remote access from only a single device (inbound). In addition, once remote access is gained, all outbound access can be restricted by applying a standard access list to the outbound interface.

Standard access lists are also used to limit SNMP access. SNMP is used in a data network to manage network devices such as servers and routers. SNMP is used by network administrators and requires the use of a password or authentication scheme called a community string. Standard access lists are used to limit the IP addresses that allow SNMP access through routers, reducing the exposure of this powerful capability.

Standard access lists are also used to filter network ranges, especially where redistribution routes exist between different routing protocols. Filtering prevents routing redistribution from an initial protocol into a second protocol and then back to the initial protocol. That is, the standard access list is used to specify the routes that are allowed to be distributed into each protocol.

Extended IP Access Lists

As indicated by their name, extended access lists are more powerful than standard access lists, providing much greater functionality and flexibility. Both standard and extended access lists filter by source address; however, extended lists also filter by destination address and upper layer protocol information. Extended access lists allow for filtering by type of service field and by IP precedence. Another feature of extended access lists is logging. Access list matches can be logged through the use of the LOG keyword placed at the end of an access list entry. This feature is optional and, when invoked, sends log entries to a database facility enabled by the router.

When establishing a security policy on the network using router access lists, a couple of key points must be noted. With regard to the placement of the access list relative to the interface, the standard access list should be placed as close to the destination as possible and the extended access list should be placed as close to the source as possible. Because standard access lists use only the source address to determine whether a packet is to be permitted or denied, placement of this list too close to the source would result in blocking packets that were intended to be included. As a result, extended access lists would be more appropriately placed close to the source because these lists typically use both source and destination IP addresses.

A strong security policy should also include a strategy to combat spoofing. Adding "anti-spoofing" access list entries to the inbound access list would help support this effort. The anti-spoofing entries are used to block IP packets that have a source address of an external network or a source address that is invalid. Examples of invalid addresses include loopback addresses, multicast addresses, and unregistered addresses. Spoofing is a very popular technique used by hackers. The use of these invalid address types allows hackers to engage in attacks without being traced. Security administrators are unable to trace packets back to the originating source when these illegitimate addresses are used.

Dynamic Access Lists

Dynamic access lists provide the capacity to create dynamic openings in an access list through a user authentication process. These list entries can be inserted in all of the access list types presented thus far — traditional, standard, and extended access lists. Dynamic entries are created in the inbound access lists after a user has been authenticated and the router closes the Telnet session to the router invoked by the user. This dynamic entry then is used to permit packets originating from the IP address of the user's workstation. The dynamic entry will remain until the idle timeout is reached or the maximum timeout period expires. Both of these features, however, are optional, and if not utilized, will cause the dynamic entries to remain active until the next router reload process occurs. Timeout parameters, however, are recommended as an important security measure.

Use of dynamic access lists must be carefully planned because of other security limitations. Only one set of access is available when using dynamic access — different levels of access cannot be provided. In addition,

when establishing the session, logon information is passed without encryption, allowing hackers access to this information through sniffer software.

Conclusion

Network router security is a critical component of an organization's overall security program. Router security is a complex and fast-growing technology that requires the constant attention of security professionals. This chapter has examined the important aspects of basic router security features and how they must be enabled to protect organizations from unauthorized attacks. Future security improvements are inevitable as the threat and sophistication of attacks increase over time.

DIAL-UP SECURITY CONTROLS

Alan Berman and Jeffrey L. Ott

INSIDE

Direct-Dial and Packet-Switching Transmission, Passwords, Microcomputer Access, Dial-Up/Callback Systems, Encryption Intrusion Monitoring

PROBLEMS ADDRESSED

As the need to provide information has grown, the capacity for unauthorized users to gain access to online dial-up computer systems has increased. This threat — and the consequences inherent in such an exposure — may have devastating consequences, from penetrating defense department computers to incapacitating large networks or shared computer facilities. Increased reliance on LAN-based microcomputers not only raises the threat of unauthorized modification or deletion of company critical data, but it also adds the possibility of infecting network users.

Providing dial-in access is not limited only to network or system access for the general user. There is often a greater exposure hidden in modems connected to maintenance ports on servers, routers, switches, and other network infrastructure devices. Any computing device with an attached modem is a potential target for someone looking for a device to hack. The problems associated with maintenance ports are the following:

- Little attention is given to these ports because only one or two people use it, including a vendor.
- They provide immediate access to low-level administrative authority on the device.
- Often, they are delivered with default user IDs and passwords, which are never changed.
- If used by a vendor, vendors have a notorious habit of using the same ID and password on all their machines.

PAYOFF IDEA

Several measures are available to help protect computer resources and data from unauthorized dial-up users. Some or all of these measures can be implemented to increase computer and data security. This article discusses products and services currently available to minimize the risk that a system may be compromised by an intruder using a dial-up facility.

Look for modems directly attached to host systems, servers, switches, routers, PCs (both in offices and on the computer room floor), PBXs, and CBXs. Check with the department providing telecommunication services. They may have a list of phone numbers assigned to modems. However, do not count on this. At the very least, they should be able to provide a list of analog lines. Most of these will be fax machines, but some will be modems. Finally, to ensure the identification of all modems, run a war-dialer against the phone numbers in the company's exchange.

Although the threats are numerous and consequences great, very few organizations have complete security programs to combat this problem. This article describes the steps that need to be taken to ensure the security of dial-up services.

TYPES OF DIAL-UP ACCESS

Dial-up capability uses a standard telephone line. A modem, the interface device required to use the telephone to transmit and receive data, translates a digital stream into an analog signal. The modem at the user's site converts computer data coded in bits into an analog signal and sends that signal over a telephone line to the computer site. The modem at the computer site translates the analog signal back to binary-coded data. The procedure is reversed to send data from the computer site to the user site.

Dial-up capability is supplied through standard telephone company direct-dial service or packet-switching networks.

Direct Dial

With a direct-dial facility, a user dials a telephone number that connects the originating device to the host computer. The computer site maintains modems and communications ports to handle the telephone line.

Standard dial-up lines can be inordinately expensive, especially if the transmission involves anything other than a local call. For example, a customer in California who needs access to a brokerage or bank service in New York would find the cost of doing business over a standard telephone company dial-up line prohibitive for daily or weekly access and two-way transmission.

Packet Switching

Packet-switching networks provide a solution to the prohibitive telephone costs of long-distance dial-up service. The California user, for example, need only install the same type of telephone and modem on a direct dial-up system. Instead of dialing a number with a New York area code, the user dials a local telephone number that establishes a connection to the switching node within the area.

Internally, packet-switching data transmission is handled differently from direct dial-up message transmission. Rather than form a direct connection and send and receive streams of data to and from the host computer, packet-switching networks receive several messages at a node. Messages are then grouped into data packets. Each packet has a size limitation, and messages that exceed this size are segmented into several packets. Packets are passed from node to node within the network until the assigned destination is reached. To indicate the destination of the message, the user enters an assigned ID code and a password. The entered codes correlate to authorization and specify the computer site addressed. For the user's purposes, the connection to the host computer is the same as if a dial-up line had been used, but the cost of the call is drastically reduced.

In both dial-up service and packet-switching networks, the host site is responsible for protecting access to data stored in the computer. Because packet-switching networks require a user ID and a password to connect to a node, they would appear to provide an extra measure of security; however, this is not always the case, and this should not be a reason to abrogate the responsibility for security to the packet-switching network vendor.

For some time, users of certain vendor's packet-switching network facilities have known that it is possible to bypass the user ID and password check. It has been discovered that with very little experimentation, anyone can gain access to various dial-up computer sites in the United States and Canada because the area codes of these computer site communications ports are prefaced with the three digits of the respective telephone network area codes. The remainder of the computer address consists of three numeric characters and one alphanumeric character. Therefore, rather than determine a 10-digit dial-up number, which includes the area code, a hacker must simply determine the proper numeric code sequence identifier. The alphabetic character search is simplified or eliminated by assuming that the first address within the numeric set uses the letter A, the second B, and so on, until the correct code is entered. Accessing a computer site requires only a local node number, and these numbers are commonly posted in packet-switching network sites. Use of the local node number also substantially reduces dial-up access line costs for the unauthorized user. Packet-switching network vendors have responded to this problem with varying degrees of success, but special precaution should be exercised when these networks are used.

MINIMIZING RISKS

Hackers have a myriad of ways to obtain the phone number that can provide them with access to computer systems. Attempts can be made to randomly dial phone numbers in a given area code or phone exchange

using demon dialers or war dialers. These were popularized in the 1980 movie, *War Games.* These hacking programs can be very useful in locating all the authorized and unauthorized modems located on the premises. War dialers can be written using a scripting language, such as that provided by the communications software package Procomm Plus, or several can be found at various sites on the Internet. Understanding these dialers is very helpful in understanding the requirements needed for securing dial-in connections.

Simpler methods, such as calling a company and asking for the dial-up number, may meet with success if the caller is believable and persistent. Calling operational personnel at the busiest time of the day (e.g., end of the day, before stock market or bank closes) is more likely to get a response from a harried computer operator or clerk.

Other methods consist of rummaging through trash to locate discarded phone records that may reveal the number of the dial-up computer. A hacker will try these numbers manually, hoping to find the right line. This will most likely be the one that has the longest duration telephone call.

There are also less esoteric means by which phone numbers can be acquired. Online services for such applications as E-mail, ordering merchandise, bank access, stock trading, and bulletin boards often have their numbers published in the sample material that they mail. In fact, it is often possible to look over the shoulder of someone demonstrating the service and watch him or her dial the number. If the demonstration is automated, the number may appear on the screen.

Although the practice of listing the number in the phone directory or having it available from telephone company information operators has been curtailed, this remains a potentially effective method.

No matter how it is obtained, the phone number can be quickly spread throughout the hacker community by means of underground bulletin boards. Once the number is disseminated, the phreaker's game begins. It is now a matter of breaking the security that allows users to log on.

Despite the fact that there are physical devices (e.g., tokens, cards, PROMS) that can be used to identify users of remote computer systems, almost all of these systems rely on traditional user identification and password protection mechanisms for access control.

Identification

The primary means of identifying dial-up users is through the practice of assigning user IDs. Traditionally, the user ID is 6 or 7 alphanumeric characters. Unfortunately, user IDs tend to be sequential (e.g., USER001, USER002), which provides an advantage to hackers. For example, hacker bulletin boards will report that company XYZ's user ID starts at XYZ001 and runs consecutively. The hacker who posted the note will state that he is attacking ID XYZ001. The first hacker who reads the notice will

leave a note saying that she will try to log on as user XYZ002, and the next hacker will take XYZ003. The net result is that multiple hackers will attack simultaneously, each targeting a different user ID. This significantly increases their chances of penetrating the system.

Unknowingly, some security software can actually aid in identifying valid user IDs. When a hacker attempts to enter the user ID and password, the system may respond to the entry of an invalid user ID with the message "Invalid ID, Please Reenter." This allows the hacker to focus his efforts on finding a valid ID, without having to deal with the far more complex effort of obtaining a valid ID and password.

The same type of security system will invariably tell the intruder that he has found a valid user ID by issuing the error message "Invalid Password, Please Reenter." This in effect tells the hacker that he has found a valid ID. He may then proceed to try to find the user ID sequence pattern (to post on the bulletin board) or focus his attention on trying to break the password protection.

Log-ons that request a valid user ID before requesting the password can also provide system attackers with a major advantage. The best security system requires entry of both user ID and password at the same time. The system attempts to validate the combination; if it is found invalid, it responds with "User ID/Password Invalid, Please Reenter." This is the only error message sent, regardless of which item is not valid.

Passwords. Use of passwords is the most widely employed method of authenticating the identity of a computer system user. Passwords are easy to design and can be implemented quickly without requiring additional hardware. When the proper methodology is used, password security provides a significant deterrent to unauthorized system access without major expenditure.

Certain rules should be followed to make password identification and authentication an effective security tool:

- Passwords should be of sufficient length to prevent their discovery by manual or automated system attack or pure guesswork.
- Passwords should not be so long that they are difficult to remember and must therefore be written down.
- Passwords should be derived by algorithm or stored on a one-way encrypted file.
- Passwords are most effective when they are arbitrarily assigned.
- Passwords should be distributed under tight controls, preferably online.
- An audit trail of previously issued passwords should be established.
- Individual passwords should be private.
- The use of portable token-generated random passwords should be encouraged. The tokens are relatively inexpensive and highly reliable.

If sufficient time is not available for an in-depth study of password identification methodology, a basically sound password structure can be created using a six-character password that has been randomly selected and stored on an encrypted file. Such a procedure provides some measure of security, but should be taken to design and implement a more substantial methodology.

Multiple passwords can be used for accessing various levels of secured data. This system requires that the user have a different password for each increasingly more sensitive level of data. Even using different passwords for update and inquiry activities provides considerably more security than one password for all functions.

Computer and network security systems have made some gains over the last decade. Former problems that resulted from accessing a dropped line and reconnecting while bypassing log-on security have been resolved. Even direct connect (i.e., addressing the node and bypassing user ID and password validation) has been corrected.

Aside from obtaining telephone numbers, user IDs, and password information from other hackers through bulletin boards or other means, hackers have three basic ways of obtaining information necessary to gain access to the dial-up system:

- Manual and computer-generated user ID and password guessing
- Personal contact
- Wiretaps

Given a user ID, the hacker can attempt to guess the password in either of two ways: by trying commonly used passwords or programming the computer to attack the password scheme by using words in the dictionary or randomly generated character sets. The hacker can have the computer automatically dial the company system he wishes to penetrate, and attempt to find a valid user ID and password combination. If the host system disconnects him, the computer redials and continues to try until the right combination is found and access is gained. This attack can continue uninterrupted for as long as the computer system remains available. The drawback to this approach is that the call can be traced if the attempts are discovered.

A simpler approach is for the hacker to personally visit the site of the computer to be attacked. Befriending an employee, he or she may be able to gain all the information needed to access the system. Even if the hacker is only allowed on the premises, he or she will often find a user ID and password taped to the side of a terminal, tacked on the user's bulletin board, or otherwise conspicuously displayed. Basic care must be taken to protect user IDs and passwords. For example, they should never be shared or discussed with anyone.

Potentially the most damaging means of determining valid user IDs and passwords is the use of the wiretapping devices on phone lines to record information. Plaintext information can be recorded for later use. Wiretapping indicates serious intent by the hacker to commit a serious act. It exposes the hacker to such risk that it is often associated with theft, embezzlement, or espionage.

Even encryption may not thwart the wiretapping hacker. The hacker can overcome the inability to interpret the encrypted data by using a technique called replay. This tactic involves capturing the cipher text and retransmitting it later. Eventually, the hacker captures the log-on sequence cipher and replays it. The data stream is recognized as valid, and the hacker is therefore given access to the system. The only way to combat a replay attack is for the ciphered data to be timed or sequence stamped. This ensures that the log-in can be used only once and will not be subject to replay.

The best defense against wiretapping is physical security. Telephone closets and rooms should be secured by card key access. Closed-circuit cameras should monitor and record access. If the hacker cannot gain access to communications lines, he cannot wiretap and record information.

Microcomputer Password Problems. The use of microcomputer and communications software packages has presented another problem to those who rely on passwords for security. These packages enable the user to store and transmit such critical information as telephone numbers, user identification, and passwords.

Many remote access programs, such as Microsoft Windows 95 Dial-Up Network program or Symantec's pcANYWHERE, give the user the option of saving the user ID, password, and dial-in phone number for future use. This practice should be strongly discouraged, especially on laptop computers. Laptop computers are prime targets for theft, both for the physical item and for the information contained on them. If a thief were to steal a laptop with the dial-up session information (phone number, user ID, and password) saved, they would have immediate full access to whatever system the owner had access.

The discussion of laptop security is worthy of an entire section in and of itself; however, for the purposes of this discussion, suffice it to say that users should be thoroughly educated in the proper way of using and securing dial-up applications.

An effective but more cumbersome way to enhance security is to obscure the visible display of destination and identification information. The user can either reduce the display intensity until it is no longer visible, or turn off the monitor until the sign-on is completed and all security information is removed from the screen. Some software packages alert the user when the sign-on process is completed by causing the computer

to issue an audible beep. Even software packages that do not issue an audible signal can be enhanced by this blackout technique. An estimation of the amount of time required to complete the sign-on process can give an idea of when to make the information visible again.

A BRIEF AUTHENTICATION REQUIREMENTS REVIEW

Throughout human history and lore, a person has been authenticated by demonstrating one of the following:

- Something you know
- Something you have
- Something you are

Whether it was Ali Baba saying, "Open Sesame" (something you know), Indiana Jones with the crystal on the staff (something you have), or "Rider coming in … It's Dusty!! Open the gates! Open the gates!!" (physical recognition — something you are), one person has permitted or denied access to another based on meeting one of these "factors of identification."

Satisfying only one factor, such as knowing a password (something you know), can easily be defeated. In secure environments, it is better to meet at least two of the three factors of identification. This can best be seen in the application of a bank ATM card. To use the card — to access an account — one must have an ATM card (something you have) and know the PIN assigned to that card (something you know). When and only when one can meet both factors of identification, can one access the money in the account.

The third factor of identification is represented today through the use of biometrics, such as retinal scans, fingerprints, and voiceprints.

Secure dial-in in today's market is the ability to meet at least two of these three factors of identification.

Physical Devices

Whereas passwords are a relatively inexpensive means of providing identification and authentication security in the dial-up environment, physical devices involve capital expenditure. The cost depends on the intricacy of the device. Determining which device is best suited to a particular environment requires careful analysis of the consequences of unauthorized dial-up penetration.

The market is constantly changing in response to the available technology and market forces. Currently, one technology is dominant in protecting dial-in resources: dynamic password generators. In its most basic form, there are two components to a dynamic password generator authentication system: (1) the host system, which could be a server execut-

ing vendor-supplied remote access code, or (2) a vendor-supplied hardware/software front-end and a handheld device, often resembling a calculator or credit card. There are two variations in this field, time synchronous and challenge/response.[1]

Time Synchronous. One vendor prevails in this market, Security Dynamics Technologies, Inc. (http://www.securid.com/). Their product line incorporates proprietary software that generates a new six-digit password every 60 seconds, based, in part on Greenwich Mean Time (GMT). A user is issued a small credit-card-sized "token" that has been registered in a central database on the remote access device. When a user dials in, he or she reaches the remote access device, which authenticates the user based on the user ID and the password displayed at that moment on the token. After authentication, the user is granted access to the target device or network. Security Dynamics has several types and implementations of their tokens (credit card sized, key fobs, PCMCIA cards, and software based) and many different implementations of their authentication "kernel" or code. Additionally, many third-party products have licensed Security Dynamics code in their remote access/authentication products.

Challenge/Response. Several vendors have implemented another dial-in authentication method that also utilizes hand-held tokens and PC software. Whereas the time-synchronous tokens rely on a password generated based on the current GMT, challenge/response tokens utilize a shared algorithm and a unique "seed" value or key. When a dial-in user accesses a remote access device using a challenge/response token, he or she is authenticated based on the expected "response" to a given "challenge" generated by the user's token. Challenge/response technology also comes in different types and implementations of tokens, software, and hardware. Major vendors of challenge/response technology include AssureNet Pathways, Inc. (http://www.assurenetpathways.com/) and LeeMah Datacom Security Corporation (http://www.leemah.com/).

Dial-Up/Callback Systems

To protect against the kind of system penetration possible when only precoded identifiers are used, manufacturers have developed dial-up/callback systems. With this technique, two telephone calls must be completed before access is granted. After dialing the host computer, the user must enter a valid password. On receipt of the password, the host computer terminates the connection and automatically places a call to the telephone number associated with the password. If an authorized terminal is being used, the connection is established and the user can proceed. Some dial-up/callback systems place the return call through least-cost routing on local lines, WATS lines, and other common carrier facilities, thereby reducing the cost of the callback procedure.

One problem associated with dial-up/callback systems is that the authorized caller is restricted to a single predetermined location. This restriction prohibits the use of portable terminals for travel assignments. It also requires multiple IDs for use at different sites.

Other Technologies

This field is changing. An organization may wish to investigate newer or less popular technologies, depending on their organizational requirements. Included are devices that attach to a serial or parallel port of a PC or laptop, PCMCIA cards, and biometrics.

If dynamic password generators are the authentication of choice today, biometrics will be the authentication of choice tomorrow. Recent developments have increased reliability considerably and lowered costs. Expect to see more product offerings in biometric authentication in the next few years.

The decision to purchase any of these devices depends on such factors as cost of installation and cost of labor to monitor the hardware.

ENCRYPTION

If an unauthorized dial-up user penetrates the identification and authentication defenses of a computer system, encryption can forestall if not prevent data modification and theft. Encryption is technically a privacy measure, as opposed to a pure security precaution. It is intended to make the information unintelligible to anyone who does not have the proper decryption capability (key, algorithm, or decryption device). This prevents unauthorized personnel who do access a system from being able to read the data that they may want to alter, destroy, or circulate.

For data communications, messages are encrypted at the point of transmission and can only be decrypted at a terminal supplied with the key used in the encryption process. Various encryption algorithms are available, and the complexity of the algorithm should depend on the value of the data being protected. The National Institute of Standards and Technology's Data Encryption Standard (DES), which is the only encryption method to be used by civilian agencies of the federal government, is widely used and highly resistant to automated attack. Encryption should be considered for microcomputer transmissions, especially when it is likely that cellular communications will be used. This eliminates sending cleartext over open airwaves.

Although the encryption and decryption process is primarily used in data transmission, it can also protect critical files and programs from external threats. Encryption data and program source code make it very difficult for an unauthorized user to determine what information or code is contained in a file. Encrypting files also protects file relationships that can be determined by reading the source code of programs that use such

files. For the intruder unfamiliar with an organization's data components and flow, such an obstacle can discourage any further unauthorized activity. Even for authorized users, encrypted files bear no relationship to the information the users are accustomed to seeing. In addition, if used only for key files and programs, encryption does not involve significant use of storage.

THE FINAL DEFENSE

Hackers are becoming more and more proficient in accessing computer systems, despite the best efforts to stop them. There is a good chance that any system's security may be breached. If this happens, it is imperative that effective security measures be in place to identify the hacker and either trace the call or disconnect. After the unauthorized access is halted, the security administrator needs to determine how access was gained and the nature and extent of the damage. This is necessary for repairing damage and strengthening defenses from further attack.

One of the ways to identify an unauthorized user is to monitor users' attempts to access transactions, files, and data that are not in their security profile. If there are repeated violations (e.g., five consecutive denied accesses), some security action should be taken. This could be in the form of disconnecting the line, invalidating the user ID, or at a minimum logging the violations for further discussion with the user.

A major credit reference firm uses postintrusion monitoring software equipped with artificial intelligence to establish a normal pattern of activity for how a user accesses information. For example, user XYZ001 may usually access customer information through searching by social security number. User XYZ002 may access information using a person's name and address. When a user logs on, that person's activity pattern is monitored and compared to the user's normal activity profile. Should major discrepancies arise, the company attempts to contact the customer to ensure the validity of his or her requests. Such activity monitoring has thwarted many unauthorized users.

Ultimately, it is every user's responsibility to help protect systems from unauthorized access. The best way to help is to be wary. End users should check the last log-on time and date displayed during a successful log-on. If the user has any doubts that this was a valid log-on, he or she should contact the appropriate authority. This not only protects the system, it also relieves the authorized user of the liability created when an intruder uses another person's ID.

RECOMMENDED COURSE OF ACTION

The security method chosen to protect central data sources has great impact on the organization's resources and procedures. Initial costs, implementation time, client reaction, and related factors can be addressed only

by performing a thorough risk analysis that examines current as well as future needs. The measures described in this article should be interpreted not as an isolated set of precautions, but as components of an overall security umbrella designed to protect the organization from all internal and external threats. The data security administrator must ensure that the first step provides a basis for establishing an organizational awareness that will lead to a more secure environment for dealing with all dial-up users. Specifically, the administrator should ensure that:

- A complete list of valid dial-up users and their current status is maintained, eliminating all employees who are no longer with the company or whose position no longer requires access
- Protection is provided for all password schemas and files
- A minimum of two factors of identification are provided
- A test machine (not connected to any network) is used to validate newly downloaded software
- All users are regularly reminded of security policies and current versions of such policies are distributed to employees.

These steps, combined with a thorough set of policies and an educated user community, can significantly enhance the security of a dial-up environment.

Alan Berman has been involved in the evaluation, design, and implementation of online security systems since 1974. He has written numerous articles and conducted seminars on security-related topics. He resides in Irvington, NY.

Jeffrey L. Ott has 13 years of applied experience in international information security services. During his career, he has consulted with and worked for financial organizations, Fortune 500 corporations, as well as small and mid-sized companies. He currently manages Price Waterhouse's Enterprise Security Solutions group in Denver, CO.

Note

1. Reference to or exclusion of specific companies and their products in this discussion is neither an endorsement or denouncement. These companies represent market leaders at the time of this writing. One should thoroughly understand their organizational dial-in requirements and select a dial-in solution based on the ability of the vendor to meet or exceed one's stated needs.

23

What's Not So Simple about SNMP?

Chris Hare, CISSP, CISA

The Simple Network Management Protocol, or SNMP, is a defined Internet standard from the Internet Engineering Task Force, as documented in Request for Comment (RFC) 1157. This chapter discusses what SNMP is, how it is used, and the challenges facing network management and security professionals regarding its use.

While several SNMP applications are mentioned in this chapter, no support or recommendation of these applications is made or implied. As with any application, the enterprise must select its SNMP application based upon its individual requirements.

SNMP Defined

SNMP is used to monitor network and computer devices around the globe. Simply stated, network managers use SNMP to communicate management information, both status and configuration, between the network management station and the SNMP agents in the network devices.

The protocol is aptly named because, despite the intricacies of a network, SNMP itself is very simple. Before examining the architecture, a review of the terminology used is required.

- *Network element:* any device connected to the network, including hosts, gateways, servers, terminal servers, firewalls, routers, switches and active hubs.
- *Network management station (or management station):* a computing platform with SNMP management software to monitor and control the network elements; examples of common management stations are HP Openview and CA Unicenter.
- *SNMP agent:* a software management agent responsible for performing the network management functions received from the management station.
- *SNMP request:* a message sent from the management station to the SNMP agent on the network device.
- *SNMP trap receiver:* the software on the management station that receives event notification messages from the SNMP agent on the network device.
- *Management information base:* a standard method identifying the elements in the SNMP database.

A network configured to SNMP for the management of network devices consists of at least one SNMP agent and one management station. The management station is used to configure the network elements and receive SNMP traps from those elements.

Through SNMP, the network manager can monitor the status of the various network elements, make appropriate configuration changes, and respond to alerts received from the network elements (see Exhibit 23.1). As networks increase in size and complexity, a centralized method of monitoring and management is essential. Multiple management stations may exist and be used to compartmentalize the network structure or to regionalize operations of the network.

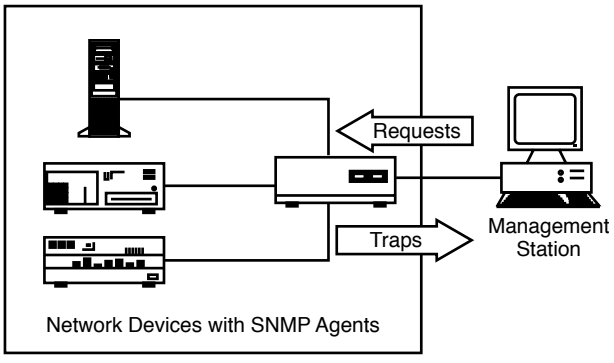


EXHIBIT 23.1 The SNMP network manager.

SNMP can retrieve the configuration information for a given network element in addition to device errors or alerts. Error conditions will vary from one SNMP agent to another but would include network interface failures, system failures, disk space warnings, etc. When the device issues an alert to the management station, network management personnel can investigate to resolve the problem. Access to systems is controlled through knowledge of a community string, which can be compared to a password. Community strings are discussed in more detail later in the chapter, but by themselves should not be considered a form of authentication.

From time to time it is necessary for the management station to send configuration requests to the device. If the correct community string is provided, the device configuration is changed appropriately. Even this simple explanation evidences the value gained from SNMP. An organization can monitor the status of all its equipment and perform remote troubleshooting and configuration management.

The Management Information Base (MIB)

The MIB defines the scope of information available for retrieval or configuration on the network element. There is a standard MIB all devices should support. The manufacturer of the device can also define custom extensions to the device to support additional configuration parameters. The definition of MIB extensions must follow a defined convention for the management stations to understand and interpret the MIB correctly.

The MIB is expressed using the ASN.1 language; and, while important to be aware of, it is not a major concern unless you are specifically designing new elements for the MIB. All MIB objects are defined explicitly in the Internet standard MIB or through a defined naming convention. Using the defined naming convention limits the ability of product vendors to create individual instances of an MIB element for a particular network device. This is important, given the wide number of SNMP capable devices and the relatively small range of monitoring station equipment.

An understanding of the MIB beyond this point is only necessary for network designers who must concern themselves with the actual MIB structure and representations. Suffice it to say that for this discussion, the MIB components are represented using English identifiers.

SNMP Operations

All SNMP agents must support both inspection and alteration of the MIB variables. These operations are referred to as *SNMP get* (retrieval and inspection) and *SNMP set* (alteration). The developers of SNMP established only these two operations to minimize the number of essential management functions to support and to avoid the introduction of other imperative management commands. Most network protocols have evolved to support a vast array of potential commands, which must be available in both the client and the server. The File Transfer Protocol (FTP) is a good example of a simple command set that has evolved to include more than 74 commands.

The SNMP management philosophy uses the management station to poll the network elements for appropriate information. SNMP uses *traps* to send messages from the agent running on the monitored system to the monitoring station, which are then used to control the polling. Limiting the number of messages between

the agent and the monitoring station achieves the goal of simplicity and minimizes the amount of traffic associated with the network management functions.

As mentioned, limiting the number of commands makes implementing the protocol easier: it is not necessary to develop an interface to the operating system, causing a system reboot, or to change the value of variables to force a reboot after a defined time period has elapsed.

The interaction between the SNMP agent and management station occurs through the exchange of protocol messages. Each message has been designed to fit within a single User Datagram Protocol (UDP) packet, thereby minimizing the impact of the management structure on the network.

Administrative Relationships

The management of network elements requires an SNMP agent on the element itself and on a management station. The grouping of SNMP agents to a management station is called a *community*. The community string is the identifier used to distinguish among communities in the same network. The SNMP RFC specifies an authentic message as one in which the correct community string is provided to the network device from the management station. The authentication scheme consists of the community string and a set of rules to determine if the message is in fact authentic. Finally, the SNMP authentication service describes a function identifying an authentic SNMP message according to the established authentication schemes.

Administrative relationships called communities pair a monitored device with the management station. Through this scheme, administrative relationships can be separated among devices. The agent and management station defined within a community establish the SNMP access policy. Management stations can communicate directly with the agent or, in the event of network design, an SNMP proxy agent. The proxy agent relays communications between the monitored device and the management station.

The use of proxy agents allows communication with all network elements, including modems, multiplexors, and other devices that support different management frameworks. Additional benefits from the proxy agent design include shielding network elements from access policies, which might be complex.

The community string establishes the access policy community to use, and it can be compared to passwords. The community string establishes the password to access the agent in either read-only mode, commonly referred to as the public community, or the read-write mode, known as the private community.

SNMP Requests

There are two access modes within SNMP: *read-only* and *read-write*. The command used, the variable, and the community string determine the access mode. Corresponding with the access mode are two community strings, one for each access mode. Access to the variable and the associated action is controlled by:

- If the variable is defined with an access type of *none*, the variable is not available under any circumstances.
- If the variable is defined with an access type of *read-write* or *read-only*, the variable is accessible for the appropriate *get, set,* or *trap* commands.
- If the variable does not have an access type defined, it is available for *get* and *trap* operations.

However, these rules only establish what actions can be performed on the MIB variable. The actual communication between the SNMP agent and the monitoring station follows a defined protocol for message exchange. Each message includes the:

- SNMP version identifier
- Community string
- Protocol data unit (PDU)

The SNMP version identifier establishes the version of SNMP in use — Version 1, 2, or 3. As mentioned previously, the community string determines which community is accessed, either public or private. The PDU contains the actual SNMP trap or request. With the exception of traps, which are reported on UDP port 162, all SNMP requests are received on UDP port 161. RFC 1157 specifies that protocol implementations need not accept messages more than 484 bytes in length, although in practice a longer message length is typically supported.

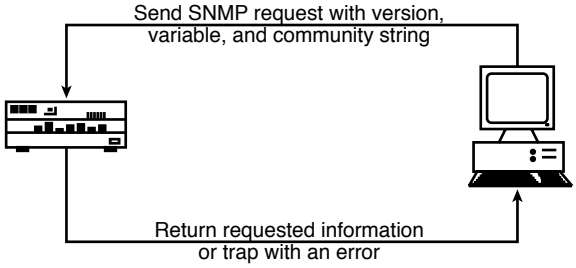


EXHIBIT 23.2 The SNMP transmission process.

There are five PDUs supported within SNMP:

1. GetRequest-PDU
2. GetNextRequest-PDU
3. GetResponse-PDU
4. SetRequest-PDU
5. Trap-PDU

When transmitting a valid SNMP request, the PDU must be constructed using the implemented function, the MIB variable in ASN.1 notation. The ASN.1 notation, the source and destination IP addresses, and UDP ports are included along with the community string. Once processed, the resulting request is sent to the receiving system.

As shown in Exhibit 23.2, the receiving system accepts the request and assembles an ASN.1 object. The message is discarded if the decoding fails. If implemented correctly, this discard function should cause the receiving system to ignore malformed SNMP requests. Similarly, the SNMP version is checked; and if there is a mismatch, the packet is also dropped. The request is then authenticated using the community string. If the authentication fails, a trap may be generated indicating an authentication failure, and the packet is dropped.

If the message is accepted, the object is again parsed to assemble the actual request. If the parse fails, the message is dropped. If the parse is successful, the appropriate SNMP profile is selected using the named community, and the message is processed. Any resulting data is returned to the source address of the request.

The Protocol Data Unit

As mentioned, there are five protocol data units supported. Each is used to implement a specific request within the SNMP agent and management station. Each will be briefly examined to review purpose and functionality.

The *GetRequest* PDU requests information to be retrieved from the remote device. The management station uses the *GetRequest* PDU to make queries of the various network elements. If the MIB variable specified is matched exactly in the network element MIB, the value is returned using the *GetResponse* PDU. We can see the direct results of the *GetRequest* and *GetResponse* messages using the *snmpwalk* command commonly found on Linux systems:

```
[chare@linux chare]$ for host in 1 2 3 4 5
> do
> snmpwalk 192.168.0.$host public system.sysDescr.0
> done
system.sysDescr.0 = Instant Internet version 7.11.2
Timeout: No Response from 192.168.0.2
system.sysDescr.0 = Linux linux 2.4.9—31 #1 Tue Feb 26 07:11:02 EST
2002 i686
Timeout: No Response from 192.168.0.4
Timeout: No Response from 192.168.0.5
[chare@linux chare]$
```

Despite the existence of a device at all five IP addresses in the above range, only two are configured to provide a response; or perhaps the SNMP community string provided was incorrect.

Note that, on those systems where *snmpwalk* is not installed, the command is available in the net-ucb-cnmp source code available from many network repositories.

The *GetResponse* PDU is the protocol type containing the response to the request issued by the management station. Each *GetRequest* PDU results in a response using *GetResponse*, regardless of the validity of the request.

The *GetNextResponse* PDU is identical in form to the *GetResponse* PDU, except it is used to get additional information from a previous request. Alternatively, table traversals through the MIB are typically done using the *GetNextResponse* PDU. For example, using the *snmpwalk* command, we can traverse the entire table using the command:

```
# snmpwalk localhost public
system.sysDescr.0 = Linux linux 2.4.9—31 #1 Tue Feb 26 07:11:02 EST
2002 i686
system.sysObjectID.0 = OID: enterprises.ucdavis.ucdSnmpAgent.linux
system.sysUpTime.0 = Timeticks: (4092830521) 473 days, 16:58:25.21
system.sysContact.0 = root@localhost
system.sysName.0 = linux
system.sysLocation.0 = Unknown
system.sysORLastChange.0 = Timeticks: (4) 0:00:00.04
…
<end of snmpwalk output>
```

In our example, no specific MIB variable is requested, which causes all MIB variables and their associated values to be printed. This generates a large amount of output from *snmpwalk*. Each variable is retrieved until there is no additional information to be received.

Aside from the requests to retrieve information, the management station also can set selected variables to new values. This is done using the *SetRequest* PDU. When receiving the *SetRequest* PDU, the receiving station has several valid responses:

- If the named variable cannot be changed, the receiving station returns a *GetResponse* PDU with an error code.
- If the value does not match the named variable type, the receiving station returns a *GetResponse* PDU with a bad value indication.
- If the request exceeds a local size limitation, the receiving station responds with a *GetResponse* PDU with an indication of too big.
- If the named variable cannot be altered and is not covered by the preceding rules, a general error message is returned by the receiving station using the *GetResponse* PDU.

If there are no errors in the request, the receiving station updates the value for the named variable. The typical read-write community is called *private*, and the correct community string must be provided for this access. If the value is changed, the receiving station returns a *GetResponse* PDU with a "No error" indication.

As discussed later in this chapter, if the SNMP read-write community string is the default or set to another well-known value, any user can change MIB parameters and thereby affect the operation of the system.

SNMP Traps

SNMP traps are used to send an event back to the monitoring station. The trap is transmitted at the request of the agent and sent to the device specified in the SNMP configuration files. While the use of traps is universal across SNMP implementations, the means by which the SNMP agent determines where to send the trap differs among SNMP agent implementations.

There are several traps available to send to the monitoring station:

- coldStart
- warmStart
- linkDown

- linkUp
- authenticationFailure
- egpNeighborLoss
- enterpriseSpecific

Traps are sent using the PDU, similar to the other message types, previously discussed.

The *coldStart* trap is sent when the system is initialized from a powered-off state and the agent is reinitializing. This trap indicates to the monitoring station that the SNMP implementation may have been or may be altered. The *warmStart* trap is sent when the system restarts, causing the agent to reinitialize. In a *warmStart* trap event, neither the SNMP agent's implementation nor its configuration is altered.

Most network management personnel are familiar with the *linkDown* and *linkUp* traps. The *linkDown* trap is generated when a link on the SNMP agent recognizes a failure of one or more of the network links in the SNMP agent's configuration. Similarly, when a communication link is restored, the *linkUp* trap is sent to the monitoring station. In both cases, the trap indicates the network link where the failure or restoration has occurred.

Exhibit 23.3 shows a device, in this case a router, with multiple network interfaces, as seen in a Network Management Station. The failure of the red interface (shown here in black) caused the router to send a *linkDown* trap to the management station, resulting in the change in color for the object. The green objects (shown in white) represent currently operational interfaces.

The *authenticationFailure* trap is generated when the SNMP agent receives a message with the incorrect community string, meaning the attempt to access the SNMP community has failed. When the SNMP agent communicates in an Exterior Gateway Protocol (EGP) relationship, and the peer is no longer reachable, an *egpNeighborLoss* trap is generated to the management station. This trap means routing information available from the EGP peer is no longer available, which may affect other network connectivity.

Finally, the *enterpriseSpecific* trap is generated when the SNMP agent recognizes an *enterpriseSpecific* trap has occurred. This is implementation dependent and includes the specific trap information in the message sent back to the monitoring station.

SNMP Security Issues

The preceding brief introduction to SNMP should raise a few issues for the security professional. As mentioned, the default SNMP community strings are public for read-only access and private for read-write. Most system and network administrators do not change these values. Consequently, any user, authorized or not, can obtain

EXHIBIT 23.3 Router with multiple network interfaces.

information through SNMP about the device and potentially change or reset values. For example, if the read-write community string is the default, any user can change the device's IP address and take it off the network.

This can have significant consequences, most notably surrounding the availability of the device. It is not typically possible to access enterprise information or system passwords or to gain command line or terminal access using SNMP. Consequently, any changes could result in the monitoring station identifying the device as unavailable, forcing corrective action to restore service.

However, the common SNMP security issues include:

- Well-known default community strings
- Ability to change the configuration information on the system where the SNMP agent is running
- Multiple management stations managing the same device
- Denial-of-service attacks

Many security and network professionals are undoubtedly familiar with the Computer Emergency Response Team (CERT) Advisory CA-2002-03 published in February 2002. While this is of particular interest to the network and security communities today, it should not overshadow the other issues mentioned above because many of the issues in CA-2002-03 are possible due to the other security issues.

Well-Known Community Strings

As mentioned previously, there are two SNMP access policies, read-only and read-write, using the default community strings of public and private, respectively. Many organizations do not change the default community strings. Failing to change the default values means it is possible for an unauthorized person to change the configuration parameters associated with the device.

Consequently, SNMP community strings should be treated as passwords. The better the quality of the password, the less likely an unauthorized person could guess the community string and change the configuration.

Ability to Change SNMP Configuration

On many systems, users who have administrative privileges can change the configuration of their system, even if they have no authority to do so. This ability to change the local SNMP agent configuration can affect the operation of the system, cause network management problems, or affect the operation of the device.

Consequently, SNMP configuration files should be controlled and, if possible, centrally managed to identify and correct configuration changes. This can be done in a variety of ways, including tools such as *tripwire*.

Multiple Management Stations

While this is not a security problem per se, multiple management stations polling the same device can cause problems ranging from poor performance, to differing SNMP configuration information, to the apparent loss of service.

If your network is large enough to require multiple management stations, separate communities should be established to prevent these events from taking place. Remember, there is no constraint on the number of SNMP communities that can be used in the network; it is only the network engineer who imposes the limits.

Denial-of-Service Attacks

Denial of service is defined as the loss of service availability either through authorized or unauthorized configuration changes. It is important to be clear about authorized and unauthorized changes. The system or application administrator who makes a configuration change as part of his job and causes a loss of service has the same impact as the attacker who executes a program to cause the loss of service remotely.

A key problem with SNMP is the ability to change the configuration of the system causing the service outage, or to change the SNMP configuration and imitate a denial of service as reported by the monitoring station. In either situation, someone has to review and possibly correct the configuration problem, regardless of the cause. This has a cost to the company, even if an authorized person made the change.

The Impact of CERT CA-2002-03

Most equipment manufacturers, enterprises, and individuals felt the impact of the CERT advisory issued by the Carnegie Mellon Software Engineering Institute (CM-SEI) Computer Emergency Response Team Coordination Center (CERT-CC). The advisory was issued after the Oulu University Secure Programming Group conducted a very thorough analysis of the message-handling capabilities of SNMP Version 1. While the advisory is specifically for SNMP Version 1, most SNMP implementations use the same program code for decoding the PDU, potentially affecting all SNMP versions.

The primary issues noted in the advisory as it affects SNMP involve the potential for unauthorized privileged access, denial-of-service attacks, or other unstable behavior. Specifically, the work performed by Oulu University found problems with decoding trap messages received by the SNMP management station or requests received by the SNMP agent on the network device.

It was also identified that some of the vulnerabilities found in the SNMP implementation did not require the correct community string. Consequently, vendors have been issuing patches for their SNMP implementations; but more importantly, enterprises have been testing for vulnerabilities within their networks.

The vulnerabilities in code, which has been in use for decades, will cost developers millions of dollars for new development activities to remove the vulnerabilities, verify them, and release patches. The users of those products will also spend millions of dollars on patching and implementing other controls to limit the potential exposures.

Many of the recommendations provided by CERT for addressing the problem are solutions for the common security problems when using SNMP. The recommendations provided by CERT can be considered common sense, because SNMP should be treated as a network service:

- *Disable SNMP.* If the device in question is not monitored using SNMP, it is likely safe to disable the service. Remember, if you are monitoring the device and disable SNMP in error, your management station will report the device as down.
- *Implement perimeter network filtering.* Most enterprises should filter inbound SNMP requests from external networks to prevent unauthorized individuals or organizations from retrieving SNMP information about your network devices. Sufficient information exists in the SNMP data to provide a good view of how to attack your enterprise. Secondly, outbound filtering should be applied to prevent SNMP requests from leaving your network and being directed to another enterprise. The obvious exceptions here are if you are monitoring another network outside yours, or if an external organization is providing SNMP-based monitoring systems for your network.
- *Implement authorized SNMP host filtering.* Not every user who wants to should be able to issue SNMP queries to the network devices. Consequently, filters can be installed in the network devices such as routers and switches to limit the source and destination addresses for SNMP requests. Additionally, the SNMP configuration of the agent should include the appropriate details to limit the authorized SNMP management and trap stations.
- *Change default community strings.* A major problem in most enterprises, the default community strings of public and private should be changed to a complex string; and knowledge of that string should be limited to as few people as possible.
- *Create a separate management network.* This can be a long, involved, and expensive process that many enterprises do not undertake. A separate management network keeps connectivity to the network devices even when there is a failure on the network portion. However, it requires a completely separate infrastructure, making it expensive to implement and difficult to retrofit. If you are building a new network, or have an existing network with critical operational requirements, a separate management network is highly advisable.

The recommendations identified here should be implemented by many enterprises, even if all their network devices have the latest patches implemented. Implementing these techniques for other network protocols and services in addition to SNMP can greatly reduce the risk of unauthorized network access and data loss.

Summary

The goal of SNMP is to provide a simple yet powerful mechanism to change the configuration and monitor the state and availability of the systems and network devices. However, the nature of SNMP, as with other network protocols, also exposes it to attack and improper use by network managers, system administrators, and security personnel.

Understanding the basics of SNMP and the major security issues affecting its use as discussed here helps the security manager communicate concerns about network design and implementation with the network manager or network engineer.

Acknowledgments

The author thanks Cathy Buchanan of Nortel Network's Internet Engineering team for her editorial and technical clarifications.

And thanks to Mignona Cote, my friend and colleague, for her continued support and ideas. Her assistance continues to expand my vision and provides challenges on a daily basis.

References

Internet Engineering Task Force (IETF) Request for Comments (RFC) documents:

RFC-1089 SNMP over Ethernet

RFC-1157 SNMP over Ethernet

RFC-1187 Bulk Table Retrieval with the SNMP

RFC-1215 Convention for Defining Traps for Use with the SNMP

RFC-1227 SNMP MUX Protocol and MIB

RFC-1228 SNMP-DPI: Simple Network Management Protocol Distributed Program

RFC-1270 SNMP Communications Services

RFC-1303 A Convention for Describing SNMP-Based Agents

RFC-1351 SNMP Administrative Model

RFC-1352 SNMP Security Protocols

RFC-1353 Definitions of Managed Objects for Administration of SNMP

RFC-1381 SNMP MIB Extension for X.25 LAPB

RFC-1382 SNMP MIB Extension for the X.25 Packet Layer

RFC-1418 SNMP over OSI

RFC-1419 SNMP over AppleTalk

RFC-1420 SNMP over IPX

RFC-1461 SNMP MIB Extension for Multiprotocol Interconnect over X.25

RFC-1503 Algorithms for Automating Administration in SNMPv2 Managers

RFC-1901 Introduction to Community-Based SNMPv2

RFC-1909 An Administrative Infrastructure for SNMPv2

RFC-1910 User-Based Security Model for SNMPv2

RFC-2011 SNMPv2 Management Information Base for the Internet Protocol

RFC-2012 SNMPv2 Management Information Base for the Transmission Control Protocol

RFC-2013 SNMPv2 Management Information Base for the User Datagram Protocol

RFC-2089 V2ToV1 Mapping SNMPv2 onto SNMPv1 within a Bi-Lingual SNMP Agent

RFC-2273 SNMPv3 Applications

RFC-2571 An Architecture for Describing SNMP Management Frameworks
RFC-2573 SNMP Applications
RFC-2742 Definitions of Managed Objects for Extensible SNMP Agents
RFC-2962 An SNMP Application-Level Gateway for Payload Address
CERT Advisory CA-2002–03

24

Network and Telecommunications Media: Security from the Ground Up

Samuel Chun, CISSP

Introduction

One of the most challenging aspects of understanding telecommunications and network security is the overwhelming number of resources that are required to maintain it. Making telecommunications and networking "work" involves millions of miles of cabling, thousands of communications devices, and an uncounted number of people all working together to deliver information among devices. Whether the information is a word-processing document, an e-mail message, an Internet phone call, or an ATM transaction, it starts from a device and traverses media that are largely unknown to most people. The focus of this chapter is on those media that carry the information. From the thousands of miles of optical cable that run deep beneath the oceans to connect continents, to the inexpensive "patch" cables that are sold in hardware stores, to home users, each has an important role to play and each has an implication in securing a network environment from one end to the other.

A later chapter introduces the Open System Interconnect (OSI) model to present a conceptual view of how computers communicate with each other over a network. Although the OSI model is only a framework, it is the accepted architectural reference model for all computer communications. The OSI model layers network communications in a logical hierarchical format that is easy to understand and apply. At the lowest layer, the physical layer, data is converted into patterns of electrical voltage changes and transferred in the appropriate medium — cabling. Without this fundamental function taking place at the lowest and earliest layer, network and telecommunication traffic would not be possible. It is a wonder why, then, the cabling and transport medium is one of the least emphasized aspects of network security. Its function is vital, and vulnerabilities and weaknesses of a given network's cabling infrastructure can potentially impact all aspects of the Availability, Integrity, and Confidentiality triad.

Cabling Issues

Before discussing the various types of wiring and transport media, it is important to review some of the more important issues involving cabling that also impact security. Some of the issues are a result of the nature of the materials used in manufacturing, while others deal with the matter in which they are produced. All of these factors should be considered when deploying a new cable infrastructure and certainly when evaluating the security posture of a given network at its lowest component level.

Maximum Transmission Speed

Depending on the wiring and network equipment that is used, a wide array of transmission speeds can be accomplished in a network. From the 16 Mbps that can be supported on Category 4 unshielded twisted pair (UTP) cabling, to the 10 Gbps that can be run on single mode fiber (SMF), the nature of the wiring can determine the maximum transmission speed a network can support. When a service or application's transmission requirements exceed the supported limit, system availability or data integrity issues may occur. A typical example of this is the potential for synchronization problems or dropped video frames in video conferencing and its high bandwidth requirements. Wiring infrastructure based on 2-Mbps thin-net coaxial cable will not support it, while fiber and Category 5 UTP with its support for 100-Mbps transmission speeds will.

Susceptibility to Interference

Different media types have varying levels of susceptibility to ambient environmental interference. Consequently, different types of wiring are generally, but not always, implemented for specific situations. For example, optical fiber cables, which transmit light waves, are used as the *de facto* standard in connecting buildings or geographical regions, due their to immunity from interference caused by electricity, light, heat, and moisture. Copper cable-based wiring, on the other hand, is vulnerable to a variety of environmental factors because its function is based on electrical conduction over a strand (or multiple strands) of wire.

There are three specific interference issues that are important to consider when selecting an appropriate wiring medium: attenuation, crosstalk, and noise.

Attenuation

Attenuation is the degradation of any signal resulting from travel over long distances. It is often referred to as signal "loss," and occurs as signal power, measured as voltage for traditional copper cabling and light intensity for fiber, degrades over distance due to resistance in the medium. Regardless of medium or signal, attenuation is the measure of signal loss per distance unit.

Attenuation in networking is generally measured in decibels of signal loss per foot, kilometer, or mile. Attenuation is a bigger problem for higher frequency signals. For example, a wireless Gigabit Ethernet connection transmitting at 38 GHz will experience more attenuation than one running at 18 GHz over the same distance. Consequently, there are specific cable length standards for different networking speeds, media, and technologies. Generally, less attenuation means greater distances and clearer signals between network devices and components. When any cabling is installed for a network, regardless of the type, it should be thoroughly tested for the effects of attenuation.

Crosstalk

The phenomenon of hearing other voice conversations during a telephone conversation is a classic example of crosstalk. Crosstalk, as the name implies, is the interference caused by one channel during transmission to another nearby channel. Crosstalk in a network medium could result in packet collisions and retransmissions that can impact performance and reliability. Reducing crosstalk results in better cable efficiency. A common method for reducing crosstalk is to sheathe the metal wire with insulating materials. For example, shielded twisted pair (STP) cables are less likely than UTP cables to experience crosstalk.

Noise

The broadest definition of noise is the negative effect of environmental conditions on a transport medium's signal. Noise can result from numerous causes, including heat or cold, weather, light, electricity, and ionizing radiation. From common sources such as electrical appliances, fluorescent lights, or x-ray machines, to powerful environmental events such as rain or fog, numerous conditions can influence a given network transmission medium's ability to send a signal effectively. One of the best examples of environmental noise influencing network availability is the effect of inclement weather on microwave-based WAN connections. Unlike the postal service, wireless networks can be brought to a standstill by rain, sleet, or snow.

Maximum Distance

Distance plays a big role in the network media selection. The distance that the cable will need to "run" before it is attached to another device can amplify attenuation, noise, and crosstalk. There are standards that specify

how long different types of cables can be specifically run before a repeater is necessary to boost the signal. The maximum distance between repeaters can vary with some media that can only span hundreds of meters, while some, such as microwave, can span miles. The maximum required distances between physical connections can dictate the type of media that needs to be used.

Susceptibility to Intrusion

One of the factors to consider when selecting a medium for a network is its susceptibility to intrusion. Some transmission media are more of a target for eavesdropping than others, just by the nature of the material used for manufacturing. Others are, by design or as a side effect, more difficult to "tap." For example, unshielded twisted-pair cables are easy to tap into and also emanate electrical current. Conversely, optical fiber does not emanate at all and is almost impossible to tap. If confidentiality is a big factor in a network, then it will help determine which media can be used best in that particular environment.

Manageability, Construction, and Cost

Overall cost often plays a major role in choosing network media. Many factors influence the cost of media: the type of materials used, quality of construction, and ease of handling all play roles in the overall cost of ownership of a particular networking media deployment. In addition, there are also indirect costs that should be considered. For example, when optical fiber is used as the networking transmission medium, there are greater costs associated from a networking equipment standpoint than an otherwise identical network made of copper cabling. Fiber network cards, switch and router modules, and media testing equipment tend to be much more expensive than their copper counterparts. All these cost factors — both direct and indirect — should be considered during the evaluation process.

Coaxial Copper Cable

Background and Common Uses

Coaxial copper cable, invented prior to World War II, is perhaps the oldest wire-based communications medium. Before the advent and explosive growth of UTP cabling, coaxial cabling was commonly used for radio antennae, cable TVs, and LAN applications. The cable is referred to as "coaxial" because it contains a thick, conductive metal wire at the center that is surrounded by meshed or braided metal shield along the same parallel axis. The thick wire in the center of the cable is generally separated from the metal shield by PVC insulation. The meshed metal shield that surrounds the core copper wire insulates the cable from interference such as crosstalk and noise. Compared to UTP, coaxial cable can transmit signals greater distances and at a higher bandwidth. Due to these factors, "coax" was commonly deployed in a variety of different applications. By the mid to late 1980s, coax cable was found almost everywhere — in homes as wiring for cable TVs and radios, as LAN cabling for business and government (especially school systems), and by telephone companies to connect their poles. However, during the 1990s, the inexpensive UTP gained favor in almost all LAN-based installations. Today, coaxial cabling is rarely seen in LAN applications; however, it continues to be popular as a medium for high-speed broadband communications such as cable TVs.

Categories and Standards

There are two main types of coaxial cabling. The 75-ohm cable is the most familiar to the average person because it is commonly used in homes to connect AM/FM radios to antennae and TV sets to cable boxes. The 75-ohm coaxial cable is unique in that, in addition to analog signals, it can also transmit high-speed digital signals. Consequently, it is commonly used in digital multimedia transmissions (e.g., digital cable TVs) and broadband Internet connections (mainly cable modems) in many people's homes.

The 50-ohm coaxial copper cable is the other type of coaxial cabling. It is most commonly used for LAN purposes. There are also two types of 50-ohm coaxial cables used in networking.

Thin Coax, Also Known as "Thinnet" or 10Base2 Specification

RG58 is a 52-ohm, low-impedance copper coaxial cable that can carry a 10-Mb Ethernet signal for approximately 200 meters (specifically, 185 meters) before requiring a repeater. Thin coax was typically deployed in a bus topology fashion in many networks, especially in educational environments. Thinnet "daisy chains" were known as "cheapernets" due to their low cost and low reliability. Thinnet Ethernet and AppleTalk networks were popular network configurations during the 1980s. However, Thinnet quickly lost favor to the inexpensive, reliable star topology of hub-based UTP networks during the 1990s.

Thick Coax, Also Known as "Thicknet" or 10Base5 Specification

Thicknet can carry a 10-Mb Ethernet signal for 500 meters. The rigid RG8 and RG11 cables, as the name implies, are thicker than Thinnet due to its larger core and extra layers of insulation. Thicknet was commonly used to connect bus-based networks across long distances (due to its thick insulation) and had the unique ability to allow for a connection to be added while signals were being transmitted — "vampire taps."

Strengths and Capabilities

Compared to UTP, coaxial cables can transmit signals at higher bandwidths and over longer distances without requiring the signal to be boosted by a repeater. The wire braid shielding, the insulation, and thick plastic jacket protect the cable from electromagnetic interference (EMI) and environmental effects such as heat and moisture. In addition, the insulation makes electronic eavesdropping more difficult because electric emanations are also minimized.

Vulnerabilities and Weaknesses

The two drawbacks to using coaxial cabling for networking are its difficulty in installation and its cost. The elements that make coax so effective — the insulation and thick core — also make it difficult to deploy and relatively expensive compared to UTP. In addition, the widespread proliferation of network hubs and switches have negated the distance advantages of coax cables. Manufacturers of networking equipment have wholeheartedly supported the widespread deployment of UTP by making coaxial cable-based networking equipment difficult to find and procure. Currently, it is nearly impossible to find networking infrastructure equipment such as switches, hubs, or even network cards that are based on a coaxial cable connection.

Future Growth

The use of coaxial cables for general-purpose networking is likely to become an anomaly within the next five to ten years. The latest standards and products for high-speed networking are increasingly focusing on fiber- and UTP-based networks. Most large organizations have already migrated away from coax, and, as time progresses, the likelihood of encountering 10Base2 or 10Base5 networks will become increasingly slim. However, the tried-and-true 75-ohm "home" coaxial cables that can transmit both analog and digital signals will continue to play a strong role in delivering high-speed data to peoples' homes. The use of 75-ohm copper cable in cable boxes, and increasingly with cable modems, ensures that the coaxial copper cable medium will continue to play a role, even if only a small one, in the future of networking.

Unshielded Twisted Pair (UTP) Cable

Background and Common Uses

Unshielded twist pair (UTP) cable is the most commonly installed networking medium. It supports very high bandwidths, is inexpensive, flexible, easy to manage, and can be used in a variety of networking topologies. 10 Mbps Ethernet, 100 Mbps Fast Ethernet, 4/16 Mbps Token Ring, 100 Mb FDDI over copper, and 1000 Mbps Gigabit Ethernet can all be run over UTP cabling. UTP cable and its properties are well known and are utilized in almost all network environments.

Categories and Standards

As the name implies, UTP cables have four pairs of conductive wires inside the protective jacket, tightly twisted in pairs. UTP cables do not have any shielding other than the insulation of the copper wires and the outer plastic jacket. The most important properties of UTP cabling are derived from the characteristic twisting of the pairs of cables. These twists of the conductive material help to eliminate interference and minimize attenuation. The tighter the twisting per inch, the higher the supported maximum bandwidth and the greater the cost per foot. Because there are different levels of twisting, conductive material, and insulation, the Electronic Industry Association/Telecommunications Industry Association, also known as EIA/TIA, has established EIA/TIA 568 Commercial Building Wire Standard for UTP cabling and rated the categories of wire:

- Category 1:
  - — Maximum rated speed: generally less than 1 Mbps (1 MHz)
  - — Pairs and twists per foot: generally two pairs; may or may not be twisted
  - — Common use: analog phone lines and ISDN; not used for data
- Category 2:
  - — Maximum rated speed: 4 Mbps (10 MHz)
  - — Pairs and twists per foot: four pairs; generally two or three twists per foot
  - — Common use: analog phone lines, T-1 lines, ISDN, IBM Token Ring, ARCNET
- Category 3:
  - — Maximum rated speed: 10 Mbps (16 MHz)
  - — Pairs and twists per foot: four pairs; three twists per foot
  - — Common use: 10Baset-T, 4-Mbps Token Ring
- Category 4:
  - — Maximum rated speed: 20 Mbps (20 MHz)
  - — Pairs and twists per foot: four pairs; five or six twists per foot
  - — Common use: 10Base-T, 100Base-T4, 100VG-AnyLAN, 16-Mbps Token Ring
- Category 5:
  - — Maximum rated speed: 100 Mbps (100 MHz)
  - — Pairs and twists per foot: four pairs, 36–48 twists per foot
  - — Common use: 100Base-T4, 100Base-TX, FDDI, and 155-Mbps ATM
- Category 5e:
  - — Maximum rated speed: 1 Gbps (350 MHz)
  - — Pairs and twists per foot: four pairs; 36–48 twists per foot
  - — Common use: 100Base-T4, 100Base-TX, 1000Base-TX, 155-Mbps ATM
- Proposed Category 6:
  - — Maximum rated speed: 300 Mbps (Unknown; vendors manufacturing 400 MHz)
  - — Pairs and twists per foot: four pairs; twists per foot not specified
  - — Common use: anticipated to be used in high-speed environments, especially 1000-Base-TX and ATM
- Proposed Category 7:
  - — Maximum rated speed: 600 Mbps (600 Mz)
  - — Pairs and twists per foot: four pairs; twists per foot not specified
  - — Common use: anticipated to be used in high-speed environments. Cat 7/Class F is anticipated to have a completely different plug/interface design.

Strengths and Capabilities

UTP cabling in all of its different flavors has become ubiquitous in networking. It is difficult to find a networking environment where UTP, especially Category 5 UTP cabling and "patch" cables, is not used. It is relatively inexpensive per foot, easy to install and terminate, and has broad support from networking equipment vendors. Because it is able to support multiple networking topologies, protocols, and speeds, it has rapidly replaced most cabling, other than high-speed fiber, for network use.

Vulnerabilities and Weaknesses

UTP cabling's drawbacks are based on its lack of shielding. It is flimsy and easy to cut and damage, and susceptible to interference and attenuation due to its lack of shielding and use of copper as a conductor. Because data transmission is based on electrical conduction (without shielding), it radiates energy that potentially can be intercepted by intruders. The easy manageability of UTP cabling also allows it to be easily tapped into. Consequently, highly secure environments are more likely to use optical fiber for their media needs.

Future Growth

UTP cabling, without a doubt, will continue to play a major role in networking. Its flexibility in its ability to support different protocols and speeds allows its use in a variety of environments. In addition, its low cost is a big plus in selecting media. Although the latest bandwidth and speed advancements are always introduced through fiber, there is always an initiative that quickly follows to support it on copper — and mainly UTP copper cabling. This was the case when Fast Ethernet was devised and was certainly the case recently when Gigabit Ethernet was introduced. Although Gigabit Ethernet was supported on fiber first, the development of CAT 5E and 6 cables quickly followed, with networking companies offering to switch modules and NIC cards very quickly. This trend is likely to continue with further advances in networking with CAT 6 and CAT 7 cables offering even higher maximum transmissions speeds to feed the growing appetite for data transmission bandwidth.

Shielded Twisted-Pair (STP) Cable

Background and Common Uses

Shielded twisted-pair (STP) cabling was initially developed by IBM for its Token Ring networks during the 1980s. The original Type 1 STP cable was a bulky, shielded cable with two pairs of conductive wire that was commonly deployed with Token Ring networks. The Token Ring STP combination offered a 16-Mbps deterministic network topology that was ideal for networks that needed the extra bandwidth, because Ethernet 10Base2 and 10Base5 coaxial were the only competitors during the early years. With the development of inexpensive UTP and the ever-increasing bandwidth that it supports, Type 1 STP with its one topology and one-speed support has been deemed almost obsolete in networking.

A new type of STP, which is basically a Category 5 UTP cable wrapped in shielding, has recently been introduced and holds some promise for specific network environments.

Categories and Standards

The original Type 1 STP cable was distinctive in its presentation. It was thick due to the braided shielding that surrounds both pairs of 150-ohm conductive copper core. Its end connectors were large (compared to modern-day RJ-45 caps of UTP) square blocks that plugged into network devices called multi-station access units (MAUs). Many engineers with Token Ring/Type 1 cable experience will recall the familiar "clicks" that preceded a network connection on the MAUs. Type 1 cables were rated up to 16 Mbps and were eventually replaced by Category 3, 4, and 5 cables for Token Ring.

The newer STP cable is similar to Category 5 UTP cable in that it has four pairs of tightly wound copper wire. However, a thin layer of aluminum foil shielding surrounds all four pairs of the cable in lieu of the heavy braided layers of Type 1. There is also metal in the plugs themselves to allow grounding and additional shielding. The new STP is referred to as screened twisted pair (ScTP) or foil twisted pair (FTP) and is more flexible, lightweight, and easier to deploy than Type 1. Currently, there are no standards for this new type of STP

cabling, but most vendors follow the EIA/TIA 568 UTP Category 5 standard that allows for 100-Mbps transmissions.

Strengths and Capabilities

The strength of STP cable is in its shielding and insulation. The braided aluminum/copper mesh that surrounds the twisted pairs allows the cable to resist noise and electromagnetic interference (EMI). Although the old Type 1 cables are no longer being actively deployed, the new STP cables are being manufactured and marketed for high-interference environments. The newer STP cables offer some of the advantages of UTP cabling — high-bandwidth, multi-topology support, and lower cost — and have the added benefit of resistance to EMI. Environments such as medical facilities, airports, and manufacturing plants can derive benefits from using ScTP/FTP.

Vulnerabilities and Weaknesses

The weaknesses of the Type 1 STP medium are well documented. Type 1 is bulky, difficult to deploy, slow, and only supports one network topology. It is not surprising that Type 1 STP cables have been almost forgotten for general-purpose networking. Although the new ScTP and FTP cables show great promise, they still have some of the limitations based on the disadvantages of metal shielding. All STP cabling systems require careful emphasis on grounding because an STP cable that has not been grounded on both ends offers little resistance against EMI. In addition, unlike UTP, the cables must be deployed with great care so that none of the shielding elements, such as the connectors or the cable itself, are damaged. For STP cables to work, both grounding and shielding integrity must be maintained during installation, or the benefits of using shielded cables are lost.

Future Growth

The future of STP media is uncertain. The Type 1 cabling so common during the 1980s has been all but abandoned during the "Fast Ethernet" rush of the 1990s. The new lightweight, flexible STP cables, drawing on the strength of the characteristics of UTP cabling, have yet to be deployed in mass due to their narrow marketing focus and high overall cost. However, renewed focus in the United States and abroad on ensuring that cabling, regardless of type, be electromagnetically compatible (EMC) with its environment holds some promise for the growth of STP.

Optical Fiber Cable

Background and Common Uses

At the time of writing this chapter (March 2003), Stanford University's Linear Accelerator Center set a new speed record for transmitting data on the Internet, by sending 6.7 gigabytes of data across 6800 miles in less than 60 seconds. That technological marvel is equivalent to sending all of the data on the two-DVD set of "Gone with the Wind" from New York City, in the United States, to Tokyo, Japan, in about the time it takes to read this paragraph. This amazing accomplishment is part of the continuing evolution of the networking technologies that are being used by millions of people every day. The common network component that has fueled this growth in data transmission speed and volume on the Internet has been the increased reliance on hair-thin strands of silica glass — better known optical fiber cable.

The idea of transmitting data with light dates back to the 1800s with Alexander Graham Bell having the first recorded patent of a light-data transmitting device — his Photophone — in 1890. However, real advances in transmitting light through strands of glass fiber did not occur until after World War II. The advent of semiconductor diode lasers that can be used at room temperature and advances in the manufacturing processes of optical fiber cables in the early 1980s set the stage for the first large-scale commercial use of optical fiber cables by AT&T. By the mid to late 1980s, fiber was being laid across oceans, with the first being the English Channel; and by the 1990s, fiber-optic cables were beginning to be widely used in local area network environments, primarily as backbones for office networks.

Today, with the exponential advances in network speeds, optical fiber is the *de facto* standard for connecting wide area and local area networks. Two general types of fiber cable — single mode (SMF) and multimode

(MMF) — are commonly used to connect cities, buildings, floors, departments, and even homes. Fiber-optic cable's inherent resistance to attenuation (allowing for long distances and speeds), noise, and EMI make it a perfect choice for transmitting data.

Categories and Standards

Optical fiber refers to the medium that allows for the transmission of information via light. Fiber cable consists of a very clear, thin filament of glass or plastic that is surrounded by a refractive sheath called "cladding." The core, or axial, part of fiber-optic cable is the intended area for transmission, while the cladding is intended to "bounce" errant light beams back into the center. The core has a refractive index approximately 0.5 percent higher than that of the surrounding cladding so that errant light rays transmitted at shallow angles to the cladding are reflected back into the center core. This transmitting "center," made of a thin strand of glass, generally needs to be protected because, unlike copper metal wire, it is brittle and fragile. Often, the cladded core is coated with plastic, and Kevlar fibers are embedded around the outside to give it strength. The outer insulation is generally made of PVC or Teflon.

There are three specific types of fiber cables, and each has its specific uses.

Step-Index Multimode Fiber

Step-index multimode fiber has a relatively thick center core and is almost never used for networking. It has a thick, 100-micron core surrounded by cladding that allows light rays to reflect randomly, which results in the light rays arriving at different times at the receiver, resulting in what is known as modal dispersion. Consequently, information can only be transmitted over limited distances. Step-index multimode fiber is most often used in medical instruments.

Graded-Index Multimode Fiber or Multimode Fiber (MMF)

Graded-index multimode fiber, or MMF, is likely the most well-known fiber medium to most network administrators and engineers. MMF cables are commonly used in local network backbones to connect floors and departments between networking components such as switches and hubs. The graded-index MMF has the characteristic of the refractive index between the cladding and core changing gradually. Consequently, multiple light rays that traverse the core do not "bounce" off the cladding in a random manner. Rather, the light refracts off the core in a helical fashion, allowing for most of the beams to arrive at the receiver at about the same time. The end result is that the light rays arrive less dispersed. MMF fiber, although designed to minimize modal dispersion, is still best suited for shorter distances compared to single-mode fiber, which can transmit data for miles. Although MMF fiber is limited as to the distances over which it can be used, it is still able to transmit far greater distances than traditional copper wires. Consequently, it is widely used and widely supported by networking equipment companies to connect network backbones in traditionally UTP-cabling-based environments.

Single-Mode Fiber (SMF)

Single-mode fiber (SMF) has the narrowest core of all fiber cables. The extremely thin core, generally less than 10 microns in diameter, is designed to transmit light parallel to the axis of the core in a monomode fashion, attempting to eliminate modal dispersion. This single-beam mode of transmission permits data transmission over far greater distances. SMF is generally used to connect distant points and therefore is commonly used by telecommunications companies. In addition, SMF is increasingly being used by cable television companies to deliver digital cable as well as broadband data connections to homes. However, SMF use in LAN applications is generally not common due to its high cost and the limited support for SMF components in network equipment intended for LANs.

Strengths and Capabilities

Optical fiber media have distinct advantages due to their use of light instead of electrical impulses through a metal conductor. Light, and consequently fiber-based media, is highly resistant to attenuation, noise, and EMI. Consequently, fiber-based connections can traverse distances much farther and transmit more data than wire-based media. Fiber is perfect for high-bandwidth applications such as multimedia and video conferencing. In

addition, because no electrical charges travel across it, it does not emanate any data, thereby providing security that no other media can offer. Its fragility also offers protection from intruders in that it is very difficult to tap into fiber-based networks without detection. It is commonly accepted that fiber-based networks run farther, faster, and more securely than any other available medium.

Vulnerabilities and Weaknesses

Unfortunately, fiber has some drawbacks that prevent it from being used in almost all situations. Because fiber is made of glass or plastic, it is more difficult to manufacture and work with than copper. It is not malleable, is difficult to terminate and install, and can be more easily damaged than wire-based media. In LAN-based environments, it is common for administrators and engineers to "crimp" or custom-create cable lengths in data centers and server rooms for use with UTP cabling. This is almost never the case with fiber, which is generally purchased in specific lengths.

In summary, although fiber has some distinct advantages, it has a very high cost of ownership. It is expensive to purchase, install, and maintain a fiber-based infrastructure. Even the network components that support fiber, such as router and switch modules, fiber-based NIC cards, etc., are much more expensive and rare than their UTP-based counterparts. Although prices for all types of PC and networking equipment have decreased dramatically in the past seven or eight years, the difference in support costs between fiber and copper media is not expected change in the future.

Future Growth

Most networking experts agree that Internet traffic has, on average, doubled each year since the mid-1980s. With the increased availability of high-speed network connections in people's homes and the increases in application demand for bandwidth, it is difficult to imagine being able to support these ever-increasing needs without the availability of fiber-optic media. Although fiber and its infrastructure are expensive, it will without a doubt, remain a critical component of network technologies with its seemingly endless potential for increased speeds and bandwidths. Millions of miles of optical fiber are being laid throughout the world each year by governments and private companies, and this trend can be expected to continue to grow as the world's needs for higher bandwidths increase each year.

Wireless Media

Background and Common Uses

When most people think of wireless technologies, they often seem to forget that wireless was developed more than a century ago by Guglielmo Marconi. Before the advent of "Wi-Fi" (Wireless Fidelity) networking, satellites, and cell phones, the good old-fashioned radio had been sending information through the wireless medium for decades. Recently, wireless has been introduced in almost every home with remote control TVs, garage door openers, and now even wireless appliances and PCs. The extension of attempting to use wireless technology into the area of PC and network computing was an easy one with obvious benefits. The topic of wireless technologies is broad and is rich with information; this section focuses on an overview of three specific, commonly available and well-known wireless network technologies.

First, wireless local area networks (WLANs), based on the IEEE 802.11 standard and now available in many offices, homes, coffee shops and restaurants will be reviewed. Then we discuss the extension of wireless LANs into metropolitan areas (WMANs) will be discussed, followed by a brief introduction to the new wireless arena intended to cover an extremely small area known as the personal area network (WPANs).

Categories and Standards

Wireless Local Area Network

The IEEE 802.11 standard, also known as "Wi-Fi," is specifically geared for wireless LANs. Almost all wireless LANs are based on 802.11 and are being increasingly installed in offices, homes, airports, and even in fast-food restaurants. All "Wi-Fi" networks have transmitting antennae known as access points that PCs connect

to. The access point is generally connected to a traditional wired network LAN that allows access to the Internet via an ISP or to local resources such as file servers and printers. The laptops and PCs that connect to the access point must also have a "Wi-Fi" antenna. Although the specific components of all 802.11 wireless networks are the same, there are three different standards of 802.11 that are commonly seen. Each has its different strengths and uses.

IEEE 802.11a.

The 802.11a-based WLANs transmit data at the unlicensed frequency of 5 GHz. This high-frequency WLAN allows a maximum speed of 54 Mbps with fairly good encryption of the data transmitted. It also is able to handle more concurrent users and connections than 802.11b. Unfortunately, 802.11a has a limited effective range and is generally used in line-of-sight situations. It is ideal for office environments with cubicles and conference rooms where the access points are mounted in the ceiling. It is also more expensive to deploy than 802.11b; consequently, 802.11a WLANs do not have a large install base.

IEEE 802.11b.

The 802.11b WLANs use the unlicensed 2.4-GHz frequency range (which is currently used by common appliances such as cordless phones) and has an effective range of up to 100 yards. It was the first low-cost wireless LAN technology made available and has a comparatively large install base. The 802.11b-based networks generally transmit at speeds of 11 Mbps, but some network vendors use data compression algorithms to be able to offer maximum transmission speeds of 22 Mbps. The 802.11b standard allows for much greater distances than 802.11a (approximately 100 yards) and is cheaper to deploy. Consequently, it has a large install base in public and home use.

IEEE 802.11g.

This new proposed standard works in the same 2.4-GHz frequency band as 802.11b but offers a maximum speed of 54 Gbps. Because it works in the same frequency range as 802.11b, it is able to support existing 802.11b installations, which is a big plus. Vendors have already released networking devices based on the proposed 802.11g standard, and its performance capabilities are promising. In addition, 802.11g network devices are even less expensive than 802.11b devices. With the promise of better performance for less cost, 802.11g will likely replace 802.11b, and possibly even 802.11a.

Wireless Metropolitan Area Network (WMAN)

"Wi-Fi" networks in actual use are confined to a relatively small area of approximately 300 feet. However, there are obvious advantages to being free from having to rely on fiber- or metal-based media that frequently make up for the limitations of short available "Wi-Fi" ranges. The IEEE 802 committee set up the 802.16 working group in 1999 to develop a standard for wireless metropolitan broadband access. There were three working groups of 802.16: 802.16.1 through 802.16.3. The 802.16.1 has shown the most potential and interest because it focuses on a readily available frequency range. The 802.16.1 WMAN infrastructure relies on a core network provider, such as the telephone company, offering wireless services to subscribers who will access the core network through their fixed antennae. In effect, subscribers in homes and offices will access the core switching center through base stations and repeaters. The connections will be provided through dynamic wireless channels ranging from 2 Mbps to 155 Mbps via an 802.16.1-based frequency range of 10 GHz to 66 GHz.

Wireless Personal Area Network (WPAN)

The personal area network (PAN) is a low-power, short-range, wireless two-way connection that connects personal devices such as PDAs, cell phones, camcorders, PC peripherals, and home appliances. The Bluetooth specification with its associated technology is the front-runner in providing personal wireless connectivity to users. It uses the unlicensed 2.4-GHz frequency with signal hopping to provide an interference-resistant connection for up to seven concurrent devices. Typically, a small Bluetooth network will be set up with a common authentication scheme and encryption so that other Bluetooth networks will not be able to connect automatically.

The Bluetooth standard has been around for many years. The Bluetooth Special Interest Group (SIG), a consortium of vendors that intends to develop and promote Bluetooth products, agreed on the third and current iteration Version 1.1 in 1999. Since that time, a host of new products has been introduced and new ones are planned — from PC peripherals to microwave ovens, cell phones, and even washers and dryers — all based on the Bluetooth PAN standard.

Strengths and Capabilities

Wireless networking has the obvious advantage of freeing one from the need to run cabling. The medium through which the communication travels is publicly available and free. Wireless networks allow for truly mobile computing, with the greatest benefit for roving laptop users. Wireless "hotspots" are springing up in many places, allowing Internet access for a growing number of users. Coupled with VPN technologies and wireless networking, users can extend the "office" environment beyond home networks and corporate offices.

Vulnerabilities and Weaknesses

The freedom of mobility that wireless networks provide their users also has its limitations. Wireless networking has not been widely deployed due to several issues. Wireless networks are slower than traditional cabled systems, are more expensive to deploy, and are susceptible to interference from environmental conditions such as weather and EMI.

However, the most important vulnerability that inhibits wireless networking from becoming more widely used is its lack of security. Because wireless uses a public medium in which data is transmitted, it is susceptible to "snooping" and eavesdropping. In the most widespread LAN application of wireless (i.e., 802.11b), networks are generally secured using LAN authentication by means of the wireless adapter's hardware MAC address. This is not really secure because MAC addresses can easily be falsified. Other techniques of encrypting the data using shared keys on the access point and receiver are available but not practical in large enterprise organizations due to difficulty in managing large numbers of keys. Even protocols intended to assist with wireless key management, such as Wired Equivalent Privacy (WEP), are cumbersome because key distribution and updates must be done in a secure medium outside of 802.11. In addition, although WEP encrypts the data that is being transmitted through the airwaves (via the RC4 algorithm), it is not completely secure. WEP can be easily cracked by anyone who has extensive knowledge of network sniffers.

Future Growth

It is clear that wireless networking holds a promising future for specific applications. The proliferation of 802.11b/802.11g-based "hot spots" grants greater freedom to casual users who need access to the Internet from a variety of locations. In addition, the relative ease of deploying wireless in home environments, as opposed to wiring cable, provides a niche market for networking companies. For enterprise-level environments, wireless networking will likely only play a small role due to its limitation in performance, lack of security, and high administration costs. However, for specific users and needs, such as areas in which wiring is difficult or impossible, conference room applications, mobile users, and roving service staff, there may be a natural fit for wireless.

Broadband: Digital Subscriber Line and Cable Modem

Digital Subscriber Line (DSL)

Digital Subscriber Line (DSL) is a broadband-based technology that uses existing telephone copper cabling to deliver high-speed Internet service to its subscribers. It largely depends on telephone companies, because it uses an upgraded telephone infrastructure. DSL signals are transmitted via special equipment over the existing phone lines and use frequencies that are higher than those of traditional voice traffic. A DSL filter, often referred to as a DSL modem, is used to segregate voice and data traffic on the recipient side.

DSL connections are always on, available 24 hours per day, regardless of the voice-phone traffic. It can theoretically provide up to 52-Mbps transmission under ideal conditions. It is inexpensive, and is becoming increasingly available in metropolitan areas. There are different types of DSL: the type depends on the carrier and what type is available in which area (see Exhibit 24.1).

DSL, however, does have its limitations. DSL technology relies on the carrier having the upgraded equipment, generally referred to as a Digital Subscriber Line Access Multiplexer (DSLAM) available in the area. The subscriber must be within a certain distance of the DSLAM and performance is impacted based on that distance. The further the subscriber is from the CO (central office) with the DSLAM, the less bandwidth it is able to achieve. In addition, other subtle factors, such as quality of the phone cables used in an installation, can impact

EXHIBIT 24.1 Types of DSL

Type	Max. Downstream Speed	Max. Upstream Speed	Max. Distance Central Office to Subscriber	Copper Pairs Used
Asymmetric (ADSL)	1.5–9 Mbps	16–640 Kbps	18,000 feet	1
Single-line (SDSL)	1.544 Mbps	1.544 Mbps	10,000 feet	1
High-rate (HDSL)	1.544 Mbps	1.544 Mbps	12,000 feet	2
Very-high-rate (VDSL)	13–52 Mbps	1.5–2.3 Mbps	4,500 feet	2

DSL performance. Even with these limitations, it is being widely accepted by remote and home users due to its low price and performance, which easily exceed that of dialup and ISDN connections.

Cable Modems

Cable television companies have been installing optical fiber cables for years to deliver digital-quality cable TV channels to their subscribers. The cabling infrastructure that cable companies have installed, mainly optical fiber to buildings and 75-ohm coaxial once inside, is increasingly being used to offer high-speed digital network service to the Internet. Similar to DSL, a specific cable modem is required to receive high-speed access through the same medium that cable television is received. It is capable of delivering approximately 50 Mbps, but its speeds are generally less because segments are shared among subscribers. Consequently, bandwidth can change over time for a particular subscriber because performance is based on aggregate segment usage.

There have been several different iterations of cable modem service. Initially, cable modems used various proprietary protocols so that a cable TV provider could only use a specific cable modem for service. Within the past three years, there has been a movement toward standardization so that various cable modems can be used regardless of the provider. So far, no formal body has established any specific standard, but, in general, three standards are used:

1. Digital Video Broadcasting (DVB)/Digital Audio-Video Council (DAVIC), also known as DVB-RCC: not very common, but still used in Europe.
2. *MCN/DOCSIS:* a predominately U.S. standard that almost all U.S. cable modems are based on.
3. *EuroDOCSIS:* a European standard based on DOCSIS.

In addition, the IEEE is attempting to develop its own standard, referred to as 802.14.

Cable modems have become popular because they are always on, readily available, inexpensive, and provide high bandwidth to most users. Unfortunately, cable modems are considered notoriously insecure because traffic within a cable modem segment is generally not filtered. Once a cable modem is installed, a packet sniffer can easily capture traffic that is being broadcast by other users in the segment.

Strengths and Capabilities of Broadband

Cable modem and DSL service rely on vastly different technologies to deliver the same type of service — high-speed Internet. Both are relatively inexpensive, not much more than analog dialup, and require minimal equipment for start-up. They both deliver speeds that far exceed traditional access methods, such as analog dialup and ISDN. They are also simple to use and do not require any connection procedures. Users generally leave them on continuously because they do not interfere with other services, whether TV, voice, or fax. These capabilities have encouraged both cable modem and DSL service to become ever more widespread in use. With advances in VPN technologies, they are commonly being used from homes not only to the Internet, but to

offices as well. The availability of inexpensive, high-speed service that can be used for personal and work functions has been an invaluable advancement for remote offices and telecommuters.

Vulnerabilities and Risks

Unfortunately, having high-speed Internet access that is continuously available poses risks. Cable and DSL modems are usually never turned off, and systems run without pause. In addition, residential DSL and cable modem consumers are less likely to be aware of the capabilities of and the need for a firewall. These users who are always on the Internet without protection are precisely the targets that hackers are looking for. They can scan ports, stage distributed denial-of-service (DDoS) attacks, and upload worms, viruses, and Trojan horses at any time and at very high speeds. Many residential broadband customers have become unwitting accomplices to DDoS attacks against innocent targets, due to ignorance or a lack of vigilance.

A potential vulnerability that one needs to be particularly mindful of is the use of DSL and cable modems with VPN connections into enterprise environments. The benefit of having high-speed, secure access from home into the office network is a wonderful productivity tool. However, having fast access to your corporate network through the Internet poses a risk to the corporate network. Imagine a scenario in which a hacker uploads a virus, a worm, or a Trojan horse to a PC with a cable modem that also has established a VPN tunnel to a corporate network. The "pathogen" is free to travel through the VPN tunnel into the corporate network and attack it from the inside. This particular type of risk is magnified in environments that allow VPNs to perform "split-tunneling." Split-tunnels allow traffic that is intended for the private protected network to travel through the tunnel AND traffic that is intended elsewhere to flow outside the tunnel. This means that users with split-tunnels are free to surf the Internet (i.e., download viruses and worms through their own broadband connection) while simultaneously sending traffic into the tunnel destined for the private protected network.

Risk Mitigation Strategies

DSL and cable modem technologies have real tangible benefits for their users at relatively low cost. These services are fast, always available, and getting easier to deploy. However, users should exercise good Internet computing habits to minimize some of the risks that have been described. There are numerous personal firewalls available that will limit hackers' ability to scan and access the vulnerable hosts. In addition, home and small office networks should use the stateful inspection firewalls that are becoming more widely available. Good computing habits, such as having updated anti-virus software and clearing caches and cookies, help to minimize the risks of having a connection that is always available on the Internet.

In using broadband technology to access corporate networks through VPN tunnels, it is especially important to have personal firewalls installed with appropriate policies. In addition, split-tunneling should be disabled on the VPNs so that all access to the Internet is done through the corporate network and its firewalls. This may seem like a lot of work for administrators, but compared to the risks to the overall network, it is definitely worth doing.

The good news is that recent advances in client VPN software have integrated many of these functions into the client itself, so that the management of personal firewall policies and anti-virus updates is easier. For example, numerous vendors allow for control of personal firewall policies from the central VPN endpoint (firewall or VPN appliance) through the VPN client.

Future Growth

One of the great success stories in networking has been the widespread proliferation of broadband in the past five years. From a relatively modest start, high-speed Internet broadband connection has become readily available in most metropolitan areas. The In-Stat Group, a digital communications market research company, estimates that U.S. broadband subscribers will surpass 39 million customers by 2005. That is roughly 13 percent of the U.S. population. The same group performed a survey in 2001 and found that 50 percent of then-current broadband users did not use any form of intrusion detection protection. This means that if current trends continue, by 2005 the possibility exists that there will be more than 20 million unprotected broadband subscribers. Broadband usage will undoubtedly grow, along with its risks. Both casual subscribers and security professionals should exercise care and diligence in protecting themselves and others from the risks that follow exposure to the Internet via cable modems and DSL "always-on" connections.

Summary

Securing an enterprise network goes beyond configuring firewalls, servers, PCs, and networking equipment. It involves the combined evaluation of all the components of the network infrastructure, including people, processes, and equipment. The focus of this particular chapter has been on the foundation of network communications — the physical transmission media. Whether the requirements call for an optical fiber-based backbone or a high-speed wireless local area network, the relative strengths and weaknesses, with particular emphasis on security, should be thoroughly reviewed before making a selection. An informed decision on the cabling infrastructure ensures that the foundation of that network is built securely from the ground up.

25

Security and the Physical Network Layer

Matthew J. Decker, CISSP, CISA, CBCP

Networks have become ubiquitous both at home and in the office, and various types of media have been deployed to carry networking traffic. Much of the Internet is now carried over a fiber-optic backbone, and most businesses use fiber-optic cables to provide high-speed connectivity on their corporate campuses. Cable providers bring high-speed networking to many homes and businesses via coaxial cable. Local exchange carriers (LECs) and competitive local exchange carriers (CLECs) bring high-speed networking to many homes and businesses via twisted-pair cables, and numerous buildings are wired with twisted-pair cables to support high-speed networking to user desktops. Wireless networks have been deployed to provide network connectivity without the need for users to connect to any cables at all, although antennas and pigtail cables (coaxial cables) can be used to great advantage in maximizing the value of a wireless environment. These information highways and back roads lie within the physical layer of the seven-layer OSI (Open Systems Interconnection) model. The physical layer of the OSI model comprises the cables, standards, and devices over which data-link (layer 2 of the OSI model) encapsulation is performed.

This chapter serves as an introduction to common physical media used for modern networks, including fiber optics, twisted-pair, coaxial cables, and antennas. The reader will develop an understanding of each type of physical media, learn how an attacker might gain access to information by attacking at the physical layer, and learn how to apply sound industry practices to protect the network physical layer.

Fiber-Optic Cables

Much of the Internet is now carried over a fiber-optic backbone, and many businesses use fiber-optic cables to provide high-speed connectivity on their corporate campuses. Although they come bundled in a multitude of ways, there are essentially two types of fiber-optic cables on the market. These commonly used types of fiber-optic cables are known as "multimode" and "single mode."

Multimode fiber gets its name from the fact that light can take multiple "modes" or paths down the fiber. This is possible because the core, at the center of the fiber, is wide enough to allow light signals to zigzag their way down the fiber. Single-mode fiber, on the other hand, has a very narrow core, only 8 to 10 micrometers (μm) in diameter. This is wide enough for light traveling down the fiber to take only one path. It is the difference in size of the cores of these fiber types that gives each its unique characteristics.

Multimode fibers come in various sizes. The two most common sizes are 50 and 62.5-micrometer cores. The core is the center portion of the cable designed to carry the transmission signal (light). Cladding comprises the outer coating that surrounds the core and keeps light from escaping the fiber. Exhibit 25.1 provides a visual reference showing the core and cladding, and will assist in explaining key differences and similarities between single and multimode fiber.

Cladding is the material surrounding the fiber core. Both single and multimode fiber-optic cables that are typically used for networking applications have the same outside diameter (125 micrometers). The core is doped with a substance that alters the refractive index of the glass, making it higher than the cladding. This

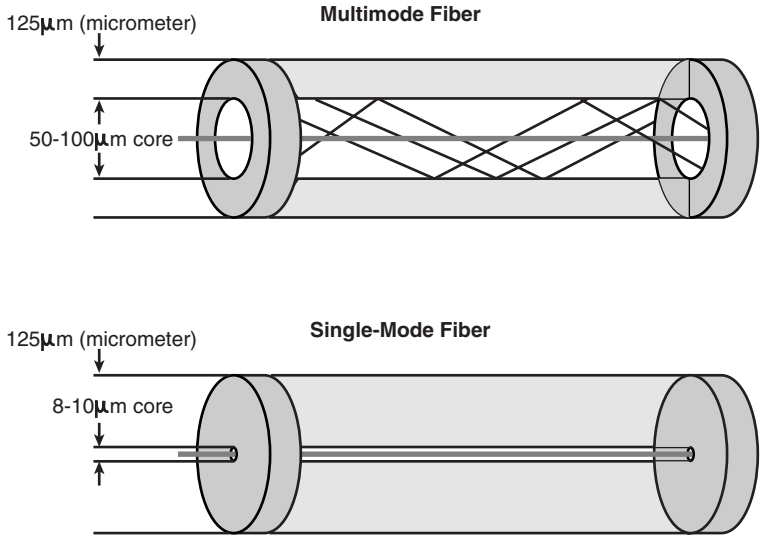


EXHIBIT 25.1 Core and cladding.

is desirable because light bends toward the perpendicular when passing from a material of high refractive index to a lower one, thus tending to keep the light from ever passing from the core into the cladding.

To clarify this point, we consider a simple test using air, water, and a flashlight. If you are in the air and shoot a flashlight into a pool at an angle, the portion of the beam that enters the water bends toward the perpendicular — toward the bottom of the pool. If you are in the water and shoot a flashlight out of the pool at an angle, the portion of the beam that enters the air bends away from the perpendicular — tending more to be parallel with the surface of the water. This is because the refractive index of water is greater than that of air. As you move the flashlight progressively more parallel to the surface of the water, less and less light escapes into the air until you reach a point at which no light escapes into the air at all. This is the principle of total internal reflection, and is the result that fiber-optic cable designers endeavor to achieve. Further, this explains why tight turns in optical fiber runs are not desirable. Bending a fiber-optic cable too tightly can change the angle at which light strikes the cladding, and thus permit some of the signal to escape from the fiber core. This is called "micro-bending" the fiber.

Another important term in the world of fiber-optic cabling is "graded index." Most multimode fiber is "graded-index" fiber, meaning that the refractive index decreases progressively from the center of the core out toward the cladding. This causes light in the core to continuously bend toward the center of the core as it progresses down the fiber. The diagram is oversimplified, in that it shows three modes of light traveling in straight lines, one traveling directly down the center of the core and two bouncing off the cladding, as they progress down the core of the fiber. With a graded-index fiber, this light beam travels in a more helical fashion down the fiber, always tending toward the center of the core as it progresses down the fiber. Further, because light traveling through a medium with a higher refractive index travels slower, the effects of "modal distortion" are significantly diminished in a graded index fiber.

There are a number of causes of signal loss in fiber-optic cables, but the two that best exemplify the differences between fiber types are "modal distortion" and "chromatic dispersion." Modal distortion is the spreading of the transmitted signal over time due to the fact that multiple modes of a signal arrive at the destination at different times. One signal takes many different paths, and each path is a different length, so the information arrives over a very short period of time rather than at a distinct point in time. The reason single-mode fiber is best for long distances is primarily because modal distortion is a factor in multimode fiber only. Single-mode fiber is most susceptible to losses due to chromatic dispersion. Light traveling through a vacuum travels at a constant speed, regardless of the wavelength. This is not so for materials like glass and plastic from which fiber-optic cables are made. "Chromatic dispersion" is signal degradation caused by the various wave components of the signal having different propagation velocities within the physical medium.

It is another type of loss that concerns us most from a security perspective. We previously introduced "micro-bending," which causes light to escape from the core into the cladding by simply bending the cable on a tight radius. This phenomenon gives us the most common means to tap a fiber-optic cable without having to perform a cable splice. By micro-bending a cable and placing an optical receiver against the cladding to collect the escaping light, the fiber can effectively be tapped, and the information traversing the cable can be captured. There are troubleshooting devices on the market that use the micro-bending technique to capture light from fiber-optic cables, and they take only seconds to install. These commonly available devices are only intended to identify whether or not a cable is active and do not actually process the data signal. Using this technique with more sophisticated equipment, a fiber-optic cable is easily tapped, although devices to do so are not readily available on the open market due to the lack of a commercial need for such a capability.

The brute-force means of tapping a fiber-optic cable involves cutting the cable and introducing a splice. This method brings the fiber-optic cable down for the minute or so required to introduce the splice, and introduces a 3-dB loss if half the light is transmitted into each half of the splice. If the target is monitoring their optical signal strengths, then this sudden added loss is easily detected, especially if found to have been introduced after a brief outage. Splices are also easily detected through use of an optical time domain reflectometer (OTDR), which is a tool that measures loss on a fiber-optic cable, and indicates the distance to points of significant signal loss.

Twisted-Pair Cables

Twisted-pair (TP) cabling is commonly used to carry network traffic within business complexes, and to bring high-speed Internet to homes and businesses through Digital Subscriber Line (DSL) services. DSL typically uses TP wiring to transport DSL signals from your home or business to your local telephone company's central office, where they terminate at a DSLAM (Digital Subscriber Line Access Multiplexer). DSLAMs translate these DSL signals into a format that is compatible with standard network equipment, such as switches and routers. CAT 3 or CAT 5 cabling, which we describe in some detail shortly, is typically used for these connections.

Twisted-pair cable is manufactured to comply with carefully crafted standards to support modern networks. A single cable is comprised of four wire pairs bundled together and bound by a protective sheath. The two types of TP cabling are identified as shielded twisted pair (STP) and unshielded twisted pair (UTP). STP cables have a conductive shield surrounding the wire bundle, which reduces EMI/RFI (electromagnetic interference/radio frequency interference) in order to:

- Limit the effects of the signal traversing the cable upon the local RF environment
- Limit the effects of a noisy RF environment upon the signal traversing the cable

UTP cables have no such shield, but the data-carrying performance characteristics of the medium are the same. Shielding a TP cable is not needed as a security measure to prevent eavesdropping, and proper installation of STP cable is a much more painstaking operation than that of UTP. It is recommended to avoid the use of STP except in environments where it is required for operational purposes, such as RF noisy industrial environments. An attacker can tap a shielded cable in the same manner as an unshielded cable, and no attacker will be found sitting in the parking lot across the street capturing your data from RF signals emanating from your unshielded cables. Fortunately, this is not where we find the interesting differences in performance characteristics among TP cables. For TP, we must dive into the various categories of cables prescribed in the prevailing standards. Exhibit 25.2 highlights the prevailing categories, standards, and bandwidth limitations for the TP cables commonly used in networking.

Note that each of these standards uses four wire pairs to carry signals. Each wire pair is twisted a certain number of times per foot of cable. These twists are not arbitrary, and, in general, the more twists per foot, the greater the bandwidth capacity of the cable. CAT 3 cables typically have about 3–4 twists per foot, while CAT 5e cables have about 36–48 twists per foot, but this varies depending on other factors, such as the distance between the wire conductors. These twists work their magic by serving two distinct purposes: (1) they reduce EMI and crosstalk between adjacent wire pairs, and (2) they play a key role in creating the proper inductance/capacitance relationship to sustain a given impedance (typically 100 ohms) for each wire pair. EMI and crosstalk are reduced because the signal from each wire of the pair cancels the electromagnetic radiation from the other. Maintaining the proper impedance for the cable minimizes signal loss and maximizes the distance over which high data rates can be sustained over the cable.

EXHIBIT 25.2 Categories, Standards, and Bandwidth Limitations for TP Cables

Category Designation	Bandwidth	Description
CAT 3	Bandwidth up to 16 MHz per wire pair (four-pair wire)	Performs to Category 3 of ANSI/TIA/EIA-568-B.1 & B.2, and ISO/IEC 11801 Class C standards. CAT 3 is standard telephone cable.
CAT 5e	Bandwidth up to 100 MHz per wire pair (four-pair wire)	Performs to Category 5e of ANSI/TIA/EIA-568-B.1 & B.2, and ISO/IEC 11801 Class D standards. 1000Base-T (IEEE 802.3a,b) supports 1000 Mbps operation over a maximum 100-meter-long Category 5e cable. Encoding is used to remain within the 100-MHz bandwidth limitation and achieve 1000-Mbps operation.
CAT 6	Bandwidth up to 250 MHz per wire pair (four-pair wire)	Performs to Category 6 requirements developed by TIA under the ANSI/TIA/EIA-568B-2.1, and ISO/IEC 11801Class E standards. The TIA/EIA 568B.2-1 standard was published in its final form in June 2002. 1000Base-TX (ANSI/TIA/EIA-854) supports 1000-Mbps operation over a maximum 100-meter-long Category 6 twisted-pair cable.
CAT 7	Bandwidth up to 600 MHz per wire pair (four-pair wire)	Performs to Category 7 of ISO/IEC 11801 Class E standard. At the time of this writing, TIA does not intend to adopt the ISO/IEC 11801 Class E standard.

Like fiber-optic cable, TP can be tapped without cutting or splicing the wires. The protective sheath must be cut to gain access to the four wire pairs, and the pairs must be separated by half an inch or so to achieve access to eight distinct wires. They must be separated to eliminate the EMI-canceling property of the closely bound and twisted arrangement. Only one wire from each pair need be tapped, but access to all four pairs may or may not need to be achieved, depending on the standard and configuration being used (e.g., 10Base-T, 100Base-TX, 100Base-T4, 1000Base-T, half-duplex, full-duplex, etc.). All four wires may or may not be in use, and they may be used for transmit or receive, depending on the standard in use. The attacker can now pull information from the targeted lines by inducing the electromagnetic signal of each onto his own cable set, and feeding it to his equipment for analysis. A more invasive technique for tapping a network is to cut the line, install connectors, and plug them into a hub, but such techniques are much easier for the targeted entity to detect.

The greatest security threat posed at the physical layer, however, is at accessible physical devices such as hubs and repeaters. A hub permits an attacker to simply plug into the device and gain direct access to the network. This permits an attacker to not only "sniff" all the information traversing a network cable, but also all the information traversing the device. Further, the attacker can initiate network traffic from a device much more easily than from a tapped cable. Further, if the hub is in an out-of-the-way place, the attacker can take an added step and install a wireless access point to provide continued remote access to the network from a nearby location.

Coaxial Cables

Cable providers bring high-speed Internet services to many homes and businesses via coaxial cable. These broadband cable modem services typically offer customers the ability to upload and download data at contracted rates. The maximum rate limits are set by the service provider and are programmed into the users' cable modems.

Coaxial cables are comprised of a center conductor surrounded by a dielectric nonconductor material, which in turn is surrounded by an outer conductor. The whole thing is wrapped in a protective sheath to form a

finished coaxial cable. The center conductor is typically used to carry the transmission signal, while the outer conductor usually functions as the signal ground.

Coaxial cable is no longer widely used to employ LANs, but the coaxial cable used for networking is typically the 50-ohm impedance variety, versus the 75-ohm variety used for CATV. A brief description of what these numbers mean is in order. Earlier, in the TP discussion, I mentioned that maintaining the proper impedance for the cable minimizes signal loss, and maximizes the distance over which high data rates can be sustained over the cable. This statement also holds true for coaxial cables.

So what does it mean that I have a 50-ohm cable? If you were to use an ohmmeter to measure the resistance across the center conductor and outer shield of a nonterminated coax cable, you would quickly learn that you do not receive a reading of 50 ohms. In fact, the reading approaches infinity. Now, if you were to transmit a signal down this nonterminated coax cable, you would find that nearly 100 percent (the remainder is absorbed by the line or radiated into the atmosphere) of the signal is reflected back to the source, because there is no load at the other end to absorb the signal. This reflected signal represents a "standing wave" on your coax line that is not desirable, as it is effectively noise on your line. If you terminate the cable with a resistor connected between the center and outer conductor, and repeat the testing process, you will find that the reflected wave is significantly reduced as the value of the chosen resistor approaches 50 ohms. Finally, you will learn that terminating the cable with a 50-ohm resistor eliminates the reflected wave, and thus provides the most efficient transmission characteristics for this cable.

This introduces the concept of impedance matching, and all coaxial cables are manufactured to an impedance specification (e.g., 50 ohms). In the real world, impedance matching can be good, but not perfect, and the way this is measured is through a metric called a voltage standing wave ratio (VSWR). A perfectly balanced transmission system with no "standing wave" on the transmission medium has a VSWR of 1:1 (one-to-one). This applies to our example of the 50-ohm coax line terminated with a 50-ohm resistor. In a worst-case scenario, such as the nonterminated test we performed, the VSWR is 1:∞ (infinity). It should be clear at this point that a lower VSWR is better. Modern communication systems and components provide VSWRs below 1:2, which is typically represented by dropping the "1:" ratio designation, and simply identifying "VSWR < 2." Failing to match the impedances of your transmission system components, including the cables, can have a dramatic impact on the rated bandwidth-carrying capacity of the system.

Do coaxial cables present a significant RFI problem, such that one needs to worry about attackers accessing the information traversing the line even if they are unable to physically tap the line? If all cables are properly terminated, the answer is no. The outer conductor completely surrounds the center conductor and provides effective RFI shielding and noise immunity. Cables that are connected to equipment on one end, and nonterminated on the other, however, can act as antennas, thus creating an RFI problem. As with all physical media, coaxial cables are susceptible to a physical tap if an attacker gains working access to the cable.

Antennas

We live in a digital world, but the laws of physics are not giving up any ground in the radio frequency (RF) analog arena. Coaxial cables are used to carry signals to and from antennas. Short coax cables, designed to permit the quick connect and disconnect of antenna components using various connector types, are commonly referred to as "pigtails." The concepts of impedance matching and VSWR, discussed earlier, are important concepts in selecting antennas, and are now assumed to be understood by the reader. Antennas are becoming increasingly important physical devices through which we achieve Internet, wide area network (WAN), and local area network (LAN) connectivity. In the networking arena, we use them for satellite communications, wireless access points, and point-to-point links between facilities. They offer the distinct advantage of establishing network connections while disposing of the need for cabling the gap between the antennas. Of course, from an attacker standpoint, these links dispense with the need to tap a physical cable to gain access to the transmission medium.

An antenna is a physical device designed to transfer electrical energy on a wire into electromagnetic energy for transmission via RF waves, and vice versa. It is tuned to a specific set of frequencies to maximize this transfer of energy. Further, an efficient antenna is impedance-matched to become part of an overall system that maintains a low VSWR. The characteristics of antennas we concern ourselves with in this chapter are gain, beam width, impedance, and VSWR. As we already have an understanding of the last two, let's look at the first two.

Gain is typically measured in terms of decibels referenced to an isotropic radiator (dBi). Isotropic means radiating in all directions, including up, down, and all around; thus, an antenna achieves gain by narrowing its focus to a limited area rather than wasting resources where no signal exists for reception, or is needed for transmission. It is important to note that dBis are measured on a logarithmic scale; thus, 10 dBi represents an increase of signal strength by 10 times, 20 dBi by 100 times, 30 dBi by 1000 times, etc. Every increase of 3 dBi is a doubling of gain; thus, 3 dBi represents an increase of signal strength by 2 times, 6 dBi by 4 times, 9 dBi by 8 times, 12 dBi by 16 times, etc.

Beam width is measured in degrees. An omni-directional antenna exhibits equal gain over a full circle, and thus has a beam width of 360 degrees. Directional antennas focus their gain on a smaller area, defined by beam width; thus, an antenna with a beam width of 90 degrees exhibits its quoted gain over an area shaped like a quarter piece of pie. Such an antenna would be a good choice for a wireless network antenna intended to serve one floor of a square building, if placed in one of the four corners and aimed at the opposing corner. Satellite antennas on Earth have narrow beam widths, as any portion of a transmitted signal that does not impact the satellite's antenna is wasted, and only a small percentage of the signal transmitted from the satellite actually reaches it. The satellite's own antenna, however, has a beam width tuned to ensure coverage of a prescribed area (e.g., all of Brazil).

By far, the most common use of antennas in current networks is for use with wireless access points (WAPs). The most common standards in use for WAPs are 802.11a, 802.11b, and 802.11g. The 802.11b and g wireless radios provide data rates up to 11 Mbps and 54 Mbps, respectively, and operate over a 2.4-GHz carrier wave (2.4 to 2.483 GHz) to transmit and receive data. These two standards use antennas with identical specifications because they share a common frequency band.

IEEE 802.11a is a physical layer standard (IEEE Std. 802.11a, 1999) that supports data rates ranging from 6 to 54 Mbps, and operates in the 5-GHz UNII band in the United States. The 5-GHz UNII band is segmented into three ranges, with the lower band ranging from 5.15 to 5.25 GHz, the middle from 5.25 to 5.35 GHz, and the upper from 5.725 to 5.825 GHz. Be careful using 802.11a devices in Europe, as these frequency ranges are not permitted for public use in many European countries. Due to the vast separation in frequencies, antennas intended for use with 802.11a are not compatible with those for 802.11b and g.

The greatest security concern for wireless networks is the fact that attackers have access to your transmitted signal. Do not assume that just because your wireless network manual told you that you would not be able to reliably connect beyond 500 feet, that an attacker cannot pick up the signal from much greater distances. The standard antennas that ship with most WAPs are omni-directional, and typically have a gain of about 1 or 2 dBi. Wireless access cards installed in user computers typically have internal antennas with similar characteristics. Given these numbers, 500 feet is generous, and the data rate will often suffer. A knowledgeable attacker is not going to rely on the default hardware to connect to your WAP. A common suite of attacker hardware includes a 5-dBi (or greater) omni-directional antenna and a 14-dBi (or greater) directional antenna with a narrow beam width (20 to 50 degrees), used in conjunction with a high-power (100 mW or more) wireless access card with dual external antenna inputs. This suite of physical layer tools permits both antennas to be connected to the wireless access card simultaneously, and the entire package fits neatly into a laptop carrying case. Using this hardware, the attacker is able to easily find the WAP using the omni-directional antenna, pinpoint the location of the WAP and receive a stronger signal (by about 10 times) with the directional antenna, and gain full duplex access to the WAP from much greater distances than can be achieved with default hardware. Note that an attacker will not likely use the same antenna to seek out 802.11a networks as 802.11b and g networks because the target frequencies are so far apart. Additional hardware is required to attack both standards.

Protecting against unauthorized access to WAPs requires that they be treated just like public access points, such as Internet connections. Connections through WAPs should be authenticated, filtered, and monitored in accordance with the organization's remote access policy, or wireless access policy, as applicable.

Protected Distribution Systems

We have discussed various types of physical media used to carry network traffic. We have made clear that a knowledgeable attacker with physical access to the transmission media can tap the cable to gain access to the data traversing that media, with the exception of antenna systems, which only require that an attacker achieve relatively close proximity. We are now prepared to address the protection of these physical layer assets. When it is impractical to use strong encryption to protect the confidentiality and integrity of data traversing a physical

link, the techniques incorporated by protected distribution systems (PDSs) may be warranted. A PDS is a wireline or fiber-optic telecommunication system that includes terminals and adequate acoustical, electrical, electromagnetic, and physical safeguards to permit its use for the unencrypted transmission of classified information [see NIS]. The physical security objective of a PDS is to deter unauthorized personnel from gaining access to the PDS without such access being discovered. There are two categories of PDS: (1) hardened distribution systems, and (2) simple distribution systems. Hardened distribution systems afford a high level of physical security by employing one of three types of carriers:

1. A hardened carrier, which includes specifications for burying cable runs and sealing protective conduits
2. An alarmed carrier, which includes specifications for the use of alarm systems to detect PDS tampering
3. A continuously viewed carrier, which mandates that carriers be maintained under continuous observation

Simple distribution systems afford a reduced level of security, can be implemented without the need for special alarms and devices, and are practical for many organizations. Some of the techniques, such as locking manhole covers and installing data cables in some type of carrier (or conduit), are sound practices. These are policy issues that promote the fundamental objective of protecting networks at the physical layer, are effective at protecting unauthorized access to critical data infrastructure, and should be considered for implementation to the extent that they are cost-effective for an organization.

Strong Security Follows Good Policy

Security of data traversing network cables and devices should be provided in accordance with written policy. Security must provide value if it is to make sense for an organization, and data management policy provides a foundation for implementing sound tactical security measures. Call it what you like, but what this author refers to as "data management policy" defines data classification and proper data handling instructions for an organization. Should we employ wireless technology for this project? Do we need to encrypt traffic over this link? Do we need to make use of a PDS to protect against unauthorized physical access to the cables that we are stringing throughout our campus? The answer to each of these hypothetical questions is a resounding "it depends," and is best resolved by referring to policy that mandates how data will be protected in accordance with its value to the organization. Sound practice in determining the value of data to an organization is to at least qualify, and, if you can do so meaningfully, quantify its value in terms of confidentiality, availability, and integrity.

The Department of Defense offers a good example of policy in action. Now, you are probably thinking, "Hey, that's the Department of Defense. What they do won't make sense for my organization." And you are right — you will need to develop your own. SANS offers a good template to work from, as do several good policy publications on the market. The DoD provides a good example because they have a policy that makes sense for them, it works, and most of us are familiar with the concepts. Everyone has heard the terms "Top Secret," "Secret," and "Unclassified," and we all understand that our ability to get our hands on documents or data gets more difficult as we tend toward "Top Secret." That is data classification, and it is important for every organization, although most organizations will probably find terms like "Proprietary," "Confidential," and "Public" to be more beneficial terms for their use. Data classification is one piece of the data management puzzle, but only addresses the confidentiality of the data. You also need to know the criticality of your data in terms of availability and integrity if you want to effectively protect it.

Conclusion

Protection at the physical layer can be accomplished by preventing an attacker from tapping the cable or device, encrypting data links, providing redundant data paths for high availability, and by reducing the likelihood of environmental impacts such as lighting strikes and excessive RF emissions. Detection and monitoring techniques must be employed to make certain that the physical assurances in place remain operational and intact. Organizations must develop a strategy, and then put that strategy in writing through sound policies that make sense for their business. Finally, they must protect the media in accordance with their policy by employing

physical network layer media that will not only meet the technical needs of the business, but also the strategic security needs of the business.

References

[NIS] National Information Systems Security (INFOSEC) Glossary, NSTISSI No. 4009, June 5, 1992, (National Security Telecommunications and Information Systems Security Committee, NSA, Ft. Meade, MD 20755-6000).

Protective Distribution Systems (PDS), NSTISSI No. 7003, 13 December 1996 (National Security Telecommunications and Information Systems Security Committee, NSA, Ft. Meade, MD 20755-6000).

1000BASE-T: Delivering Gigabit Intelligence on Copper Infrastructure, http://www.cisco.com/warp/public/cc/techno/media/lan/gig/tech/1000b_sd.htm

SANS, www.sans.org

Telecommunications Industry Association (TIA), http://www.tiaonline.org/

ISO, http://www.iso.ch/iso/en/ISOOnline.frontpage

26

Security of Wireless Local Area Networks

Franjo Majstor, CISSP

Introduction and Scope

Wireless communication represents a wide area of radio technologies, as well as protocols on a wide scope of transmission frequencies. Although initially used in venues where traditional wired networks were previously unavailable, the flexibility of wireless communication together with the adoption of the 802.11 standard has driven wireless communication to rapidly move into the information technology environment in the form of the so-called "wireless local area networks" (WLANs). This chapter aims to give information security practitioners a quick overview of WLAN technology and an in-depth view of the current security aspects of the same technology. Likewise, it presents possible solutions and directions for future developments.

WLAN Technology Overview

Wireless local area networking technology has existed for several years, providing connectivity to wired infrastructures where mobility was a requirement for specific working environments. Early networks were based on different radio technologies and were nonstandard implementations, with speeds ranging between 1 and 2 Mbps. Without any standards driving WLAN technologies, the early implementations of WLAN were relegated to vendor-specific implementation, with no provision for interoperability, thus inhibiting the growth of standards-based WLAN technologies. Even WLAN is not a single radio technology, but is represented by several different protocols and standards, which all fall under the 802.11 umbrella of the Institute of Electrical and Electronics Engineers (IEEE) standards.

Put simply, WLAN is, from the network connectivity perspective, similar to the wired local area network (LAN) technology with a wireless access point (AP) acting as a hub for the connection stations equipped with WLAN networking cards. As to the absence of wires, there is a difference in communication speed among the stations and AP, depending on which particular WLAN technology or standard is used for building the data wireless network.

802.11 Alphabet

WLAN technology gained its popularity after 1999 through the 802.11b standardization efforts of the IEEE, but it is not the only standard in the 802.11 family. Others are 802.11a, 802.11g, and 802.11i or 802.1x. For information security practitioners it is important to understand the differences between them, as well as to know the ones that have relevant security implications on wireless data communications. What is interesting to mention before we demystify the 802.11 alphabet is that particular letters (a, b, g, etc.) were assigned by the starting time of development of the particular standard. Some of them, however, were developed and accepted

faster than the others, so they will be described in the order of importance and the scope of usage instead of alphabetical order.

- *802.11b.* The 802.11b standard defines communication speeds of 1, 2, 5, and 11 Mbps at a frequency of 2.4 GHz, and is the most widely accepted WLAN standard at present with a large number of vendors producing 802.11b devices. The interoperability of the devices from different vendors is ensured by an independent organization originally called the Wireless Ethernet Compatibility Alliance (WECA), which identifies products that are compliant to the 802.11b standard with "Wi-Fi" (Wireless Fidelity) brand. WECA has recently renamed itself the Wi-Fi Alliance. From a networking perspective, the 802.11b standard offers 11 (United States), 13 (Europe), or 14 (Japan) different channels, depending on the regional setup, while only three of those channels are nonoverlapping channels. Each of the channels could easily be compared to an Ethernet collision domain on a wired network, because only stations, which transmit data on nonoverlapping channels, do not cause mutual collisions; also, each channel is very similar in behavior to a wired Ethernet segment in a hub-based LAN environment.
- *802.11a.* In 1999, the IEEE also ratified another WLAN technology, known as 802.11a. 802.11a operates at a frequency of 5 GHz and has eight nonoverlapping channels, compared to three in 802.11b, and offers data speeds ranging from 6 Mbps up to 54 Mbps. Despite its speed, at present, it is far from the level of acceptance of 802.11b due to several reasons. There are fewer vendor offers on the market and Wi-Fi interoperability testing has not yet been done. IEEE 802.11a operates at a different frequency than 802.11b and is not backwards-compatible with it. Due to different frequency allocations and regulations in different parts of the world, 802.11a might be replaced in the near future by 802.11g as a new compromise solution.
- *802.11g.* 802.11g is the late entrant to the WLAN standardization efforts; it tries to achieve greater communication speeds at the same unlicensed frequency as 802.11b (i.e., 2.4 GHz), and also tries to be backwards-compatible with it. However, 802.11g is at present not a ratified standard and there are no products offered by any of the vendors on the market. Due to practical reasons and the lateness of 802.11g standardization efforts, vendors are also offering dual-band devices that are operating at both 2.4 GHz and 5 GHz, thus offering a flexible future migration path for connecting stations.

As mentioned above, there are multiple other "letters" in the alphabet of 802.11 — 802.11d defines world mode and additional regulatory domains, 802.11e defines quality-of-service mechanisms, 802.11f is used as an inter-access point protocol, and 802.11h defines dynamic frequency selection and power control mechanisms — but all are beyond the scope of this chapter. Others, such as 802.11i and 802.1x, however, are very important from a security perspective and will be discussed in more detail in the sections on the security aspects of wireless LANs and future developments.

WLAN Security Aspects

Considering that it does not stop at the physical boundaries or perimeters of a wired network, wireless communication has significant implications on the security aspects of modern networking environment. WLAN technology has, precisely for that reason, built in the following mechanisms, which are meant to enhance the level of security for wireless data communication:

- Service Set Identifier (SSID)
- Device authentication mechanisms
- Media Access Control (MAC) address filtering
- Wired Equivalent Privacy (WEP) encryption

Service Set Identifier

The Service Set Identifier (SSID) is a mechanism similar to a wired-world virtual local area network (VLAN) identity tag that allows the logical separation of wireless LANs. In general, a client must be configured with the appropriate SSID to gain access to the wireless LAN. The SSID does not provide any data-privacy functions, nor does it authenticate the client to the access point (AP).

SSID is advertised in plaintext in the access point beacon messages. Although beacon messages are transparent to users, an eavesdropper can easily determine the SSID with the use of an 802.11 wireless LAN packet

analyzer or by using a WLAN client that displays all available broadcasted SSIDs. Some access-point vendors offer the option to disable SSID broadcasts in the beacon messages, but the SSID can still be determined by sniffing the probe response frames from an access point. Hence, it is important to understand that the SSID is neither designed nor intended for use as a security mechanism. In addition, disabling SSID broadcasts might have adverse effects on Wi-Fi interoperability for mixed-client deployments.

Device Authentication

The 802.11 specification provides two modes of authentication: open authentication and shared key authentication. Open authentication is a null authentication algorithm. It involves sending a challenge, but the AP will grant any request for authentication. It is simple and easy, mainly due to 802.11-compliancy with handheld devices that do not have the CPU capabilities required for complex authentication algorithms. Shared key authentication is the second authentication mode specified in the 802.11 standard. Shared key authentication requires that the client configure a static WEP shared key, and involves sending a challenge and then receiving an encrypted version of the challenge. Most experts believe that using shared key authentication is worse than using open authentication and recommend turning it off. However, shared key authentication could help deter a denial-of-service (DoS) attack if the attacker does not know the correct WEP key. Unfortunately, there are other DoS attacks available.

It is important to note that both authentication mechanisms in the 802.11 specifications authenticate only wireless nodes and do not provide any mechanism for user authentication.

Media Access Control (MAC) Address Authentication

MAC address authentication is not specified in the 802.11 standard, but many vendors support it. MAC address authentication verifies the client's MAC address against a locally configured list of allowed addresses or against an external authentication server. MAC authentication is used to augment the open and shared key authentications provided by 802.11, further reducing the likelihood of unauthorized devices accessing the network.

However, as required by 802.11 specification, MAC addresses are sent in the clear during the communication. A consequence for wireless LANs that rely only on MAC address authentication is that a network attacker might be able to bypass the MAC authentication process by "spoofing" a valid MAC address.

Wired Equivalent Privacy Encryption

All the previous mechanisms addressed access control, while none of them have thus far addressed the confidentiality or integrity of the wireless data communication. Wired Equivalent Privacy (WEP), the encryption scheme adopted by the IEEE 802.11 committee, defines for that purpose the use of a symmetric key stream cipher RC4 that was invented by Ron Rivest of RSA Data Security, Inc. A symmetric cipher uses the same key and algorithm for both encryption and decryption. The key is the one piece of information that must be shared by both the encrypting and decrypting endpoints. RC4 allows the key length to be variable, up to 256 bytes, as opposed to requiring the key to be fixed at a certain length. The IEEE specifies that 802.11 devices must support 40-bit keys with the option to use longer key lengths. Several vendors support 128-bit WEP encryption with their wireless LAN solutions. WEP has security goals of confidentiality and integrity but could also be used as an access control mechanism. A node that lacks the correct WEP key can neither send data to nor receive data from an access point, and also should neither be able to decrypt the data nor change its integrity. The previous statement is fully correct in the sense that the node that does not have the key can neither access the WLAN network nor see or change the data. However, several cryptography analyses listed in references have explained the possibility that, given sufficient time and data, it is possible to derive the WEP key due to flaws in the way the WEP encryption scheme uses the RC4 algorithm.

WEP Vulnerabilities

Because WEP is a stream cipher, it requires a mechanism that will ensure that the same plaintext will not generate the same ciphertext (see Exhibit 26.1). This is the role of an initialization vector (IV), which is concatenated with the key bytes before generating the stream cipher. The IV is a 24-bit value that the IEEE suggests, although does not mandate, to be changed per each frame. Because the sender generates the IV with no standard scheme or schedule, it must be sent unencrypted with the data frame to the receiver. The receiver can concatenate the received IV with the WEP key it has stored locally to decrypt the data frame.

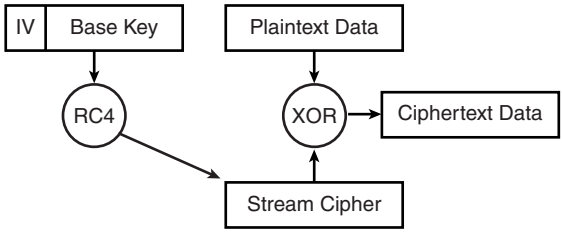


EXHIBIT 26.1 The WEP encryption process.

The IV is the source of most problems with WEP. Because the IV is transmitted as plaintext and placed in the 802.11 header, anyone sniffing a WLAN can see it. At 24 bits long, the IV provides a range of 16,777,216 possible values. Analysts at the University of California – Berkeley found that when the same IV is used with the same key on an encrypted packet (known as an IV collision), a person with malicious intentions could capture the data frames and derive information about the WEP key. Furthermore, cryptanalysts Fluhrer, Mantin, and Shamir (FMS) have also discovered inherent shortcomings in the RC4 key-scheduling algorithm. They have explained shortcomings that have practical applications in decrypting 802.11 frames using WEP, using a large class of weak IVs that can be generated by RC4, and have highlighted methods to break the WEP key using certain patterns in the IVs. Although the problem explained by FMS is pragmatic, the most worrying fact is that the attack is completely passive; however, it has been practically implemented by AT&T Labs and Rice University and some tools are publicly available on the Internet (e.g., Airsnort).

Further details about WEP weaknesses are explained in depth in the references, but for information security practitioners it is important to understand that the 802.11 standard, together with its current WEP implementation, has security weaknesses that must be taken care of when deploying WLAN networks.

WLAN Security Solutions

Major security issues in WEP include the following. First, it does not define the key exchange mechanism. Second, it has implementation flaws with the use of static keys. An additional missing security element from the current security 802.11 feature set is the lack of individual user authentication. Information security practitioners should be aware of this and look for solutions appropriate to their environments. A proposal jointly submitted to the IEEE by Cisco Systems, Microsoft, and other organizations introduced a solution for the above issues using 802.1x and the Extensible Authentication Protocol (EAP) to provide enhanced security functionality. Central to this proposal are two main elements:

1. EAP allows wireless clients that may support different authentication types to communicate with different back-end servers such as Remote Access Dial-In User Service (RADIUS)
2. IEEE 802.1x, a standard for port-based network access control

IEEE 802.1x Protocol

The 802.1x is a port-based security standard protocol developed by the IEEE 802.1 working group for network access control in wired networks. Its major role is to block all the data traffic through a certain network port until the client user authentication process has been successfully completed. In essence, it operates as a simple switch mechanism for data traffic, as illustrated in Exhibit 26.2.

Extensible Authentication Protocol

The Extensible Authentication Protocol (EAP) is a flexible authentication protocol specified in RFC 2284 that rides on top of another protocol such as 802.1x or RADIUS. It is an extension of the Point-to-Point Protocol (PPP) that enables the support of advanced authentication methods, such as digital certificates, MD-5 hashed

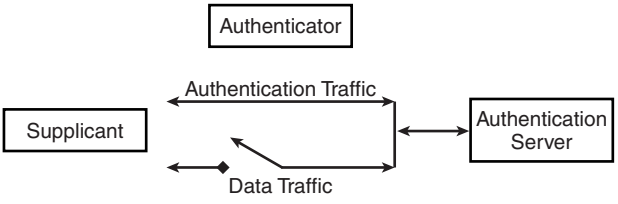


EXHIBIT 26. 2 The 802.1x port access control mechanism.

authentication, or One-Time Password (OTP) authentication mechanisms. Layers of 802.1x and EAP methods are illustrated on Exhibit 26.3.

Dynamic Key Exchange Mechanisms

Each of the EAP protocols, except EAP-MD5, provides a solution to WEP security problems by tying the dynamic key calculation process to an individual user authentication. With the EAP mechanism, each individual user obtains its own unique dynamic WEP key that is changed every time the user connects to an access point. Alternatively, it could also be recalculated based on the timeout defined on the authentication server.

EAP-MD5

EAP-MD5 (Message Digest 5) is the easiest of the EAP authentication schemes, and provides only user authentication. The user authentication scheme employed is a simple username/password method that incorporates MD5 hashing for more secure authentication. It provides neither a mutual authentication nor the method for dynamic WEP key calculation; hence, it is still requires manual WEP key configuration on both sides, clients as well as on the wireless access point (AP).

EAP-Cisco Wireless or Lightweight Extensible Authentication Protocol (LEAP)

EAP-Cisco Wireless, also known as LEAP (Lightweight Extensible Authentication Protocol), is an EAP method developed by Cisco Systems. Based on the 802.1x authentication framework, EAP-Cisco Wireless mitigates several of the weaknesses by utilizing dynamic WEP key management. It supports mutual authentication between the client and an authentication server (AS), and its advantage is that it uses a simple username/password mechanism for providing dynamic per-user, per-session WEP key derivation. A wireless client can only transmit EAP traffic after it is successfully authenticated. During user login, mutual authentication between

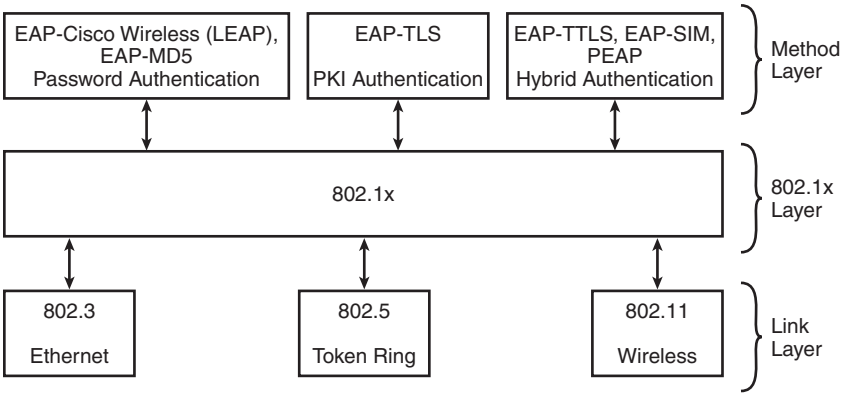


EXHIBIT 26.3 EAP and 802.1x layers.

the client and the AS occurs. A dynamic WEP key is then derived during this mutual authentication between the client and the AS, and the AS sends the dynamic WEP key to the access point (AP). After the AP receives the key, regular network traffic forwarding is enabled at the AP for the authenticated client. The credentials used for authentication, such as a log-on password, are never transmitted in the clear, or without encryption, over the wireless medium. Upon client log-off, the client association entry in the AP returns to the non-authenticated mode. The EAP-Cisco Wireless mechanism also supports dynamic re-keying based on the predefined timeout preconfigured on the AS. The disadvantages of the EAP-Cisco Wireless method is that, although it is based on an open standard, it is still proprietary and its authentication mechanism is limited to static usernames and passwords, thus eliminating the possible use of One-Time Password (OTP) user authentication.

EAP-TLS

The EAP Transport Layer Security (TLS) as defined in RFC 2716 is a Microsoft-supported EAP authentication method based on the TLS protocol defined in RFC 2246. TLS is the IETF version of Secure Socket Layer (SSL) used in most Web browsers for secure Web application transactions. TLS has proved to be a secure authentication scheme and is also available as an 802.1x EAP authentication type. TLS utilizes mutual authentication based on X.509 certificates. Because it requires the use of digital certificates on both the client and on the authentications server side, it is the most secure method for user authentication and dynamic per-user, per-session WEP key derivation that also supports OTP user authentication. EAP-TLS security superiority over any of the other EAP methods is, at the same time, its weakness, because it is overkill to require the establishment of a Public Key Infrastructure (PKI) with a certificate authority to distribute, revoke, and otherwise manage user certificates just to be able to use layer 2 WLAN connectivity. This is the main reason why TLS has resulted in the development of hybrid, compromised solutions such as EAP-TTLS and PEAP.

EAP-TTLS

The EAP-TTLS (or EAP Tunneled TLS) protocol is an 802.1x EAP authentication method that was jointly authored by Funk Software and Certicom, and is currently an IETF draft RFC. It uses server-side TLS and supports a variety of authentication methods, including passwords and OTPs.

With the EAP-TTLS method, the user's identity and password-based credentials are tunneled during authentication negotiation, and are therefore not observable in the communications channel. This prevents dictionary attacks, man-in-the-middle attacks, and hijacked connections by wireless eavesdroppers. In addition, dynamic per-session keys are generated to encrypt the wireless connection and protect data privacy. The authentication server can be configured to re-authenticate and thus re-key at any interval, a technique that thwarts known attacks against the encryption method used in WEP.

Protected EAP (PEAP)

Protected EAP (PEAP) is another IETF draft developed by RSA Security, Cisco Systems, and Microsoft. It is an EAP authentication method that is — similar to EAP-TTLS — designed to allow hybrid authentication. It uses digital certificate authentication for server-side only, while for client-side authentication, PEAP can use any other EAP authentication type. PEAP first establishes a secure tunnel via server-side authentication, and second, it can use any other EAP type for client-side authentication, like one-time passwords (OTPs) or EAP-MD5 for static password-based authentication. PEAP is, by using only server-side EAP-TLS, addressing the manageability and scalability shortcomings of EAP-TLS for user authentication. It avoids the issues associated with installing digital certificates on every client machine as required by EAP-TLS, so the clients can select the method that best suits them.

EAP-SIM

The EAP subscriber identity module (SIM) authentication method is an IEEE draft protocol designed to provide per-user/per-session mutual authentication between a WLAN client and an AS, similar to all the previous methods. It also defines a method for generating the master key used by the client and AS for the derivation of WEP keys. The difference between EAP-SIM authentication and other EAP methods is that it is based on the authentication and encryption algorithms stored on the Global System for Mobile Communications (GSM) subscriber identity module (SIM) card, which is a smart card designed according to the specific requirements detailed in the GSM standards. GSM authentication is based on a challenge–response mechanism and employs a shared secret key, which is stored on the SIM and otherwise known only to the GSM operator's

Authentication Center. When a GSM SIM is given a 128-bit random number as a challenge, it calculates a 32-bit response and a 64-bit encryption key using an operator-specific algorithm. In GSM systems, the same key is used to encrypt mobile phone conversations over the air interface.

EAP Methods Compared

It is obvious that a variety of EAP methods try to solve WLAN security problems. All of them, with the exception of the EAP-SIM method specific to GSM networks and EAP-MD5, introduce solutions for user authentication and dynamic key derivation, by using different mechanisms of protection for the initial user credentials exchange and different legacy user authentication methods. The feature of EAP method comparison is shown in table form on Exhibit 26.4.

VPN and WLAN

Combining IPSec-Based VPN and WLAN

Because a WLAN medium can carry IP over it without any problems, it comes easily as an idea for solving all security problems of WEP to simply run the IP Security Protocol (IPSec) over the WLAN. While the fairly standardized and security-robust IPSec-based solution could certainly help improve the security of communication over WLAN media, IPSec also has its own limitations. WLAN media can carry any type of IP traffic, including broadcast and multicast, while IPSec is limited to unicast traffic only. Hence, if it is necessary to support multicast application over WLAN, IPSec does not represent a viable solution. While it is possible to run IPSec encryption algorithms like DES or 3DES in hardware, it is very seldom the case that client personal computers are equipped with the additional IPSec hardware accelerators. That means that IPSec encryption is done only in the software, limited to the speed of the personal computer CPU, which certainly represents a bottleneck and thus reduces the overall speed of communication over WLAN media (in particular on low-CPU hand-held devices). IPSec authentication mechanisms support pre-shared keys, RSA digital-signatures, and digital certificates, which are all flexible options, but only digital certificates are the most scalable and robust secure option, which requires establishment of PKI services. If PKI services are already established, the same security level could also be achieved with EAP-TLS. The EAP-TLS method avoids all the limitations of IPSec with regard to the overall solution. Last but not least, running IPSec on user personal computers most of the time requires, depending on the operating systems, additional software installation plus loss of user transparency, and it keeps the device protected only while the IPSec tunnel is established. Overall, IPSec-protected WLAN communication could possibly solve WLAN security problems, but it is not always applicable and requires an examination of its benefits and disadvantages before being deployed.

Future Directions

The IEEE has formed a task group i (TGi) working on the 802.11i protocol specification to solve the security problems of the WEP protocol and provide a standardized way of doing so. The solution will most probably come in multiple phases with initial help for already-known problems, up to the replacement of the encryption scheme in the WEP protocol.

EXHIBIT 26.4 The EAP Methods Compared

	EAP-MD5	EAP-TLS	EAP-Cisco Wireless	EAP-TTLS	PEAP
Dynamic WEP key derivation	No	Yes	Yes	Yes	Yes
Mutual authentication	No	Yes	Yes	Yes	Yes
Client certificate required	No	Yes	No	No	No
Server certificate required	No	Yes	No	Yes	Yes
Static password support	Yes	No	Yes	Yes	Yes
OTP support	No	Yes	No	Yes	Yes

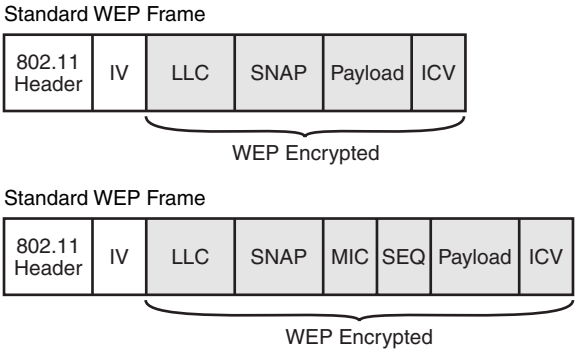


EXHIBIT 26.5 Message Integrity Check: MIC.

Temporal Key Integrity Protocol

The Temporal Key Integrity Protocol (TKIP) aims to fix the WEP integrity problem and is intended to work with existing and legacy hardware. It uses a mechanism called fast-packet re-keying, which changes the encryption keys frequently and provides two major enhancements to WEP:

1. A message integrity check (MIC) function on all WEP-encrypted data frames
2. Per-packet keying on all WEP-encrypted data frames

The MIC (Exhibit 26.5) augments the ineffective integrity check function (ICV) of the 802.11 standard and is designed to solve the following major vulnerabilities of IV reuse and bit flipping. For initialization vector/base key reuse, the MIC adds a sequence number field to the wireless frame so that the AP can drop frames received out of order. For the frame tampering/bit flipping problem, the MIC feature adds an MIC field to the wireless frame, which provides a frame integrity check not vulnerable to the same mathematical shortcomings as the ICV.

TKIP (Exhibit 26.6) is using advanced hashing techniques, understood by both the client and the access point, so that the WEP key is changed on a packet-by-packet basis. The per-packet key is a function of the dynamic base WEP key.

The Wi-Fi Alliance has accepted TKIP as an easy, software-based upgrade, an intermediate solution for WEP security issues, and has established a new certification program under the name of Wi-Fi Protected Access (WPA). On the side of TKIP for WEP encryption improvement, WPA also covers user authentication mechanisms relaying on 802.1x and EAP.

Advanced Encryption Standard

In essence, all of the above-mentioned proposals do not really fix the WEP vulnerabilities, but when combined with packet re-keying, significantly reduce the probability that an FMS (Fluhrer et al.) or Berkeley attack will be effective. Flaws with RC4 implementation still exist but are more difficult to compromise becausethere is

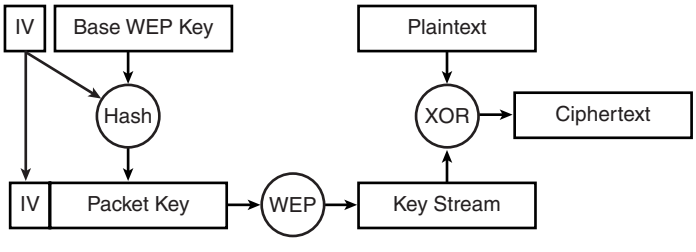


EXHIBIT 26.6 The TKIP encryption process.

less traffic with identical keys. Standards bodies are investigating the use of the Advanced Encryption Standard (AES) as a possible alternative to RC4 in future versions of 802.11 security solutions. AES is a replacement for DES (Data Encryption Standard) and uses the Rijndael algorithm, which was selected by the U.S. Government to protect sensitive information. However, the standardization of AES to solve encryption problems is still under discussion, without any commercially available products on the market today. As standards continue to develop, many security experts recommend using the Internet Protocol Security (IPSec) standard that has been deployed in global networks for more than five years as an available alternative.

Summary

WLAN technology based on 802.11 standards plays an important role in today's modern networking; and although it has its advantages in rapid and very flexible deployment, information security practitioners should be aware of its security weaknesses. Multiple proposals are on the scene to address major flaws in the WEP security protocol with different mechanisms for cryptographic integrity checks, dynamic key exchange, and individual user authentication. It is important to understand what security functionalities they offer or miss. While IPSec VPN technology deployed over WLANs is also an optional solution, it requires additional hardware and, hence, creates additional costs in addition to its limitations. Of the multiple EAP proposals for per-user/per-session dynamic WEP key derivation, it is expected that EAP-TTLS or PEAP will be the predominant solutions in the near future, assuming that either solution gets ratified. As the short-term solution for 802.11 security problems, an alliance of multiple vendors has decided to adopt the TKIP solution as a sufficient fix for existing WEP vulnerabilities under the name of Safe Secure Networks (SSN), even before its final approval by the IEEE 802.11i standards body. The Wi-Fi Alliance has adopted a similar scheme for its vendor interoperability testing under the name of Wi-Fi Protected Access (WPA). Together they predict a bright future for safer WLAN deployment.

References

Aboba, B., Simon, D., PPP EAP TLS Authentication Protocol, RFC 2716, October 1999.

Andersson, H., Josefsson, S., Zorn, G., Simon, D., and Palekar, A., Protected EAP Protocol (PEAP), IETF Internet Draft, draft-josefsson-pppext-eap-tls-eap-05.txt, September 2002.

AT&T Labs and Rice University paper, Using the Fluhrer, Mantin, and Shamir Attack to Break WEP, www.cs.rice.edu/~astubble/wep/wep_attack.pdf, August 21, 2001.

Blunk, L., and Vollbrecht, J., EAP PPP Extensible Authentication Protocol (EAP), RFC 2284, March 1998.

Bovison, N., Goldberg, I., and Wagner, D., "Security of the WEP Algorithm," www.isaac.cs.berkeley.edu/isaac/wep-faq.html.

Greem, Brian C., Wi-Fi Protected Access, www.wi-fi.net/opensection/pdf/wi-fi_protected_access_overview.pdf, October 2002.

Fluhrer, S., Mantin, I., and Shamir, A., "Weaknesses in the Key Scheduling Algorithm of RC4," www.cs.umd.edu/~waa/class-pubs/rc4_ksaproc.ps.

Funk, P., and Blake-Wilson, S., EAP Tunneled TLS Authentication Protocol (EAP_TTLS), IETF Internet Draft, draft-ietf-pppext-eap-ttls-01.txt, February 2002.

SAFE: Wireless LAN Security in Depth, white paper from Cisco Systems, Inc., Cisco.com/warp/public/cc/so/cuso/epso/sqfr/safwl_wp.htm.

27

Securing Wireless Networks

Sandeep Dhameja, CISSP

In 1999, the IEEE drafted the standard known as 802.11x, which allows multiple computers to share network Internet connection without having to provision expensive cabling. Now palm devices, hand-held computers and other POAs allow users to access stored data in hotel lounges, coffee kiosks, and airport terminals.

Wireless data networks provide always-on network connection. The data network connections do not require a physical data network connection. As a radio signal, wireless data is always pervasive; thus:

- Wireless network users can move throughout the coverage areas of the data signal between production floors, conference rooms, and offices.
- Wireless local area networks (WLAN) can be set up in hours, in comparison with days and weeks that are spent in wiring conventional data networks.
- Ease of deployment leads to more aggressive costs of installation compared to the conventional wired networks. If the average wired network costs approximately $100 per connection, extending the data network for 50 additional users will cost approximately $5000. A single wireless access point (WAP) can serve 50 users at a cost of approximately $150. In addition, WLAN clients can connect to the WAP at approximately $60 to $70 per client. Thus, a wireless data network can be configured for approximately $3200.

While wireless networks have been adopted by home users, widely reported and easily exploited weaknesses in the commercially available wireless products have affected the widespread deployment of wireless-based networks in the large business and enterprise environments. Most early adopters of the technology did not know exactly what the weaknesses were, and they have accepted the fact that wireless networks are inherently insecure.

While working with commercially available wireless networking products, some of the questions that arise include the following: Can the WLANs be deployed securely? What are the security holes in the current standard? How does the security of the wireless-based network work? Where is wireless security headed in the future? This chapter attempts to address questions related to wireless networking security in an enterprise environment.

Owing to the lack of physical control on the access to WLAN data, it is relatively easy to compromise wireless network data and information. The potential risk associated with the loss of data integrity and confidentiality is high because access to emitted radio signals and data is only limited by the physical range. Most attacks can be initiated with relative ease because wireless-data networks are deployed in hard-to-wire network environments that blindly trust all users within the proximity of a WAP; these environments include hospitals, convention centers, university classrooms, airport waiting lounges, cyber-cafes, and kiosks. Efforts to improve business productivity to the deployment of wireless data networks in data warehouses, meeting rooms, and telecommuting worker offices. Even in these trusted locations, radio transmissions propagate beyond the physical walls of the building into the side-street parking areas, parking lots, public hallways, and next-door residential area buildings.

The 802.11 standard operates in two modes: the infrastructure mode and the ad hoc mode. In the infrastructure mode, the wireless data network consists of at least one WAP and a wired connection to a set of wireless end stations. The WAP acts as a router, assigns IP addresses to workstations, controls data encryption on the network, bridge or routes wireless traffic to a wired Ethernet data network. The WAP can be compared with a base station in cellular networks, and thus the configuration is called a Basic Service Set (BSS).

When two or more BSSs are combined to form a single sub-network, then the network is referred to as an extended service set (ESS). Traffic is forwarded from one BSS to another to facilitate the movement of wireless stations between BSSs. The wired network system connecting the network is an Ethernet LAN. Because most corporate WLANs require access to the wired LAN for services (file servers, printers, Internet links), they operate in infrastructure mode.

Ad hoc mode is a set of 802.11 wireless stations that communicate directly with each other without using an access point or any connection to a wired network. In the ad hoc mode, wireless networks have multiple wireless clients talking to each other as peers to share data among themselves without the aid of a central WAP. This basic topology is useful in setting up a wireless network anywhere a data network infrastructure does not exist, such as a hotel room, a convention center, an airport, etc. Thus, the ad hoc mode is also referred to as a peer-to-peer mode or an independent basic service set (IBSS).

A malicious user can attempt to break into the network and access the data using readily available shareware tools, a wireless network interface card (NIC) operating in promiscuous mode, sitting inside the building across the street or on a different floor in the building. In many cases, the external WLAN data traffic can be modified as it enters the wired LAN data network. This can be easily done if wireless data access is not terminated before the firewall and no traffic-control measures are enforced.

The IEEE 802.11b (Wi-Fi) standard is an international standard commonly adopted to deploy networks in residential apartment buildings, houses, public places, and businesses. As part of the aggressive deployment efforts, wireless network companies are actively building Wi-Fi data networks in public places such as hotels, airports, conference centers, and retail establishments.

Owing to its very nature, wireless data is a radio signal that is not limited by any physical boundaries when it is transmitted. A WAP, using a monopole antenna, broadcasts the wireless data signal in an omni-directional pattern. Without physical obstacles, the 802.11b standard allows for full-speed data transmission at 11 Mbps (or 11×10^6 bits per second). The transmission speed at 11 Mbps is theoretical. Wi-Fi reaches a speed of only 7 Mbps up to a distance of 300 feet from the WAP. This transmission distance can be increased to approximately 2000 feet with additional wireless signal shaping.

With only 11 out of the 15 channels available operating data channels in North America, the IEEE 802.11b protocol operates using a Direct Sequence Spread Spectrum (DSSS) such that the wireless NICs automatically search for WLANs while operating on these channels. The NIC begins the data communication with the WAP once it finds the correct channel as long as the security settings on the client and the WAP match. The limited bandwidth of 11 Mbps per access point is divided among all users on the WAP. Thus, if ten users access the same WAP, communication of the data will occur at approximately 1 Mbps (equivalent of a xDSL communication link speed). Because the standard does not support load balancing of data across multiple WAPs, saturation of a WAP can be alleviated by adding another WAP. The WAP network architecture is comprised of three components: the wireless client, the wireless gateway, and the wireless ready application. Up to three WAP clients may be configured in the vicinity of each other. Each WAP client is configured with a different name and operates on a different frequency channel. While some vendors provide proprietary WAP Client load balancing solutions and architecture solutions, the basic configuration of the wireless data network is built around the three basic components.

The transmitted data consists of management data, control data, and information data.

The IEEE 802.11a (Wi-Fi5) protocol is licensed to operate in North America, at a higher frequency of 54 MHz, in a less-crowded data spectrum. While operating at a higher frequency, the protocol is limited to a distance of 1000 feet. The major advantage is its speed of data communications. The 802.11a spectrum is divided into eight sub-network segments (or channels) of about 20 MHz each. The channels are made up of 52 carriers, each of 300 kHz, and can present a maximum of 54 Mbps — thus taking the WLAN from the first-generation Ethernet (operating at 10 Mbps) to the second generation (Fast Ethernet operating at 100 Mbps). The new specification is based on an OFDM (Orthogonal Frequency Division Multiplexing) modulation scheme. The RF system operates in the 5.15–5.25, 5.25–5.35, and 5.725–.825 GHz UNII bands. The OFDM system provides eight different data rates between 6 and 54 Mbps. It uses BPSK, QPSK, 16-QAM, and 64-

QAM modulation schemes coupled with forward error correcting coding. It is important to remember that 802.11b is completely incompatible with 802.11a.

The IEEE 802.11g protocol operates in the same frequency as the IEEE 802.11b protocol and also uses the same scheme for data multiplexing OFDM as the IEEE 802.11a protocol — the exception being that it uses the 2.4-GHz data spectrum instead of the 5-GHz spectrum. Although the 802.11g protocol is backwards-compatible with the 802.11b protocol, the speed of data transmission is significantly lower, at 22 Mbps. The slower speed of data transmission is because the protocol has to delay the transmission at 22 Mbps to accommodate the lower rate of transmission. Similar to the IEEE 802.116 clients, as more and more 802.11g clients come online, the throughput of the data network also starts to drop.

Wireless-ready portable devices such as personal digital assistants (PDAs) and mobile computing devices such as cellular phones communicate based on the IEEE 802.15 specification. This specification focuses on the interoperability among both wireless and wired networks.

Wireless broadband access, on the other hand, serves as an alternate broadband access technology based on the IEEE 802.16 standard. Especially for wireless metropolitan area networks (WMANs), the range of operations varies from 2 GHz to 66 GHz.

The IEEE 802.11e provides QoS (quality-of-service) enhancements that make IEEE 802.11 b and IEEE 802.11a better standards. The IEEE 802.11i standard's security is enhanced with Advanced Encryption Standard (AES)-based data encryption replacing the Wireless Equivalent Privacy (WEP). Because the changes are made at the chip level, any wireless systems shipped prior to the standard's approval will not be capable of supporting the IEEE 802.11i standard.

A typical wireless data network is set up such that the access points (WAPs) are placed wherever it is convenient, not where the WAPs are most securely configured. To secure the WAP and the wireless data networks, it is important to understand how the WAPs communicate. Because WAPs do not know what is connected to them at all times, they send out beacon packets at a frequency of 10 Hz (or at a rate of ten packets per second). These beacon packets help a Wi-Fi client to associate with a wireless network. The client associates itself with a WAP to communicate. To successfully communicate, the WAP must be configured in the infrastructure mode. The association is a two-step process involving three states:

1. Unauthenticated and unassociated
2. Authenticated and unassociated
3. Authentic and associated

To communicate and exchange messages, clients exchange messages using *management frames*. All WAPs transmit a beacon management frame at fixed intervals. To associate with a WAP and join the Basic Service Set (BSS), a client listens for beacon messages to identify the access points within range. The client then selects the BSS to join the data network independent of the vendor.

The client association begins with the unauthenticated and unassociated state, undergoes a successful authentication, and moves into the second state, authenticated and unassociated. Next, the client sends an association request frame to the WAP. The WAP, in turn, responds with an association response frame.

Service Set Identifier (SSID)

Service set identifiers (SSIDs) are similar to authorization passwords that assist with differentiating wireless data networks from each other. Thus, SSIDs are unique identifiers that permit a wireless communication client to establish a data connection to the WAP. Most vendors of wireless equipment do not enable the data encryption on the WAP. It is good security practice to change SSIDs on a frequent basis, as is done with administrative passwords. SSIDs are set by default, depending on the product manufacturer. Some of the rather trivial names default (vendor-specific SSIDs) include:

- Intel = 101
- D-Link/GemTek, Advanced Multimedia Internet (AMI) = Default
- Linksys/GST = Linksys
- Cisco = Tsunami
- Addtron, SMC = WLAN

- Lucent/Agere/Orinoco = WaveLAN Network
- Delta/Netgear = Wireless

Because these default SSIDs are widely published, not changing the default vendor-specified settings makes it much easier to detect a WAP. As part of establishing the footprint of the wireless data network, SSIDs are discovered to be renamed; however, they are now set to something more meaningful, such as the WAP's location or IP address or department name. Just like passwords, SSIDs should be renamed and defined with a non-meaningful string of alphanumeric characters. SSID settings on the data network should be considered as the first level of WLAN security. Renaming the SSID to be less apparent and not easily guessable can only make it more difficult to run reconnaissance attacks. SSID detection is the most common exploit used by wireless network detection software.

A wireless client sends a probe request management frame to find a WAP affiliated with a desired (Service Set Identifier) SSID. Prior to establishing successful communications between the authenticated client and the WAP, a dialogue is initiated (see Exhibit 27.1). This mechanism is defined as *association*. After identifying the WAP, the client station and the WAP perform a mutual authentication by exchanging several management frames as part of the communication process. Upon completion of a successful authentication, the WAP client station moves into the second state, *open shared key authentication and unassociated*. At this point, any client may begin a conversation with the WAP. The WAP, in response, sends back a string of challenge text. The client, in turn, encrypts using the shared WEP key. Thus, the client sends an association request frame to the WAP and the WAP responds with an association response frame. If the response is encrypted correctly, the client is allowed to communicate with the WAP and thus moves on to the next layer of secured communications.

Then there is a transitioning from the second state to the third and final state, which is *open authenticated and associated*. Thus, a wireless data network can be detected using the three basic modes:

1. *Association polling,* where the Wi-Fi card associates itself with the strongest WAP that has no specific SSID setting. In this case, the SSID is set to ANY. Furthermore, the Wi-Fi's statistics are also polled to detect additional WAPs in the vicinity.
2. *Scan mode polling,* where the Wi-Fi card keeps track of received beacon and probe response packets. The Wi-Fi card sends a scan request and receives a scan response back with WAP information.
3. Monitor-mode protocol analysis, where the Wi-Fi card — when set into monitor mode — provides analysis of both beacon and probe data packet requests. This mode detects closed WAPs and wireless nodes. WAP settings include SSID, authentication in use, level of encryption, and the speed of the data network.

The principle of a wireless data sniffer is the same as that of an Ethernet data sniffer because base stations typically broadcast ten beacon data packets per second, advertising network IDs and capabilities.

WLANs transmit data in cleartext with no data protection. What does this really mean? As mentioned in the earlier *open shared key authentication* description, the challenge string is sent using cleartext transmission. A malicious attacker snooping the network traffic now obtains two of the three components that make up the

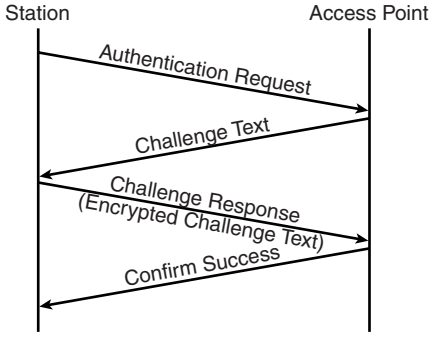


EXHIBIT 27.1 Shared key authentication.

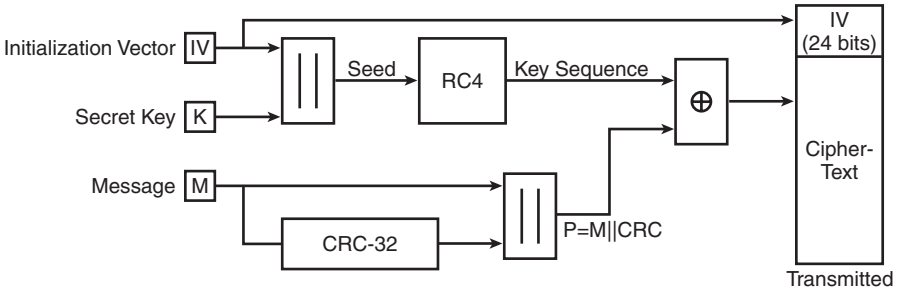


EXHIBIT 27.2 Wireless Equivalent Privacy (WEP).

authentication mechanism, that is, the cleartext challenge string and the encrypted challenge string. Extrapolating from the equations used to calculate the RC4-based message encryption, the attacker derives the shared authentication key. Most vendors ship their commercially available products with no security protection in place. As a result, malicious users are able to compromise WLANs with relative ease. These exploits are commonly referred to as *parking lot attacks* and do provide a backdoor to the wired data network. Wireless data traffic can be captured, altered, and replayed, if necessary, within a few hundred feet of a WAP. Legitimate data can be monitored using shareware tools, and communication can be hijacked using cache poisoning to gain control of TCP sessions.

Because the same keys are used for *open shared key authentication* and also for Wireless Equivalency Privacy (WEP), all wireless traffic exchanged from and to the WAP and to and from the clients can be deciphered.

A rogue access point is defined as an access location that is not authorized in an IEEE 802.11 wireless data network. A rogue access point may be a result of a group of users who are extending the existing wired Ethernet data network or a malicious attempt to access network resources without authentication. These points can be identified by capturing data packets and analyzing those that do not belong to authorized WAPs. Several commonly available open source analysis tools gather WAP data, including rogue points to capture the data packets that do not use the WAPs identified on the authorized list.

Wireless Equivalency Privacy (WEP)

Wireless Equivalency Privacy (WEP) is the encryption standard for IEEE 802.11b wireless data transmission. As part of the encryption, the Cyclic Redundancy Checksum (CRC) is calculated using CRC-32 over a plaintext message. The CRC ensures that data integrity is preserved during data transmission. A 24-bit random initialization vector (IV) is concatenated with the 40-bit secret key (k). The data encryption algorithm uses RC4 (Ron's Code 4), a stream cipher developed in 1987 by Ron Rivest. The IV is combined with a fixed-length secret key (k + IV) to form the seed as shown in Exhibit 27.2. The RC4, in combination with the seed, generates a series of pseudorandom data bits referred to as the key sequence. The series of pseudorandom data bits is bit wise XOR'd with the plaintext message to produce ciphertext (C). The RC4 cipher provides a simple-to-program encryption and decryption algorithm that is almost ten times faster than the DES algorithm. The IV is communicated to the peer by being placed in front of the ciphertext. Together, the IV, plaintext, and the CRC form a triplet of the actual data that typically makes up a wireless data frame.

The WEP decryption algorithm uses the IV from an incoming message to generate the key sequence necessary to decrypt the incoming message. The receiver has a copy of the same key generate an identical key stream. A bit-wise XOR of the RC4 pseudorandom number generator (PRNG) key sequence with the ciphertext yields the plaintext data. In addition, the integrity check vector (ICV) is used to verify decryption. This encryption can be deciphered with relative ease using open source exploit tools available on the Internet.

As shown in Exhibit 27.3, the output ICV′ is compared to the ICV. In case the comparison results in values of the two vectors that are not equal, it is concluded that the received message has been tampered with and an error indication is sent back to the sending station. The shared secret key that is used to encrypt/decrypt the data frames is also used to authenticate the wireless access client stations.

Lucent pioneered 128-bit WEP development efforts called *WEP Plus*. This extends the WEP key length from 40 bits to 104 bits. Thus, the time taken to crack the WEP key using brute force is extended from a few days

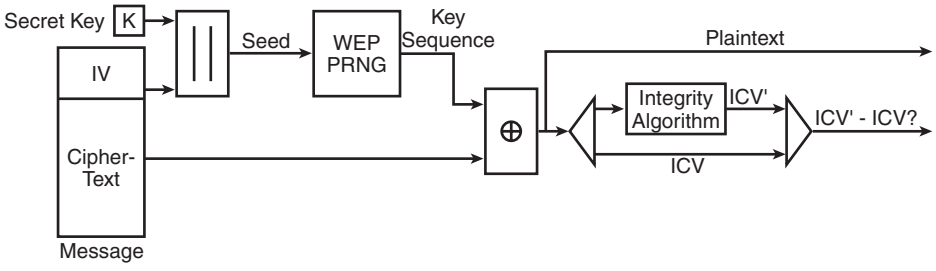


EXHIBIT 27.3 Wireless decryption algorithm.

to approximately 20 weeks. On top of the management problems using static WEP keys there are two serious issues that plague 128-bit WEP. The attacks on WEP are independent of the key length itself. A 24-bit IV is used regardless of whether a 64-bit or 128-bit WEP key is used. It is this IV that is the source of the weakness. The increase in key length does not improve overall security because it is the weakest link — the IV — that is exploited.

Once one plaintext/ciphertext pair is known, then the key stream is known, and thus all plaintext is also known because the key stream is reused. Known attacks on the WEP include IV reuse, as illustrated in Exhibit 27.4. Because the IV values can be reused, the wireless data networks lack replay protection. Also, a small IV space in WEP data is vulnerable to collision attacks.

In the best interest of maximizing efficiency and productivity, rapid deployment of WLANs security becomes paramount. However, this improvement in security does not come without added cost.

The *collision attack* exposes the finiteness or numerical limitation of the IV. This limitation, in turn, leads to identifying the WEP key. Because the IV is only 24 bits long, there are only a finite number of permutations of the IV using RC4 encryption from which to choose. Mathematically, there are only 16,777,216 (2^{24}) possible values for the IV. Some 16 million packets can go by on a heavily used wireless data network. At this point, the RC4-based encryption mechanism repeats the IVs from the already exhausted pool of values. Passive monitoring of the encrypted data and picking from the repeated IVs from the transmitted data stream can allow an attacker to begin the WEP key. Eventually, the necessary amount of data can be gathered, which in turn leads to the compromise of the WEP key.

The *replay attack* is based on the IV and centers around *weak IVs*. In this case, the encryption of data begins with RC4 choosing a random 24-bit number and then combining that number with the WEP key to encrypt the data. However, it has been found that some numbers in the range of 0 to 16,777,215 (2^{24}) do not work well as IVs for the RC4 encryption mechanism. When the RC4 algorithm picks out any of these *weak IVs*, the resulting encrypted data packet can be run through mathematical functions to reveal part of the WEP key. Capturing a large amount of data packets, a malicious attacker can pick out enough *weak IVs* to reveal the WEP key and also compromise the WLAN network's security.

The *table attack* is based on the exploit or decryption of the data captured during transmission assuming that the IV or WEP key is not compromised. This exploit is possible if the transmission contains the IV/key stream in every data packet.

EXHIBIT 27.4 Known Plaintext Attack

The data stream cipher: $C = P \oplus RC4 (IV \mid k)$ and
The plain text cipher: $P = C \oplus RC4 (IV \mid k)$

During the process of WEP-based communications, the key stream is reused such that:

$C1 = P1 \oplus RC4 (IV \mid k)$
$C2 = P2 \oplus RC4 (IV \mid k)$
— — — — — — — — — — — —
thus extending the $C1 \oplus C2 = P1 \oplus P2$

The *broadcast key attack* is based on capturing the key stream data, and deciphering the WEP encryption as wireless data transmission begins with a broadcast key. A compromise or exploit of the WEP key is only possible using very specific types of packets from the data stream. This data can be captured over a period of time for further packet analyses. Because the data packets required for analyses occur very infrequently, a compromise requires a determined hacker and large amount of data.

Managing WEP Keys

The WEP key deployment raises another concern related to key management. When WEP is enabled, per the 802.11b standard, it is necessary to configure each wireless device and type in the proper WEP key. When this configuration is rolled out to a new client setup and the key gets compromised for any reason (or a user leaves the organization, or a user shares the key over the telephone, or someone guesses the password), the key needs to be changed or all data security is lost. This may be a rather trivial effort for a few users on the network, but what if an entire university campus or hundreds of corporate network users are affected? In these cases, changing the WEP key quickly becomes a resources, time, and logistics challenge. This key change can become even more complicated if there are critical production systems that directly impact end users and clients who are accessing the data network.

Wireless access control threats, also termed as wireless access control management (WACM), results in malicious user access into the Intranet (internal data network) rather than limiting user access to the public data segment allowing restricted Internet access.

Recommendations

A fairly easy-to-implement security measure is to turn off the broadcast feature of the SSID. Now the user has to type the SSID into the wireless client. This serves as a deterrent to defend the WLAN against casual wardriving scans. While this safeguard does increase the time to manage the access client, it does not require any additional software integration.

Flaws in the WEP can be overcome using *broadcast key rotation.* As per the 802.11b protocol specification, there are two WEP keys. One encryption key is used to encrypt the individual stream of data between the WAP and the wireless client while the other key is used to encrypt broadcast DHCP or ARP transmission requests. Thus, a WLAN can be made more secure by generating broadcast data encryption keys that have a shorter life in comparison to their counterparts. The network administrator configures an expiration time on the WAP and every time the counter resets, the WAP broadcasts a new broadcast WEP key. In typical WLAN deployments and WAP configurations, the reset times are set to an excess of ten minutes. This provides enough time for attackers to intercept useful wireless data packets that, in turn, are cumulatively required to crack the WEP key. Thus, broadcast key rotation is only effective as part of an overall WLAN security implementation and policy.

The MAC address of a network interface card (NIC) is a unique, 12-digit hexadecimal number used by every card to communicate on the Internet. Because each NIC has its own individual address, the WAP can be configured such that it accepts only one MAC address (assuming that only one legitimate client connection is required). Thus, every other MAC address-based card that does not need to cannot gain access to the data network. This is made possible using a database of MAC addresses that each WAP looks at before establishing a connection to the network. While the filtering of MAC addresses is effective for communication among clients in small networks, it is an administrative challenge to maintain and manage the database for larger data networks in an enterprise environment.

MAC address filtering in itself is not secure. Using freeware or shareware wireless sniffer tools available over the Internet, a malicious user can intercept wireless network data, and extract the MAC address from the data frame communications even if the packets are encrypted. The extracted MAC address can be replaced by a spoofed MAC address to communicate with the WAP — thus defeating the MAC filter-based wireless security.

Secure Wireless Connections and Implementation Options

VPNs are and will continue to be a network access solution for handling secure wireless connectivity. Thus, unauthorized user access to the wired data network can be prevented using a VPN solution. The idea behind

the implementation of this security measure is to consider the WLAN as the equivalent of the Internet. A firewall device separates the trusted data network from the untrusted Internet. The remote users accessing the data network are challenged by the firewall and only allow legitimate users into the data network via an encrypted, secure channel. The same idea applies to the wireless networks. Using the VPN solution, all wireless network traffic is segmented behind a firewall. Each client is then configured with a VPN client and tunneled over the wireless network to a VPN concentrator on the wired network. This security setup uses proven technology to prevent outsiders from gaining access to the wired network.

The process of gaining legitimate access to a wireless network begins with the client boot-up and assignment of an IP address. Once the client has been assigned an IP address, using either static addresses or a DHCP addressing scheme, the client can negotiate a tunnel over the wireless network to begin its data communications. Malicious users also attempt to use the same process, except that they do not gain direct access to the wired network. A malicious user with a valid IP can now communicate with other wireless clients that are configured outside the firewall. Taking this intrusion a step further, the malicious user also has the ability to break into a legitimate user's client, gaining access to the wired data network. It is possible to prevent this by allowing the wireless user to only communicate with the VPN access concentrator. Because available wireless network bandwidth is shared among clients, there are only 11 Mbps available. Piggybacking on a legitimate client can degrade network access speeds significantly, leading to denial-of-service (DoS) attacks on the data network.

The 802.1x standard was ratified in April 2002 by the IEEE. This port-level security enhancement is a new layer 2 (MAC address layer) security protocol that enhances the authentication stage of the wireless secure login process. During the authentication or login process, the wireless device requests access to the WAP. The WAP demands a set of credentials. The device user responds with the credentials that the WAP forwards to a standard RADIUS server for authentication and authorization. RADIUS (Remote Authentication Dial-In User Service) is commonly used to authenticate remote access dial-in users. The exact method of supplying credentials is defined in the 802.1x standard, referred to as the EAP (Extensible Authentication Protocol). While EAP is the main security component of the IEEE 802.11x standard, it is also a flexible authentication development suite that is used to create custom methods of passing user credentials. There are four commonly used EAP methods in use today: EAP-MD5, EAP-Cisco Wireless (also known as LEAP), EAP-TLS, and EAP-TTLS.

EAP-MD5 relies on an MD5 hash of a username and password to securely communicate the user credentials on to the RADIUS server — thus preventing unauthorized users from accessing the wireless data network using the static WEP encryption scheme. This inadequate protection allows malicious users to sniff the wireless data, decrypt the WEP key, and consequently access all the wireless data. In addition, EAP-MD5 does not provide for a means of verifying the authenticity of the WAP. This weakness can be exploited by a determined hacker who has configured the rogue access point to appear as a legitimate source of data communication. Thus, EAP-MD5 is considered the least secure of all the common EAP standards. Furthermore, the EAP-MD5 authentication standard offers no additional key management or dynamic key generation.

EAP-Cisco Wireless, or LEAP, is an authentication standard developed by Cisco in conjunction with the 802.1x standard, and is the basis for much of the ratified version of EAP. Like EAP-MD5, LEAP accepts the login username and password from the wireless device and transmits the data to the RADIUS server for authentication. What sets LEAP apart from EAP-MD5 are the extra features it adds over what is explicitly called for within the 802.1x/EAP specification. When LEAP authenticates the user, one-time WEP keys are dynamically generated for that session. This means that every user on your wireless network is using a different WEP key that no one knows, not even the user. Also, you can combine this with the session timeout feature of RADIUS to have the user re-log in every few minutes (handled behind the scenes; the user does not have to type in anything) and create new dynamic WEP keys that change every few minutes. Setting your RADIUS server up this way effectively nullifies current attacks on WEP because the individual keys are not used long enough for an attacker to crack them. LEAP also stipulates mutual authentication of client-to-AP and AP-to-client above that strictly called for in 802.1x. This prevents a hacker from inserting a rogue AP into your network and fooling your wireless clients into thinking it is a secure connection.

LEAP does not come without its own drawbacks. MS-CHAPv1 authenticates both the WAP and the client by passing on the user log-on credentials. But the MS-CHAPv1 has a known set of vulnerabilities. The authentication protocol can be compromised with the right set of hacker tools. While there are no known instances of LEAP being compromised, MS-CHAPv1 is a weakness. The second drawback in implementing LEAP is that the protocol only works on Cisco end-to-end networks. While Cisco has added LEAP capabilities

to its wireless client, other vendors are working to add LEAP to their wireless client software to allow non-Cisco network cards in established LEAP implementations.

EAP-TLS is outlined in RFC 2716 and implemented by Microsoft. Instead of username/password combinations, EAP-TLS uses X.509 certificates to handle authentication. While the EAP-TLS relies on Transport Layer Security, the IETF is also drafting a standard such that the Secure Socket Layer (SSL) communications can send PKI-specific information into the EAP data buffer. Like LEAP, EAP-TLS provides dynamic one-time WEP key generation, and WAP authentication from and to the wireless client. EAP-TLS is platform independent, supporting a client written for Linux and Windows operating systems (except Windows CE).

Implementing EAP-TLS does not come without its limitations. In case the organization does not already have PKI in place for handing out certificates to trusted parties, there is a steep learning curve to understanding whether the PKI solution should be VPN-centric, authentication-centric, or network-centric, as well as how the solution of choice can be implemented. The only way to easily deploy EAP-TLS is to use an Active Directory (AD) solution that integrates with a Microsoft Certificate Server with wireless clients that only log in to the AD. All digital certificates are published to the user accounts in the AD. If Open LDAP or Novell Directory Services are used, digital certificates for the user accounts cannot be used. The RADIUS server has no standards-based mechanisms to distinguish if the digital certificate being exchanged is indeed a valid certificate. In addition, the identity exchange is completed using cleartext communications before the digital certificates are exchanged. This weakness can be exploited by *passive attacks*, allowing for footprint or fingerprint analyses by a malicious user.

As an alternate authentication option to EAP-TLS and overcoming the PKI implementation challenges, Funk Software developed the EAP-TTLS. As part of a two-step process, the first step is the authentication step. The WAP identifies itself to the WAP client with a server certificate. In the next step, a TLS tunnel is established, allowing for authentication of the client to the client with a digital certificate. The users now send their credentials, that is, the username/password format. These credentials are also referred to as the *attribute-value pairs*. In turn, the EAP-TTLS sends the user credentials through a number of administrator-specified challenge–response mechanisms, including PAP, CHAP, MS-CHAPv1, MS-CHAPv2, PAP/Token Card, or EAP.

Recommendations

A fairly easy-to-implement security measure is to turn off the broadcast feature of the SSID. Now the user has to type the SSID into the wireless client. This does serve as a deterrent to defend the WLAN against casual wardriving scans. While this safeguard does increase the time to manage the access client, it does not require any additional software integration.

Flaws in the WEP can be overcome using *broadcast key rotation.* As per the 802.11b protocol specification, there are two WEP keys. One encryption key is used to encrypt the individual stream of data between the WAP and the wireless client, while the other key is used to encrypt broadcast DHCP or ARP transmission requests. Thus, a WLAN can be made more secure by generating broadcast data encryption keys that have a shorter life in comparison to their counterparts. The network administrator configures an expiration time on the WAP and every time the counter resets, the WAP broadcasts a new broadcast WEP key. In typical WLAN deployments and WAP configurations, the reset times are set to an excess of ten minutes. This does provide enough time for attackers to intercept useful wireless data packets that, in turn, are cumulatively required to crack the WEP key. Thus, broadcast key rotation is only effective as a part of an overall WLAN security implementation and policy.

The MAC address of a network interface card (NIC) is a unique, 12-digit hexadecimal number used by every card to communicate on the Internet. Because each NIC has its own individual address, the WAP can be configured such that it accepts only one MAC address (assuming that only one legitimate client connection is required). Thus, every other MAC address-based card that does not need to cannot gain access to the data network. This is made possible using a database of MAC addresses that each WAP looks at before establishing a connection to the network. While the filtering of MAC addresses is effective for communication among clients in small networks, it is an administrative challenge to maintain and manage the database for larger data networks in an enterprise environment.

MAC address filtering in itself is not secure. Using freeware or shareware wireless sniffer tools, available over the Internet, a malicious user can intercept wireless network data, and extract the MAC address from the data frame communications even if the packets are encrypted. The extracted MAC address can be replaced by a spoofed MAC address to communicate with the WAP, thus defeating the MAC filter-based wireless security.

Conclusion

While IEEE 802.11x continues to be ratified with data security improvements, there are basic configurations and implementations that can assist with securing the wireless data network, including:

- Change the SSID on a regular basis.
- Change the passphrase for the SSID management.
- Do not allow the SSID to be broadcasted.
- Use 802.1x for key management and authentication.
- Configure WEP for the highest level of data encryption available at the WAP.
- Set the currently established idle session to timeout every ten minutes or less.
- Rename the default SSID name so that it does not provide information regarding the network.
- Set your WAP to be a *closed* network and set the authentication method to be *open*.
- Rotate the broadcast keys every five to ten minutes, depending upon the data sensitivity requirements.
- If feasible, configure the wireless network behind its own routed interface such that data communications can be shut off in case the need does arise.
- Enforce MAC address validation to ensure that unauthorized or nonregistered devices, when connected, do not gain network access.
- Maintain and enforce access policies such that unauthorized data access is denied.
- Prevent wireless data signal emanations by planning to relocate the WAP antenna to a physical area that mitigates malicious scanning.

While the 802.11i wireless networks continue to evolve, the WEP/TKIP will be replaced by the new encryption scheme called the Advanced Encryption Standard – Operation Cipher Block (AES-OCB). This new encryption standard is a version of the AES that has been adopted by the U.S. Government as the replacement for the 3-DES encryption standard. Furthermore, implementation of the AES-OCB encryption standard is expected to be stronger than the current WEP/TKIP.

References

Your 802.11 Wireless Network Has No Clothes, Arbaugh, W.A., Shankar, N., and Wan, Y.C.J., 2001

Intercepting Mobile Communications: The Insecurity of 802.11, Borisov, N., Goldberg, I., and Wagner, D., 2001

Weaknesses in the Key Scheduling Algorithm of RC4, Fluhrer,S., Mantin, I., and Shamir, A., 2001.

Wireless LAN Security: A Short History, Gast, M., 2002.

Wireless Security Blackpaper, Dismukes, T.A.

Wireless LAN: Security — WEP, Katholieke Universiteit Leuven., 2002.

802.11 Wireless Networks: The Definitive Guide, Matthew Gast, O'Reilly, 2002.

Fahey, D. and Smith, E., "Wireless Neworks: Detecting/Exploiting/Securing," SANSFIRE 2002.

28

Wireless Security Mayhem: Restraining the Insanity of Convenience

Mark T. Chapman, MSCS, CISSP, IAM

It is just past supper time in a small town in 1953. The family gathers around the black-and-white television to watch the only show on the only station in town. The father fusses with the controls and the rabbit ears to get the clearest signal. Successful, he rushes to his chair. At the exact moment he sits down, the picture turns to static. He gets up with a determinedly authoritative smile. He quenches the urge to curse aloud out of fear that it may upset the magical device. Once he gets to the TV, he merely reaches toward the controls and the picture becomes "clear as a bell." He slowly backs up to his chair — not daring to take his eyes off the TV. He crouches. His hand blindly finds the arm of the chair. He leans back… further… almost sitting… and BAM! The picture disappears. "This fool thing has a mind of its own!"

Outside, two boys are slowly walking their dog on the sidewalk. They can barely contain themselves at the ingenuity of their latest mail-order kit. They simply close the circuit on the FM transmitter to willfully jam the TV signal for several hundred feet. The art, of course, is to time it with the animations of the unsuspecting neighbor who they clearly see through the picture window. "Now *that's* television!"

While television is an entertaining example, wireless communication technologies pose significant challenges to information security practitioners. The convenience of wireless connectivity is compelling. The cost is trivial. The setup time and knowledge required to do a "default installation" is nominal.

Long before the threats from the Internet are restrained with any more than a short-sleeved straightjacket, everyone seems to be deploying inexpensive and easy-to-set-up wireless access. To combat the risks, there are enough wireless security solutions to make an information security practitioner's head spin.

For the purposes of this chapter, "wireless" refers to any communication technology that uses radio waves or similar techniques to transmit information through the air. The simplicity of this definition forces the information security practitioner to consider more than just the most popular wireless networking protocols of the moment.

The challenge is to look beyond the alphabet soup of wireless security protocols and standards by providing a set of reasonable guidelines for mitigating threats associated with many kinds of wireless access. From cordless phones to cellular, from garage door openers to car keys, from 802.11b to 802.11x, the time is here and now to take a reasonable approach toward understanding and managing the risks associated with the convenience of wireless connectivity.

An information security practitioner will consider the following:

- *Culture of convenience:* what price will people pay for availability?
- *Purpose of the network:* how to apply the concepts of least privilege.
- *Policy:* which policies are most important when it comes to wireless solutions?

- *Range of network:* how far does a network reach in space and time?
- *Cryptography:* what role should it play?

The Culture of Convenience

The convenience of wireless solutions often overshadows any concerns about information security. Many people deploy wireless solutions with almost no consideration for the confidentiality, availability, or integrity of the information exposed due to the unique aspects of wireless connectivity.

People are not accustomed to having any influence on the effectiveness of security controls. For example, there are few options other than to trust the automobile manufacturer with the appropriateness of the security level of car keys. The culture of convenience now demands that most new cars come with a remote entry system. The owners must accept whatever level of security the company uses for the wireless solution.

On the surface, this is reasonable. Any key, it seems, is simply a deterrent to keep the honest people honest. The "rock-through-the-window" approach always works if someone simply wants to steal a purse from the front seat. If someone is savvy enough and motivated enough to hot-wire a car, or tow it, then the shape of the physical key or the secrecy of the wireless entry system might not even slow them down.

The big question is whether or not the convenience of a remote entry system poses new threats in comparison to legacy car keys? In the past, if a driver lost her keys at the mall, it was an inconvenience. The chances that someone would steal something from her car were negligible because there were simply too many cars in the crowded parking lot.

Essentially, the location of her car is a secret. Thanks to the culture of convenience, the added "locate" function on a remote entry system often makes it trivial for someone to find a car. In the context of this chapter, it is the confidential information about the location of the car that has been compromised.

From an information security awareness perspective, many people understand that they should not label their car keys. It is less clear to define the reasonable measures someone should take with respect to the very convenient and very helpful remote entry systems.

All too often, convenience is the opposite of security.

Consider a typical home Internet user, Alice. Alice left the woes of dial-up networking far behind for the convenience of inexpensive, always-connected broadband. When she took that big step, the desktop computer in her den was immediately exposed to threats from the Internet "24/7" instead of just the limited hour or two each day in the dial-up era. To address the perceived increased threats from the Internet, Alice purchased anti-virus software, installed a personal firewall, and regularly applies patches to her system. For the purposes of this chapter, she takes sufficient reasonable measures to protect her computer.

A few months into "broadband heaven," Alice decides that she would rather use a laptop to connect to the Internet from her couch — or better yet, from her back porch. After spending less than $100, her laptop is able to connect to the Internet from anywhere in the neighborhood.

Alice has heard that wireless networks are "not secure." To be honest, she does not care. Alice continues to take reasonable measures to protect her laptop computer. Why would she be motivated to protect this separate thing called the "network"? From her perspective, it is all just the "Internet" anyway. If she can keep the whole world of Internet crackers out of her system, then how difficult could it be to keep the neighborhood kids at bay?

Convenience wins, even if security is a concern.

Alice's husband, Bob, is even less concerned about securing his home computer that he uses primarily for online gaming. After realizing the freedom and convenience of his unrestricted access at home, Bob puts pressure on the people at work to give him wireless access. On the surface, it is a compelling argument that he has a more convenient solution at home than at work.

The culture of convenience demands extra functionality that seems inexpensive and easy to use. Seldom, if ever, is information security a primary consideration.

What Is the Purpose of the Wireless Solution?

All wireless technology solutions should have a purpose. Is the purpose to allow café patrons to be able to surf the Internet? Is the purpose to allow doctors and nurses to access patient information in a hospital wing? Is it to allow teachers to access the grading system? Is it for building access? Tracking products in a warehouse? Is it to try out the latest technology? Is the purpose of the solution to save money over wired alternatives?

The sole purpose of a wireless solution is often the purpose of convenience. For an information security practitioner, it is important to determine the clear purpose of any wireless solution. Performing the following six steps may help clarify the purpose:

1. In one sentence, clearly define the business purpose of the wireless solution. Use terms that emphasize the desired results. Here are some examples:
   — To save money on telecommunications costs by replacing the current ISDN system with a point-to-point microwave solution.
   — To reduce the amount of time spent off the factory floor due to personal phone calls by providing cordless phones.
   — To allow students to access online coursework from any location on campus by providing full 802.11b wireless coverage.
   — To make Bob feel as if his office IT department is as technically competent as his wife Alice.
2. Identify the critical success factors for the solution. These may include specific cost reductions, productivity improvements, increased customer satisfaction metrics, or anything else that is measurable. The idea is to drill down several levels deeper than the one-sentence business purpose. It may include specific performance requirements of the solution, such as minimum bandwidth or availability constraints. Examples include:
   — Reduce telecommunication costs by 40 percent in the next six months.
   — Decrease break times by five minutes.
   — Increase online test scores by five percent.
   — Provide at least 2 MB throughput within 100 yards of any campus building.
   — Provide compatibility with any 802.11b device that supports 40-bit encryption or better.
3. Define who the targeted audience is for the solution. Be specific.
   — All teachers and faculty at the downtown campus should have access to the administrative network. All students should have Internet-only access. The idea is not to provide Internet access to the general public.
   — Any employee who is on break is allowed to use the cordless phones for local personal calls.
   — This solution is just for Bob and his handheld computer.
4. Determine who is accountable for the implementation and ongoing solution management. What is the expected level of service from these people?
   — The District Technology Coordinator is accountable for all on-campus networking services. To reduce the help desk calls, all students will receive an e-mail instruction at the start of the semester about how to connect to the network. Given the limitations of the current help desk and the variety of devices, it is not expected to receive one-on-one consulting for personal equipment. Teachers and faculty will receive the highest priority from the help desk.
   — The janitor will make certain that all cordless phones are returned to their chargers at the end of every shift. If any phones stop working during the shift, the supervisor will create a work-order for replacement within 72 hours by the telecommunications department.
5. Clearly define the owner of any hardware devices, software, or information that uses the wireless solution. It is appropriate to refer to relevant policies, guidelines, and standards.
   — The teachers and faculty will be allowed to connect the laptops that have been provided by the school to the administrative wireless network. Any information on the administrative network is the property of the school. The teachers, faculty, and students will be allowed to connect to the Internet with any device that meets the minimum compatibility requirements, including personal equipment. The school reserves the right to monitor any network traffic on any wireless or wired network.
   — The only devices that can connect to the hospital network are those provided and supported directly by the hospital.
6. Apply the concept of least privilege to the purpose of the wireless solution. The concept of least privilege is to grant the minimal amount of access required to achieve a goal. In this case, the goal is to minimize the scope of the wireless solution while maximizing the desired results. Least privilege is least likely to occur in a strong culture of convenience.

— If only students should have access to the wireless network, specify that the network is not open to the public.
— If doctors are supposed to be able to access areas of the network that nurses cannot, specify that difference as part of the purpose of the network.
— If only 100 yards of coverage are required between buildings, then specify that it is not to exceed 200 yards with the standard antennas.
— If the cordless phones are for occasional use, then let them share one extension.

Defining the purpose of a wireless solution is the first and most important step in setting clear expectations. The message to all involved is to increase awareness that the organization must strike a balance between convenience and security.

What Are Some Common Threats?

The risks associated with a wireless solution for personal use may be quite different from risks within an organization. Something as simple as an automatic garage door opener could pose a negligible incremental security risk at home because there are easier ways to break into a garage, such as entering through a window. The information security implications of the wireless communications in this environment are almost irrelevant as long as the neighbor's opener does not cause an inconvenience by opening every door on the block! In a different context, the same solution could expose the valuable content of a warehouse to theft. A garage door opener could even pose a national security threat if it uses the same frequency as some seemingly unrelated critical infrastructure component. It is imperative to perform some level of risk assessment for each wireless network.

Threats generally fall into the familiar categories of Confidentiality, Integrity, and Availability, sometimes referred to as the CIA triad. Additionally, it may be appropriate to consider a fourth category, called "Liability."

The goal of Confidentiality is to keep private information private. The idea of Integrity is to have a high confidence that nobody has purposefully or accidentally tampered with the data. The threats to both of these areas are well-known privacy and authentication issues. The unique threats from the wireless networking side are often misunderstood. For example, what good does it do to 3-DES-encrypt and RSA-sign every packet that goes across the wireless network — just to have it decrypted by the access point for plaintext travel across the wired network? What good does it do to use 3-DES encryption if the key is public knowledge? Could there be confidentially issues if someone brings his own bandwidth to work on some of the new cellular phones and personal digital assistants?

The goal of availability is that the information must be available when it is needed. With wireless solutions, this can be much more difficult than with wired counterparts. In a wired solution, it is usually crystal clear who owns the wire. In a wireless solution, almost everyone is sharing public radio frequency bands. Very few organizations have the ability or the need to license an RF band. What this means is that the availability of the most common wireless solutions is at risk by law! The FCC and similar organizations state that "This device must accept interference…." Cordless phones, 802.11b, microwave ovens, and loads of other solutions legally share the same channels. Additionally, there is not much to prevent nearby entities from causing interference by legally using exactly the same solutions.

An organization is lucky if the threats only come from law-abiding citizens. Consider the example of the two boys with the illegal TV-jamming equipment. If high availability is critical to the success of the wireless solution, it is likely that an alternative solution will be required.

Outside the CIA triad, there is one more area of threats that applies to wireless solutions: Liability. Remember Alice's attitude about her wireless Internet connection at home? She protected her laptop from a CIA perspective from the Internet and the neighborhood kids. She did not protect her wireless network as she was unconcerned or uninformed about potential liability issues. In the case of home networks, this may be acceptable for now. For organizations, there could be significant issues if someone uses the wireless network to cause harm elsewhere.

Take, for example, a school district. Assume that it has properly separated the wireless computing lab from the administrative systems. Assume that confidentiality is not a problem — because there is no sensitive information on the lab machines. All students log in as "LAB1" — so predators cannot identify them. Assume that integrity is not an issue. Even if someone changes the information, it may not be a problem

— as the purpose is to simply learn how to deploy wireless networking technology. Availability is not a problem, as the purpose is to learn to make it work. In this case, liability may still be an issue. Some possible scenarios include:

- Someone is using the wireless network to store illegal software on the school's machines. It is not required that the lab be connected to the Internet; a wireless-accessible file-store can easily be accessed from a public place outside the building.
- If the wireless network is connected to the Internet, someone might use it to hack a bank or to provide some interesting, although immoral or illegal, Web services.

A softer side of the liability issue is that of credibility. Several organizations have lost competitive advantage due to avoidable situations, such as being mentioned on the evening news in a story about "war driving," which is the act of driving around with an antenna looking for open wireless networks.

Wireless "war driving" by itself is not a new concept. In 1953, the two boys with the TV transmitter were "war"-walking the dog. By 1975, they graduated to "garage-door-opener testing." The main difference now is the level of sophistication and the coordination of efforts. This phenomenon is now almost scientific, with online nationwide street-by-street maps of the wireless world.

It is critical to identify what needs to be protected with respect to the CIA triad and liability. Does information need to be protected in transit or in storage? Is there a strong need for authentication, or is anonymity critical to success? Is non-repudiation— the ability to have an objective third party confirm or deny that an event has happened — an issue with the solution? Are there timing issues with respect to revoking access on short notice? What are the expectations for equipment failure — especially if the equipment is personal equipment? Do users have a reasonable sense of privacy, or do they expect their personal devices to be magically protected by, and from, the organization's network?

There are several threats that are much more probable, given the very nature of a wireless solution. Awareness of these types of attacks may help organizations that have been quite conservative in adopting wireless solutions. The very existence of wireless solutions changes the threat profile, whether or not an organization is implementing the technology.

Consider the "binocular attack," which works with technical and nontechnical resources. The attack is to simply use a run-of-the-mill pair of binoculars or a telescope to observe someone's monitor or papers on the desk. This is, technically, "wireless." At about the same level as "dumpster diving" or "social engineering," it is much cleaner and requires much less skill. Close the curtains on sensitive information.

Another example comes from the physical layer. In the same way that finding a remote access key to an automobile may pose an information security threat, other threats abound as more and more devices are part of wireless solutions. It is easier to steal a building access key than it is to reverse-engineer one!

Common threats also include the creative misuse of available technology. Consider a readily available and lesser-known device called a wireless serial cable. The concept is easy. On a manufacturing floor, there are a series of measurement devices, such as scales, calipers, etc. For years, these have run on an RS-232 or RS-432 interface. There are strict limitations as to the effective length of these cables. As expected, there are now several wireless solutions to the rescue! One end of the "wireless cable" is an actual cable that connects to the computer. The other end is an antenna. A similar device with an antenna and a serial cable connects to the scale. The computer and the scale are not aware that there is anything but an actual cable directly connecting them.

One of the author's favorite demonstrations is a wireless attack against a firewall. The COM port is often assumed to be physically secure; thus, minimal authentication is required to access the configuration files. The idea of the attack is to plug one end of a wireless serial cable into the COM port of the firewall. The other end of the cable can be three miles away and connected to a laptop. Yes, it does require initial physical access. Awareness of this attack is one of the best ways to prevent it. It does not take much creativity to come up with similar attacks on desktop machines, printers, and other devices.

The final category of threats falls under the familiar "Trojan horse" and "man-in-the-middle" attacks. To make wireless solutions as convenient as possible, the designers often configure the devices to automatically connect to the clearest signal. Whether a cellular phone or a laptop computer, it is often simple to set up an unauthorized access point. When the convenient device attempts to authenticate to the rogue access point, it may share some secrets about how to connect to the real thing.

What Is the *Range* of the Problem?

Wireless connectivity is not inherently less secure than wired alternatives — with the one notable exception of *range control*. Most wired solutions do not force the use of encryption or require users to authenticate before communicating. Many wired technologies are directly or indirectly connected in some way to the Internet or the phone system — both of which may pose significant threats to information. Nonetheless, it does seem that wireless solutions pose a higher security risk than wired alternatives.

Range refers to both the physical range and the temporal range. Physical range is easy — how far do the radio signals travel? Temporal range is also easy — what is the time that the system must be available — thus, exposed to threats?

The most obvious example of a threat due to physical range is the infamous "parking lot attack." The idea is that someone can access wireless communication networks from outside the building. The assumption is that a wired network is more secure because the physical range of the wires is known.

To define this concept, consider the *physical range* of the following networking technologies:

- Sneaker net (uses floppy disks to share information between computers)
- Local area network, LAN (Ethernet or Token Ring network of computers)
- Bulletin boards (dial-up terminal emulation and file transfer)
- Dial-up networking (connect to the LAN from home)
- Internet (terminal emulation, file transfer, and more)

Each of the above technologies has an expected physical range. For example, if most networks were of the "sneaker net" variety, then the range would seem quite limited. Past experience with early virus infections tells us that even a sneaker net may have a global reach.

It is a mistake to assume that the range of a wireless solution is known. It is important today to assume that the whole world can see and attempt to modify wireless communications. By designing countermeasures with this assumption in mind, it makes wireless and other technologies much easier to contend with.

Exhibit 28.1 shows a set of assumptions about the physical range of particular wireless technologies. These ranges can be extended using antennas, either by the controlling organization or by thrifty attackers. They also can be extended by bridging different technologies, such as connecting an 802.11b access point to a cellular phone for Internet access.

There are a few different components to temporal, or time-based, range control. In the example of Alice's wireless Internet connection, she could reduce her liability exposure by simply turning off her access point when she is not using it. Another example is the timely revocation of access rights to a device that has been reported stolen.

Exhibit 28.2 categorizes examples of technical and nontechnical countermeasures by physical and temporal range. The purpose is to help the information security practitioner consider the effectiveness of countermeasures with respect to the unique range control characteristics of wireless solutions.

EXHIBIT 28.1 Physical Range of Wireless Technology

Physical Range	Wireless Technology
Local	Infrared Bluetooth™ Wireless keyboards and mice Cordless phones
Regional	802.11 Special-purpose radio links (wireless serial cables?) Family radio systems (FRS) or citizens band radio
Global	Cellular phone (Internet access anywhere) FM/AM/shortwave radio Satellite communication

EXHIBIT 28.2 Range Control Countermeasures Matrix

Range Control Countermeasures Matrix		Control: Technical	Control: Nontechnical
Range	Physical	Radio spectrum analysis for rogue access point detection. Layer 2 or layer 3 device detection, (i.e., look for new MAC addresses)	Acceptable use policy Human review of new devices detected
Range	Temporal	Password expiration Certificate revocation Time-based access control lists Time-based authentication protocols Exception reports	Acceptable use policy Employee add/move/terminate procedures Human review of exception reports Convenience of high availability

Which *Policies* Can Help?

Information security policies, standards, and guidelines are key components in an information security management system. It is critical to define reasonable expectations for the mutual benefit of the users and the owners of the network. There are several policy elements that address some of the unique characteristics of wireless solutions.

One of the biggest differences in wired versus wireless solutions is mobility. Information, whether written or voice, is no longer constrained to the desktop or the office. It is imperative to inform users about the risks associated with enhanced mobility. Acceptable use policies are critical on any network. A wireless acceptable use policy defines the expectations for the reasonable use and protection of information. Consider elements such as when to use encryption and when to report a stolen or lost device.

Wireless-enabled devices are becoming so inexpensive that many people can afford to bring their own devices to work. To avoid liability, consider a "use at your own risk" policy for personal devices connecting to any company-provided wireless solutions. Be clear as to the level of protection that is or is not provided, such as firewall, virus protection, Internet content filtering, encryption of data in transit, etc. Include clear expectations as to who has a right to view or monitor the information as it travels across the organization's network.

Another interesting area of concern is that people can afford to bring their own wireless solutions to work. "Rogue access points" are but one example — where individuals extend a wired network by adding inexpensive and unauthorized access points. Be certain that there are strict policies regarding unauthorized wireless solutions. The best policy is to simply disallow any unauthorized wireless solutions from extending the range of the network. An authorization procedure should be put in place for the likely exceptions.

People can afford their own wireless devices and their own wireless access points. Additionally, people can afford their own wireless connectivity. For example, it used to be that everyone used the organization's phone system. Many times, there were acceptable use policies to cover items such as personal long distance calls. Procedurally, some organizations reviewed phone logs by extension to determine productivity loss due to personal phone calls. Today, employees bring their own cellular phones to work.

In the near future, people will bring their own broadband Internet access to work. This will limit the effectiveness of proxy server logs, filtering services, firewalls, and user activity measurement tools. Many organizations will want to adopt the passenger airplane policy of turning off all portable electronic devices. One reason is productivity. Another reason may be due to the interference that these devices may cause with other wireless solutions. A common example is found in the "no cell phones" signs in hospitals.

Finally, it is imperative to define policies and standards that may limit liability exposure for the wireless solution itself. For example, an organization with minimal CIA requirements may choose to require simple encryption to avoid being misconstrued as a "free network" for public use.

Where Does *Cryptography* Fit?

Wireless information security has less to do with wireless-specific encryption protocols than it might seem. Although properly implemented cryptographic protocols may be a necessary component in certain wireless solutions, wireless encryption protocols by themselves are seldom sufficient to provide adequate protection.

Cryptography can help with confidentiality and integrity. Confidentiality can be enhanced by encrypting or scrambling the information stream. Integrity can be enhanced by using authentication protocols to identify the users, clients, services, and data streams.

With rare exceptions, availability is unlikely to be enhanced using cryptographic techniques. Liability can be reduced through the concept of non-repudiation — where cryptographic techniques provide objective evidence that someone performed a particular action.

There is a wealth of documentation on the weaknesses of particular protocols. It seems there are countless standards-based or proprietary solutions that promise to fix the problems. "Get your silver bullets here!" Most of these solutions address well-known problems. Take the classic problem of key distribution. This problem, which is thought to be effectively solved with mathematical ingenuity, could be summarized by saying that it is difficult to share and distribute passwords. From a cryptography perspective, there are several approaches to solve this problem. From a practical perspective, the choices made often invalidate the effectiveness of the math. For example, if an organization uses a single WEP key to access the wireless network, one might ask how secret the key is if 10,000 people know it?

End-to-end encryption is often more important than the key length, block length, or other metrics that describe the relative strength of an encryption protocol. When evaluating cryptographic solutions, it is best to consider an end-to-end solution that is independent of the media. For example, in many cases, a Secure Socket Layer (SSL) connection or a virtual private network (VPN) tunnel is good enough for the Internet. Similar end-to-end authentication and encryption solutions provide more comprehensive coverage compared to wireless-only solutions.

Cryptography does play an important role in the security of wireless solutions. The main problem is that people do not really understand the limitations of cryptography. A popular physical example is to consider a lock on an office door. Everyone knows how to use a key to open the door. Fewer people understand how the tumblers work inside the lock. Most people understand that a more expensive lock may be more difficult to pick. The most important thing is that people recognize the limited effectiveness of any lock if it is installed on a glass door.

Cryptography works the same way. Most people can use a key. Fewer people need to understand the internal workings. Most people can understand that some attributes of encryption, such as key length, provide a higher level of protection. The challenge is to help recognize if the cryptographic "lock" is on a "steel door" or a "glass door."

Conclusion

The convenience of wireless connectivity is compelling. Despite known threats to the confidentiality, integrity, and availability of information, the demand is increasing for wireless communications. An information security practitioner should take a careful look at the purpose of a wireless solution, address common threats, implement reasonable policies, understand the concepts of range control, and carefully select an appropriate level of cryptography. Although it might seem crazy, it is possible to manage many of the risks associated with the tools designed for the culture of convenience.

References

William A. Arbaugh, Narendar Shankar, and Y.C. Justin Wan, Your 802.11 Wireless Network Has No Clothes. Internet, March 2001. http://www.netsys.com/library/papers/wireless.pdf.

Ross Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems.* John Wiley & Sons, New York, 2001.

Open ssh Project Home Page. Internet, November 2002. http://www.openssh.org/.

Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd edition.* John Wiley & Sons, New York, 1996.

Wi-Fi Alliance Home Page. Internet, November 2002. http://www.weca.net/.

Weaknesses in the Key Scheduling Algorithm of RC4. Scott Fluhrer, Itsik Mantin, Adi Shamir. http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf.

The Unofficial 802.11 Security Web Page. http://www.drizzle.com/~aboba/IEEE/.

Netstumbler Home Page. http://www.netstumbler.org/.

29

Wireless LAN Security Challenge

Frandinata Halim, CISSP, CCSP, CCDA, CCNA, MSCE and Gildas Deograt, CISSP

The WLAN (wireless local area network) is getting more popular due to its simplicity and flexibility. In today's computing era, wireless installation is very easy and people are able to connect to a network backbone in a very short timeframe. Undoubtedly, wireless interconnection offers more flexibility than a wired interconnection. Using a wireless interconnection, people are able to sit in their preferred spot, step aside from a crowded room, or even sit in an open-air area and continue their work there. They do not have to check any wall outlet and, moreover, they do not have to see any network cables tailing to their device.

Following the proliferation of wireless technology, many Internet cafés started to offer a wireless Internet connection. Internet access areas are available in airports and other public facilities. People can also access their data in the server using their handheld devices while they walk to other rooms. Past visions of such wireless network technology have now become a reality.

However, in addition to the wide use of wireless technology throughout home-user markets, easily exploitable holes in the standard security system have stunted the wireless deployment rate in enterprise environments. Although many people still do not know exactly where the weaknesses are, most have accepted the prevailing wisdom that wireless networks are inherently insecure and nothing can be done about it. So, is it possible to securely deploy a wireless network in today's era? What exactly are the security holes in the current standard, and how do they work? Toward which direction will wireless security be heading in the near future? This chapter attempts to shed some light on these questions and others about wireless networking security in an enterprise environment.

A WLAN uses the air as its physical infrastructure. In reality, it is quite difficult to capture a complete set of traffic on the Internet because each network packet may go through different paths. However, some parties, like ISP employees or intelligence organizations, are likely to possess such ability. Moreover, people around the wireless neighborhood may be within the signal coverage area, and hence they can capture the WLAN traffic. Therefore, physical security in wireless technology is no longer as effective as it is on wired technology because there are no physical boundaries within wireless technology.

There are many new risks concerning WLANs, wherein certain security measures must be taken to preserve the confidentiality, availability, and integrity of information passing through a wireless interconnection. Hence, the level of convenience offered by WLAN technology will consequently be adversely affected. In fact, the only security offered by WEP as the current security feature defined in the 802.11 standard also has its own vulnerabilities. Furthermore, the easiness of installing a rogue (unauthorized) access point within a wireless system also introduces a new risk of backdoors to a system that bypass the perimeter defense system (e.g., firewall).

WLANs offer many challenges and this demands that security professionals creatively invent a defense-in-depth solution to answer those challenges. International standards organizations also have an increasing challenge to provide a secure and robust standard to the industry.

WLAN Overview

In 1997, the IEEE established a standard for wireless LAN products and operations based on the 802.11 wireless LAN standards. The throughput for the 802.11 standard was only 2 Mbps, which was below the IEEE 802.3 Ethernet standard of 10 Mbps. To make the standard more acceptable, IEEE then ratified the 802.11b standard extension in late 1999. The throughput in this new standard has been raised to 11 Mbps, thus making this extension more comparable to the wired equivalent.

The 802.11 standard and its subsequent extension, 802.11b, are operating under the unlicensed Industrial, Scientific, and Medical (ISM) band of 2.4 GHz. As with any of the other 802 networking standards, the 802.11 specification affects the two lower layers of the OSI reference model — the physical and data-link layers. There are some other devices operating in this band, such as wireless cameras, remote phones, and microwave ovens. In operation, the 802.11 standard defines two methods to control RF propagation in airwave media: frequency hopping spread-spectrum (FHSS) and direct sequence spread-spectrum (DSSS). DSSS is the most widely used; it utilizes the same channel for the duration of transmission. The band is divided into 14 channels at 22 MHz each, with 11 channels overlapping the adjacent ones and three nonoverlapping channels.

802.11 Extensions

Several extensions to the 802.11 standard have been either ratified or are in progress by their respective task group committees within the IEEE. Below are the three current task group activities that affect WLAN users most directly.

802.11b

802.11b operates at 2.4 GHz with a maximum bandwidth of 11 Mbps and is the most widely used implementation today. Both 802.11a and 802.11b standards have at least 30 percent of protocols overhead and errors. The 802.11b extension increases the data rate from 2 Mbps to 11 Mbps.

802.11a

802.11a is a WLAN standard that operates at 5.2 GHz with a maximum bandwidth of 54 Mbps. Because the frequency is higher, the effective transmission distance in 802.11a is consequently shorter than in 802.11b. Due to this disadvantage, many vendors try to adopt both technologies in order to derive the greatest benefit from them.

802.11g

802.11g is the compatibility standard between 802.11a and 802.11b, using the 2.4-GHz band and also 5 GHz while supporting 54-Mbps data transmission. This makes the standard backward compatible with 802.11b. It is also interesting because the 802.11b backward compatibility preserves previous infrastructure investments.

Other Extensions

- 802.11i deals with 802.11 security weaknesses, and, as of this writing, has not been completed.
- 802.11d aims to produce 802.11b, which works at another frequency.
- 802.11e works by adding a QoS capability to enhance audio and video transmission on an 802.11 network.
- 802.11f tries to improve the roaming mechanism in 802.11 to offer the same mobility as cell phones.
- 802.11h attempts to provide better control over the transmission power and radio channel selection to 802.11a.

Wireless LAN Working Mode

There are two possibilities of how to operate WLAN network access: ad hoc mode (Exhibit 29.1) and infrastructure mode (Exhibit 29.2). Ad hoc mode is used for PC-to-PC direct connection.

The ad hoc mode is simply multiple wireless clients in communication with each other as peers in the range of a radio signal. It is spontaneously created between the wireless clients. All processes are handled by a station, as there are no access points (APs) in this mode. An AP will deny any association and will cause a failed authentication when the wireless client is explicitly configured to use ad hoc mode.

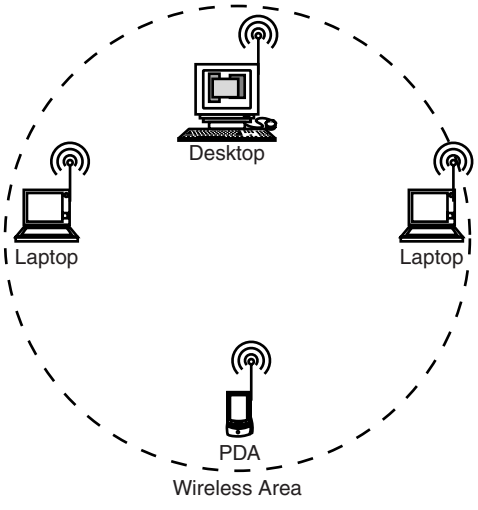


EXHIBIT 29.1 Ad hoc mode wireless LAN.

During implementation, WLAN bridge products are based on the infrastructure mode for PC-to-AP (network) connection.

As shown in Exhibit 29.2, the infrastructure mode consists of several clients talking to one or more APs that act as a distribution point. The AP will then act as a permanent structure and provide connectivity between the client and the wired network. Because an AP handles the connectivity control, the infrastructure mode offers several security protections, which are discussed further below.

As previously described, the 802.11 standard uses an unlicensed Industrial-Scientific-Medical (ISM) 2.4-GHz band, which is divided into 15 channels. (In some countries, legislation may limit the use of all available channels. For example, it might allow only the first 11 channels.) Wireless clients automatically scan all the channels to identify any listening channel by finding any available Access Points. If the parameter settings are matched, the connectivity will be established and users may use the network resource.

To differentiate one network from another, the 802.11 standard defines the Service Set Identifier (SSID). SSID makes all components under the same network use the same identifier and form a single network. Consequently, the components from different networks will not be able to talk to each other. This is similar to assigning a subnet mask for a particular network group. An AP will take only a transmitted frame with the same SSID and will disregard the others. An SSID can consist of up to 32 characters.

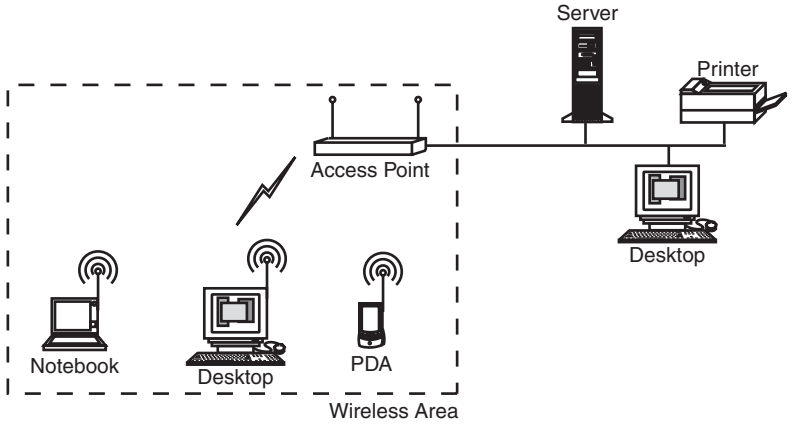


EXHIBIT 29.2 Infrastructure mode wireless LAN.

The 802.11 standard network uses a special transmission method called Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA). This media access sharing method is similar to the CSMA/CD method used by the 802.3 standard. The CSMA/CA method will listen to airwaves for any activity. If there is no activity detected, it will send the frame to airwaves. If the sender detects a collision, it will wait for a random time and then resend the frame. According to the recent and wide implementation of 802.11b, the bandwidth used by the system is up to 11 Mb per access point. Regarding the CSMA/CA sharing method, the real bandwidth used is divided among all users on that frequency. One can add another access point in the same area using different frequency channels (a maximum of three channels) to increase the network bandwidth.

Association Process

A process called an "association process" is needed to connect a network device to an AP. During this process, each device will authenticate to each other, similar to the handshake process in other protocols. The step-by-step process, shown in Exhibit 29.3, is as follows:

- *Unauthenticated and unassociated.* The client searches and selects a network name, called the SSID (Service Set Identifier).
- *Authenticated and unassociated.* The client does authentication with the access point.
- *Authenticated and associated.* The client sends an association request frame to the access point and the access point replies to the request.

Exhibit 29.3 shows this process.

There are two optional mechanisms during the authentication process: open authentication and shared key authentication. In open authentication, the client must know the SSID value and the WEP keys, if WEP is activated. The process will begin without any previous handshake and will use the SSID and WEP key value in the frame. In shared key authentication, wireless clients must first associate before they can use the access point to connect to a network. The association process starts with the client sending an association request to an Access Point. The access point will then reply with a challenge (some random cleartext) to the client. The

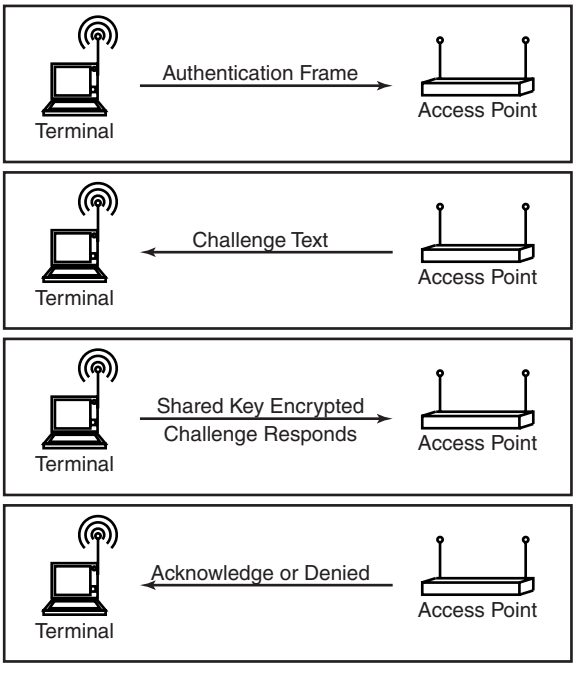


EXHIBIT 29.3 Association process.

client will have to encrypt the challenge with its WEP key and send back the response to the access point. The access point then decrypts the response and compares the result with the challenge. If they are matched, then both are authenticated. However, this authentication process is vulnerable to a known plaintext attack.

WLAN Security

In 1997, when the 802.11 standard was ratified, the authors were aware that this system needed privacy protection. That is why this standard is equipped with a security and privacy solution to make it equal to its traditional solution, which is the wired network. That is also where the name for the privacy solution "Wired Equivalent Privacy" originated. The idea was not to provide the most robust security solution, but only to provide an equivalent level of privacy to that offered by the wired network and thereby prevent standard eavesdropping.

WEP uses a 64-bit RC4 encryption algorithm, which consists of a 40-bit key and a 24-bit initialization vector (IV). The two available methods to use WEP keys are to use four shared different keys between stations and the access point or to use a key mapping table where each MAC will have a dedicated key.

Many papers have proven that there is an inadequate security mechanism offered by WEP keys. It is quite easy to attack the WEP and it is difficult to manage the keys. Changing the hard-coded keys in the station configuration frequently will not be suitable in a large WLAN deployment. Stolen devices and malicious users are just two examples of how the secret keys can be leaked out.

How WEP Works

Let's look at the step-by-step process of WEP to get more insight into how WEP actually works. Initially, the message will go through an integrity check process to ensure that the message is not changed due to the encryption process or a transmission error. The 802.11 standard uses CRC-32 to produce an integrity check value (ICV). The ICV will then be added to the end of the original message, and this combination will be encrypted at once. The next step is to create the key stream; in this case, WEP will use RC4 as its stream cipher encryption. The key stream generated by RC4 uses a combination of a random 24-bit initialization vector (IV), which is then added into the 40-bit secret key (declared in the authentication process). Both the 64-bit IV and secret key will then become the input for the RC4 algorithm and produce a key stream called a WEP pseudo-random number generator (PRNG). The WEP PRNG length is the same as the message plus the ICV. Once the stream cipher is working, the message and the ICV are XORed to produce a ciphertext. This ciphertext, together with the IV and key ID, are then ready to be transmitted. The key ID is an eight-bit value, consisting of six bits with a static value of zero and two bits for the actual key ID value. The key ID is used to figure out which one of the four secret keys (previously entered into both the access point and the client) is used to encrypt the frame. Now we can see that WEP only uses 40 bits of the secret key effectively; on the other hand, it uses a 64-bit input to generate the key stream. It is the 24-bit IV at the beginning of the key that has created a cryptographic flaw, as it is transmitted in plaintext and in the small IV space. Exhibit 29.4 shows this process.

IV Length Problem

The first standard for WEP, as defined in the 802.11 standard, is to use a 24-bit IV. This can lead to attacks due to the short length of the IV. A 24-bit length will produce approximately 16 million possible IVs. For an 11-Mbps wireless network, available IVs are used up in a few hours and will force the system to reuse previous IV values. It will then be up to the vendors to choose which IV selection method to use, because it is not defined yet within the standard. Some vendors use an incremental value starting from 00:00:00 during the device initialization and then incrementing by 1 until it reaches FF:FF:FF. This is similar to the TCP sequence number incrementation method from UNIX legacy. This IV collision problem can lead to cryptographic flaws, such as key stream reuse and the known plaintext attack.

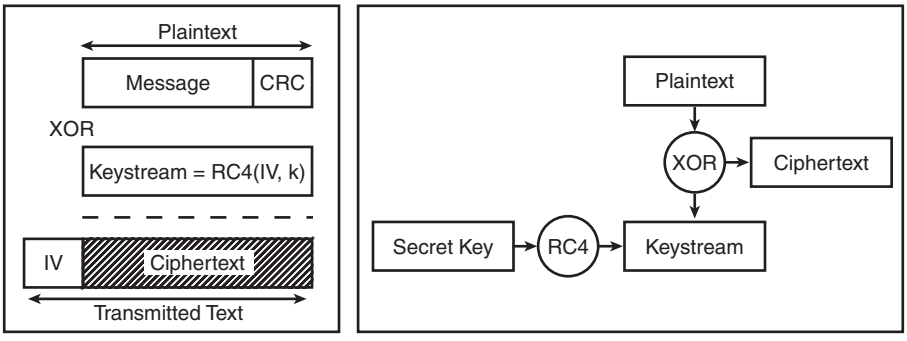


EXHIBIT 29.4 WEP encryption.

Wired Equivalent Protocol Version 2

Realizing the many problems within the standard, the IEEE then proposed an improvement for WEP security. WEP 2 uses a 128-bit key with the same RC4 algorithm and provides for mandatory Kerberos support. Despite the increased key length, it still uses the same IV length, which results in a 104-bit shared key and a 24-bit IV. Because the IV bit length is still the same, the entire problem related to the short IV length, such as known plaintext attacks and key stream reuse, will still be relevant. Furthermore, denial-of-service attacks and rogue access point problems are not yet solved in this new version of WEP. Hence, WEP 2 does not really solve the cryptographic flaw in the previous WEP.

RC4 Cryptographic Flaw

Further insight into the RC4 algorithm has revealed several problems associated with the RC4 stream cipher algorithm, as mentioned by the Cryptography Newsgroup in 1995. In 2001, Fluhrer, Mantin, and Shamir described the weaknesses of the key scheduling algorithm in RC4 — that is, the invariance weakness and the IV weakness. Invariance is the presence of numerous weak keys, where a small number of the keys are used to generate a major portion of the bits of the key scheduling algorithm (KSA) output. The second weakness — the IV weakness — is related to the common technique used to prevent a stream cipher from using the same key for all encryption sessions by using a different variable. This variable is commonly called the initialization vector, which is combined with the secret key to be used as input for RC4 algorithm and produce a PRNG. When the same IV is used with a number of different key stream values, the secret key can be extracted by analyzing the initial word of the key stream. Shamir et al. once demonstrated how to conduct a ciphertext attack to break an RC4 algorithm in WEP. This vulnerability also applies to the enhancement of WEP in WEP version 2. The Fluhrer, Mantin, and Shamir analysis was also proved by Adam Stubblefield of Rice University and John Ioannidis and Aviel Rubin of AT&T Labs in August 2001. In their research and with the permission of their administrator, Stubblefield and Ioannidis were able to crack WEP and pull out the secret key within a few hours. Although Stubblefield did not put the source code in his paper, there are several tools available on the Net to do it, such as Airsnort and WEPCrack. This software automates the process of secret key gathering and allows people without any knowledge of cryptography to attack WEP.

Some Attacks on WLAN

Keystream Reuse

One important thing to obtain to crack WEP-encrypted packets is the key stream, which can be extracted by XORing the ciphertext with the plaintext. This key stream can then be used to decrypt the WEP-encrypted packets as long as it is the one associated with the index value used during that particular communication session. There are two possible methods to obtain both the plaintext and ciphertext, along with its associated index value. The first method involves assuming that an attacker is able to send stimulus plaintext packets through the victim access points and is able to get the associated ciphertext by capturing the communication

traffic between the victim access points. When the associated ciphertext can be obtained, the particular index value used during this particular WEP-encrypted session will also be obtained because the index value information is available within the frame header. This information, the index value and the key stream, is then kept in a reference library. This process is then performed many times until the library contains all the possible index values along with the associated key stream. Once the attacker has this complete library, any WEP-encrypted packets passing through the victim access points can then be decrypted by XORing the ciphertext with a particular key stream obtained from the library and based upon the particular index value used during that particular session. The other method involves obtaining both the plaintext and ciphertext, along with the particular index value, sent during the initial association process. To have the complete library containing the key stream with its associated index value, this initial association process needs to occur many times. Such a circumstance can be set by sending a disassociation process to one of the points so that the already-established WEP-encrypted communication will be disconnected and another initial association process will need to occur.

Session Hijacking

Even after the client successfully performs the authentication and association processes, an attacker may still be able to hijack the client session by monitoring the airwaves for the client frame. By spoofing the access point information, an attacker can send a disassociation frame to the client, which will cause the client's session to be disconnected. Then, the attacker can establish a legitimate connection with the access point on behalf of the client and continue accessing the network resources. This session hijacking can occur in a system with no WEP activated. Unfortunately, the 802.11 standard does not provide any session-checking mechanism, and hence it creates the possibility for an attacker to hijack the session. The access point also does not know whether or not the original wireless client is still connected or whether or not the remote client is fake.

Man-in-the-Middle Attack

Basically, this type of attack is similar to a session hijacking attack, especially at the beginning of the process. Initially, the attacker will need to listen and monitor the airwaves. After adequate information is successfully gathered, the attacker will send a disassociated frame to the victim client. The client will send broadcast probes and try to re-associate itself. The attacker will answer the request using fake access-point software to answer the re-associate request. In the next phase, the attacker will try to establish an association with the real access point by spoofing the client's MAC address. If the real access point accepts the association, then the attacker can intercept and alter the information exchanged between the victim client and the access point. This type of attack may still occur although a MAC address-filtering scheme has been applied, as it is not very difficult to spoof a MAC address. This problem arises because the 802.11 standard only describes one-way authentication.

Denial-of-Service Attack

Because the airwaves are used as the transmission medium, the WLAN and its versions are very likely to be vulnerable to a denial-of-service (DoS) attack. The goal of a DoS attack is to make a remote system unavailable so that legitimate clients will not be able to use the victim computing resource. A high noise attack that uses a radio jamming technique by sending a strong transmission power on a transmission band can disturb the radio frequency propagation. All airwave-based connections on that particular frequency will then be broken. This disturbance can also accidentally happen, such as interference by other products like phones, WLAN cameras, etc. A similar attack is the traffic injection attack, where the WLAN is using the CSMA/CA mechanism and the attack uses the same radio channel as the target network. The target network will then accommodate the new traffic. This particular threat is getting worse because the attacker can send a broadcast disassociation frame in a very short period of time.

Common WLAN Security Problems

A Service Set Identifier (SSID) is an identifier used by WLANs to differentiate one network from another. The SSID provides the first level of security that differentiates corporate networks from others. That is why an SSID value should be managed carefully. It should not be predictable or incorporate any known word, but should

use letter-type combinations and other best practices for password creation. Usually, an access point is initially configured with a default SSID value such as "tsunami" for Cisco Aironet AP, "3com" or "101" for 3Com, and "linksys" for Linksys AP. Most engineers realize this but are too lazy to change it.

Another problem arises because many network personnel think that a stronger signal is better. Their objective is that the client must be able to receive a good signal level in as many places as possible. Such thought will introduce a higher exposure because attackers will be able to capture the traffic from the road or the parking area. Signal coverage should become an important point of consideration when implementing a wireless LAN. Several Internet sites even reveal how to make a strong signal interceptor from a Pringles® can and some PVC.

Connecting a WLAN access point into an internal network requires careful consideration because any failure can cause the entire network to be compromised. Improper implementation might also let an attacker bypass security defense systems, such as a firewall or an intrusion detection system (IDS). During product evaluation and testing, the WLAN device is attached directly to the internal network with its default configuration to see its life performance. Most engineers do not realize that by doing this, their corporate network may be compromised through this unsecured device during evaluation.

WEP, as the security feature currently available, is not really widely used. A survey conducted by Worldwide Wireless Wardrive reveals that many organizations install WLANs without using any security protection. Most implementations do not even use simple encryption. Another reason why WEP technology is not incorporated in most WLAN implementations is the connectivity mindset that believes that as long as the link connection is working properly, then the engineers' job is done. They do not pay much attention to the security aspect. Some engineers even refuse to configure WEP because they do not want to face additional difficulties.

Another reason is key management. WEP has a bad reputation because some WEP-supported products require entering the WEP key in hexadecimal while some other products accept alphanumeric characters. The inconsistency and difficulties of entering keys in a hexadecimal product are getting worse because WEP keys need to be changed periodically. WEP keys are stored in the access point and laptop. This leads to a chance that other users accessing this laptop may figure out the WEP configuration keys. Hence, the key protection mechanism is vulnerable, especially if the laptop is stolen. Every accident happening to the keys will require the keys to be renewed; and for preventive reasons, the key can be periodically refreshed. Imagine the problem with the current WEP if the administrator has to change the keys for hundreds of users. The final reason to drop WEP is that when WEP is enabled the throughput will decrease up to 50 percent.

Some problems exist when the ad hoc mode is used and the clients act as a bridge to the wired network. An attacker can try to enter the network by passing all the firewall and VPN protection. It is a problem similar to the split-tunnel in a VPN client. Therefore, it is not recommended to use ad hoc mode together with 802.3 Ethernet within a single device.

Countermeasures for WEP Limitation

The IEEE, the author of the 802.11 series, has accepted the standard protocol for WLAN by developing a task group to fix the security problem in the current protocol. The task group is working on the security protocol assigned the name 802.11i, which is expected to be finalized in early 2004. Meanwhile, vendors offer their own solutions to securing the 802.11 implementations. Organizations have to know the existing solution today and choose one that fits their needs in order to have a secure implementation of WLAN.

One solution is to provide an additional security protocol at the network layer, which is IP Security (IPSec). A mature security protocol like IPSec can overcome the weaknesses of WEP and should be jointly implemented to provide another layer of defense. However, the implementation of IPSec is a little more complex because each client will have to install an IPSec client in order to connect to the IPSec gateway. This gateway should be placed between the access point and the wired network. Operating systems that are already equipped with the IPSec feature will offer more advantages, as the process will be more transparent and use a single credential with the system logon. Examples of such operating systems include Windows 2000 and Windows XP. For a bridging solution, the implementation is easier because it will only consist of a pair of WLAN connected sites where the IPSec implementation will occur just after the WLAN bridge.

Some vendors have adopted the Extensible Authentication Protocol (EAP) defined in IEEE 802.1x, which is also called Robust Security Network (RSN). EAP uses a challenge–response scheme. An access point can open a port access only if the use has been authenticated. The access point will pass the challenge–response process between the client and the RADIUS server. The authentication process is done on the network layer

instead of the data-link layer. Several vendors are adopting this solution as an acting solution until the 802.11i standard is finalized. The EAP access points, by default, provide backward compatibility for clients that do not support RSN. This can lead to a new problem because, despite the recognition of RSN as a better security mechanism, the backward compatibility feature can still bypass it. The other limitation is the absence of mutual authentication between client and authenticator (AP), which mistakenly assumes that every access point can always be trusted. Other solutions are emerging in security equipment made by companies specializing in WLAN security, such as BlueSocket, Cranite Systems, Fortress Technologies, ReefEdge, and Vernier Networks. Some of them even offer appliances that can be installed between a WLAN network and a wired network. Examples include the solutions from BlueSocket, SMC, and Vernier Networks. Others offer software-based security solutions, such as NetMotion, ReefEdge, and Cranite Systems. Most of these systems provide an identification mechanism for users who need to get access into their organization resources by providing an authentication server or passing it to another authentication server like RADIUS.

Despite the weaknesses and risks associated with WEP, it is still possible to deploy a secure WLAN implementation by implementing several additional security configurations. The ease of cracking WEP-encrypted traffic is getting worse with the emergence of several tools that can automatically crack it. Hence, a WLAN must be considered an untrusted network. Non-built-in security features may be used in addition to securing the network with firewalls and IDSs.

Design Architecture and Implementation Guidelines

It is assumed that most security officers (hopefully this includes system and network administrators) understand the value of a security policy, yet many do not show much interest in starting work on it. As previously discussed, WLANs offer plenty of vulnerabilities and risks. Although an organization may not have a WLAN yet, it would be a good practice to have a policy on it. This is equivalent to the company information monitoring policy although the company may not yet really conduct information surveillance.

Including the WLAN implementation within a company security policy may bring concerns about WLAN insecurity into discussions within the security awareness program, management, network personnel, users, etc. The paradigm that a stronger signal is better will have to be put aside. Organizations should have limited the RF propagation if they want to have a secure WLAN implementation. They need to choose the right antenna and proper implementation design in order to get the most benefit from security. There are several types of antennas available on the market, such as the Yagi antenna, patch antenna, parabolic antenna, omni antenna, etc. Each type has its own characteristics. In a very sensitive organization such as the military or government, specially designed walls can be used to control signals coming in and out of a building. This requirement can be achieved with a Faraday cage theorem such as the one used in TEMPEST technology. Exhibit 29.5 shows the antenna implementation option.

Design and antenna considerations are just a small part of a set of defense-in-depth components, and hence the security efforts of a WLAN implementation should not be limited to these two components only. Some of these antennas do not require high-technology manufacturing or a high-cost product. It has been proven that an antenna can be made from an old Pringles can with a cost that is not more than U.S.$10.

As previously described, each WLAN device will have a unique identifier called an SSID. In operation, an access point will usually broadcast its SSID every few seconds; these are called beacon frames. The goal is to offer an easy and transparent process for use and quicken the association process for the user. Some NICs (network interface cards) can scan the airwaves and check the SSIDs available. Using a supported NIC and a supported operating system (e.g., Windows 2000 and Windows XP), a user could instantly join access to network resources while in the RF range. The problem is that this feature allows unauthorized users easy access without knowing the SSID for that network. Another problem is that this process speeds the recognition process for a bad guy to gather wireless network information, because the access point publishes its availability. For security reasons, it is highly recommended to turn off the beacon broadcast on every access point. Again, do not forget to change default SSIDs to some strong identifier.

Network design also has an important role in WLAN security implementation. According to the risk described earlier, separating the access point from other local networks is a must. Security practitioners or administrators could use a VLAN to separate the WLAN from the local network. A more robust solution is to put all WLAN networks on a dedicated interface to a firewall and treat them with scrutiny rules that check where all WLAN users can go and what services they can access. Even WLAN users can be treated as external

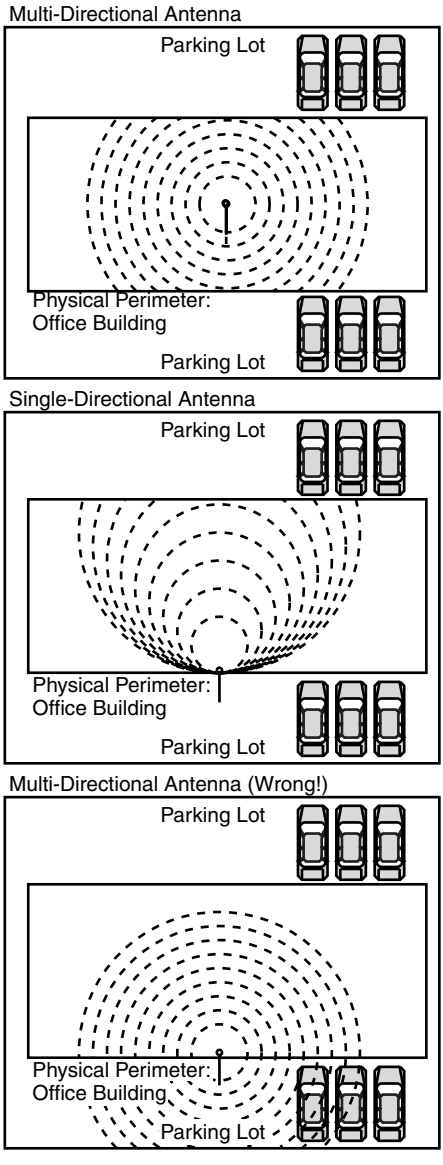


EXHIBIT 29.5 Antenna implementation option.

users. Using an intrusion detection system (IDS) on a WLAN segment can be a good idea to prevent any unauthorized access.

While WEP is the layer of defense available today and is proven to be not secure, it is still necessary to use built-in security features for minimum protection. Other security features could be added to provide more levels of security, including MAC filtering, although MAC addresses can be spoofed. The reason we must use that security feature is because there is no need to give a bad guy an easy way to attack the network. Security officers must decide what security baseline or what level of security is needed in their organizations based on the organization security policy (see Exhibit 29.6).

EXHIBIT 29.6 Wireless Security Policy Checklist

- ☐ Change the default SSID.
- ☐ Turn off SSID broadcasting.
- ☐ Enable WEP with a well-chosen key (flawed WEP is better than no WEP at all).
- ☐ Change WEP keys regularly.
- ☐ Use MAC address filtering.
- ☐ Locate all APs on dedicated port of firewall.
- ☐ Use a VPN to encrypt and authenticate all WLAN traffic.
- ☐ Use higher layer authentication and encryption (i.e., IPSec, SSH, etc.).
- ☐ Do a "signal audit" to determine where your wireless can be intercepted.
- ☐ Use an authentication mechanism, if possible (RADIUS, NoCatAuth, 802.1x, LEAP/PEAP, TTLS).
- ☐ Buy hardware with newer WEP replacements (TKIP, AES).
- ☐ Use anti-virus and personal firewalls on the client.
- ☐ Ensure client integrity before it connects to information resources

Auditing the Network

Most people believe they do not need to think about WLAN security problems because they do not use WLANs. It is completely wrong to think this way. Auditing the network to find an unauthorized access point is very important, even if one is not using the WLAN. Anyone (e.g., cleaning service, visitor, maintenance technician, employee, etc.) can easily attach an access point to an active network port that lets someone from outside the building attack the system. It is similar to having a network hub at the bus stop, but even worse.

Implementation and Change Control

It is virtually impossible to use the initial design throughout the entire life span of an application system. Business is dynamic, so systems that support the business should also be ready to change. A change control policy and procedure for WLAN and its related systems will ensure that systems remain secure after changes. It is important to audit to ensure that everyone follows the policy and procedure.

Operation

PC and network technicians can accidentally change the WLAN configuration during the troubleshooting period. New or temporary access points to replace a broken access point could have a different configuration (e.g., enabling SNMP) that effectively changes the security level of the system. Users or PC technicians could accidentally or intentionally change the configuration by enabling the ad hoc mode. Human error is always a potential security problem. Ensuring WLAN client integrity is another challenge.

Monitoring

There are several tools currently available on the market to monitor and audit your system, including freeware such as NetStumbler, Kismet, Airosniff, Ethereal, Aerosol, AirTraf, and Prism2Dump. Some examples of commercial tools include Airopeek, Sniffer Wireless, and Grasshopper. To audit an organization's perimeter and network, all that is needed is a notebook, a supported WLAN NIC, and a selected program. With selected programs, a security practitioner can start to map the organization's perimeter, looking for WLAN activity. With an installed program on a notebook, the security practitioner could walk to each room and each corner to check for a hidden or rogue access point. Some programs, such as Kismet and NetStumbler, can react with a sound every time a new network is discovered. With GPS support on the audit software and a GPS receiver, the security practitioner can map all discovered data and write a policy based on the findings. The security practitioner should check the exact location of the wireless perimeters. The audit process should be done regularly and randomly. If necessary, some organizations have left a dedicated device to monitor WLAN activity in their physical perimeter.

The New Security Standard

Wi-Fi Protected Access

Wi-Fi Protected Access (WPA) is a subset of the IEEE 802.11i draft standard and is designed to be forward-compatible with 802.11i when it is launched. WPA was announced by the Wi-Fi Alliance stating that it is not a standard, but instead a "specification for a standards-based, interoperable security enhancement."* Several members of the Wi-Fi Alliance teamed up with members of the IEEE 802.11i task group to develop WPA. WPA attempts to answer some problems in the present state of WLAN security by providing key management and robust user authentication.

To address the WEP key management problem, WPA chose the Temporal Key Integrity Protocol (TKIP). TKIP uses a master key that produces an encryption key from a mathematical computation. TKIP changes the encryption key regularly and uses the key only once. The entire process is to be done automatically in the system device. Something interesting to know is that the throughput delay time using TKIP is still unknown, and will have to wait until implementation of the protocol in real products at a later date.

The other major part that WPA addresses is the user authentication system. To provide easy and robust authentication, WPA uses the 802.1x standard and the Extensible Authentication Protocol (EAP) as its authentication technology. WPA supports two authentication modes: enterprise level authentication and SOHO (small office/home office) or consumer-level authentication. In the enterprise implementation, WPA requires another authentication server, usually RADIUS, as the user repository and authentication server to authenticate users before they can joint the network. For the SOHO authentication level, WPA uses single keys or passwords called pre-shared keys (PSKs) that have to be entered into both the access point and the client device. The password entered into both is used by TKIP to automatically generate encryption keys. WPA in SOHO mode standardizes the PSK to use an alphanumeric password instead of a hexadecimal in some WEP implementations.

The good thing about WPA is that the solution could be applied without having to purchase new hardware, because WPA still uses the same hardware and the same RC4 encryption method. All system upgrades to WPA can be done using software and firmware upgrades or some patching.

IEEE 802.11i

To address security problems in the current WLAN standard, the IEEE developed a new robust solution that will be 802.11i. This standard will address most of the WEP vulnerability issues and become a superset of the WPA solution from the Wi-Fi Alliance. The enhancements adopted by the WPA security solution excluded some specific features in the 802.11i draft, including secure IBSS, secure fast handoff, secure de-authentication and disassociation, and enhanced encryption protocols such as AES-CCMP.

To see products with 802.11i security, the professional will have to be patient because the first product that absorbs this standard is predicted to be launched in the beginning of 2005, or, at the very earliest, by the end of 2004. This long delay is because the standard has not been released yet and is only predicted to be released in 2004. The product needs some hardware upgrade and redesign because it uses different technology, such as an encryption engine change from RC4 to AES.

IEEE 802.1x Standard

IEEE 802.1x is a port-based network access control that uses an authenticating and authorizing devices mechanism to attach to a LAN and to prevent access to that port in cases in which the authentication and authorization process fails. IEEE 802.1x provides mutual authentication between clients and access points via an authentication server. Supporting WLAN security, 802.1x provides a method for dynamic key distribution to WLAN devices and solves the key reuse problem in the current standard. Vendors used this standard as part of their proprietary WLAN security solution to enhance the current 802.11b security standard. Unfortunately, two University of Maryland researchers have recently noted serious flaws in client-side security for 802.1x.

*Wi-Fi Alliance, "Wi-Fi Protected Access," www.weca.net/OpenSection/pdf/Wi-Fi_Protected_Access-Overview.pdf. October 31, 2002.

Temporal Key Integrity Protocol

The Temporal Key Integrity Protocol (TKIP) is a solution that fixes the key reuse problem associated with WEP. The TKIP process begins with a 128-bit temporal key shared among clients and access points. To add a unique identifier on each site, TKIP combines the temporal key with the client's MAC address and then adds a relatively large 16-octet initialization vector to produce the key that will encrypt the data. This process makes every client and access point use a different key stream to encrypt the payload data. TKIP changes temporal keys every 10,000 packets to ensure the confidentiality of the encrypted payload. Because TKIP still uses the RC4 algorithm to encrypt the payload, it is possible for current WLAN devices to upgrade with a simple firmware upgrade. TKIP is one of the methods used in Wireless Protected Access (WPA).

Conclusion

Wireless LANs, by design, have many higher risks than the simple ones, such as being stolen or subjected to high-technology attacks, eavesdropping, and encryption break-in. Often, a machine that holds important company data is exposed in connecting it to a wireless device without any additional protection. This should never happen.

Wireless LANs must get the same if not more protection than other technology. Even the more robust standard has not been released as yet, so proprietary solutions should be used to fill the security gap when wireless implementation becomes a choice.

30

ISO/OSI and TCP/IP Network Model Characteristics

George G. McBride, CISSP

Introduction

The development and implementation of standards is a requirement for the widespread growth and adaptation of the Internet and all the protocols it uses. In the late 1970s, the International Standards Organization (ISO) initiated efforts to develop a network communications standard based on open systems architecture theories from which other networked systems could be designed. The move from stand-alone mainframe systems to a networked infrastructure was underway and standards had to be developed to allow systems from one company to effectively communicate with systems from another company using intermediary networking devices developed by yet another company.

OSI Reference Model Overview

By the early 1980s, the ISO had introduced the Open Systems Interconnection (OSI) Reference Model. The OSI model provides a framework for any vendor to develop protocol implementations facilitating communications with other systems also using the OSI Reference Model. The OSI model has seven layers that are sometimes referred to as levels. Those seven layers make up a system's "stack" and are listed in order in Exhibit 30.1.

Although each of the seven layers are explained in detail later in this chapter, it is worthwhile to provide a brief overview of each layer here. Although the application layer is considered the "highest" layer, or layer 7, it is often convenient to discuss the OSI model from the "lowest" layer, or layer 1.

1. *Physical layer:* the hardware that carries those electrical values through the network between hosts.
2. *Data-link layer:* resolves synchronization issues, formats data into frames, and is responsible for converting between bits and electrical values.
3. *Network layer:* provides routing and forwarding of data between hosts.
4. *Transport layer:* manages the end-to-end control, including error checking and flow control.
5. *Session layer:* initiates, controls, and terminates communications between communicating systems
6. *Presentation layer:* handles the formatting and syntax issues for the application layer.
7. *Application layer:* acts as an interface to applications requesting network services.

Each of the seven layers was designed with certain guiding principles. For example, layers were created when different levels of abstraction were required to process the data. Each layer is cohesive and performs well-

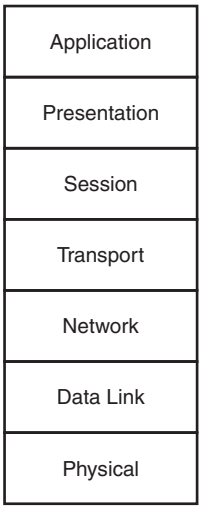


EXHIBIT 30.1 The OSI Reference Model stack.

defined and documented functions. Each layer is loosely coupled with its peer layers and minimizes the data flow between layers.

Layers communicate with adjacent layers strictly through interfaces. Lower layers provide services to upper layers through primitives that pass data and control information, and describe functions that need to be performed (i.e., "send this data to www.lucent.com" on port 80).

Data travels vertically within the OSI model. In general, each OSI layer adds layer-specific information to each message being sent as it travels downward in the OSI stack. That information, also called a header, is used to facilitate communications at the peer layer of other systems. As the remote system receives and processes the message, the header for that layer is processed and removed before passing the message up the stack. Additionally, a layer may find it necessary to fragment the data it receives from an adjacent layer. This data must be reassembled by the peer layer of the destination prior to moving the data up the stack.

Exhibit 30.2 shows a typical operation where the presentation layer has received data from the application layer. The data received at the presentation layer already has the application layer header added. The presentation layer does its processing, adds the presentation layer header, and then sends the data to the session layer where the process is repeated until the bottom of the OSI model is reached.

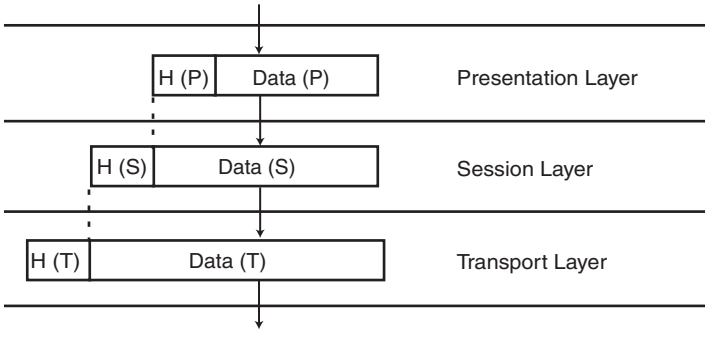


EXHIBIT 30.2 Addition of headers.

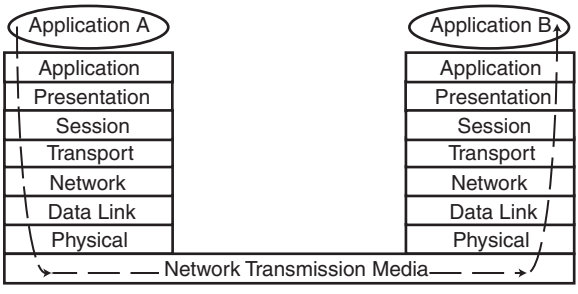


EXHIBIT 30.3 Data communication path.

Exhibit 30.3 shows the transmission of data from application A on a system with an OSI stack interacting with application B on a different system with an OSI stack. The data travels from application A down the stack, across the network transmission medium, and then back up the stack to application B.

It is important to note that the concept of layering is not without its disadvantages. For example, the standards do not specify how the data will pass between layers, leaving that task to the network stack implementers. Additionally, one of the disadvantages of data hiding is that it may lead to inefficient solutions. Although designers may be aware of techniques to process the data with fewer instructions or require less overhead, due to the concept of data hiding (restricting access to particular parameters, variables, etc.), the designers might be restricted in taking advantage of that information.

Finally, intermediate layers are required to retrieve data simultaneously from adjacent layers, process that data, and then forward the data to alternate layers. In some instances, the processing would be more effectively combined with the processing at other levels. Even worse, the actions of a particular layer may be nullified by the required processing at another layer.

When a system communicates with another computer system, the data is transferred between each of the systems at the physical layer, a layer that typically does not append any headers to the message.

One of the most important steps in understanding the OSI model is the correct sequencing of layers. Several key mnemonics have been developed over the years, providing an easy way to remember the ordering. From the top layer to the bottom, **A**ll **P**eople **S**eem **T**o **N**eed **D**ata **P**rocessing, and from the bottom to the top, **P**lease **D**o **N**ot **T**hrow **S**ausage **P**izza **A**way. Choose the one that you are most comfortable with and commit the order to memory.

Physical Layer Concepts

Overview

Responsible for the transmission of the raw bit stream, the physical layer does not define the media used, but defines the physical interface between devices and how the data is passed from one interface to another. The physical layer delivers the bits to the recipient as efficiently as possible. If bits are lost, changed in transit, or delivered out of sequence, the physical layer relies on the data-link layer to correct those errors.

By not specifying whether the transmission can occur over media such as coaxial, twisted pair, or satellite, the OSI model can be implemented in a number of different ways. This is one of the most important benefits of the OSI model. The OSI model specifies what must be performed at each layer, not how. New technologies, systems, and processes do not require modified stacks to be introduced to the other clients when the stacks of its communicating partners change.

The physical layer specifies four transmission characteristics:

1. *Electrical:* specifies the voltage levels of bit values 1 and 0, the time that each signal must be held, and the time between each bit value that is transmitted.

2. *Functional:* specifies the functions that will be performed, such as data, control, and timing issues.
3. *Mechanical:* specifies physical connection information such as the size of connectors and receptacles of the network hardware.
4. *Procedural:* specifies the sequence of events required to initiate, control, and tear down connections.

Examples

It often helps to visualize a physical device or application that would communicate at each of the levels. A repeater is a device operating at the physical layer that is used to extend communications links beyond the physical transmission limitations of connected network devices. For example, to extend a CAT5 100BaseT network cable beyond the recommended maximum length of 100 meters, a repeater should be used. A repeater is an inexpensive in-line device that takes an input signal on one interface and outputs the amplified signal on another interface. A repeater solely amplifies and relays the data from one interface to another; and because it does not care about the sequence or values of bits, errors in the signal will also be retransmitted.

Data-Link Layer Concepts

Overview

The primary purpose of the data-link layer is to convert between "frames" of data from upper layers and "bits" of data from the physical layer, and vice versa. In addition to the frame/bit conversion, the data-link layer provides addressing as well as reliability through error and flow control.

Although the data-link layer cannot correct errors, errors can be detected using checksums contained in the header. Typically implemented through a cyclical redundancy check (CRC), the receiver can request that the data be resent if its computed CRC value does not equal the received CRC value computed by the sending host.

At this layer of the OSI model, data is transferred in frames. To determine where one frame ends and the next frame begins, the data-link layer can utilize several different methods to separate frames, including:

- *Character count.* Each frame indicates in its header how many characters are in the frame.
- *Byte stuffing.* Also referred to as character stuffing, it uses an ASCII character to terminate a frame with some predefined end-of-frame character.
- *Bit stuffing.* Each frame begins and ends with a predefined bit pattern, such as "01111110."

It is worth noting that both bit and byte stuffing methods introduce the potential condition that the predefined delimiter is legitimately contained in the text and must be transmitted. For example, consider a message transfer using bit stuffing with the delimiter "01111110." When the sending data-link layer encounters five consecutive "1" bits, a "0" bit is inserted into the data stream, increasing the size of the data stream. When the receiving data-link layer encounters five consecutive "1" bits, the "0" bit is removed prior to processing the stream any further.

Additionally, the data-link layer has several methods to control the flow of data between hosts, including:

- *Stop and Wait.* This method sends one frame at a time and waits for a response. A positive acknowledgment indicates that the next frame can be transmitted, and a negative acknowledgment or timeout indicates that the frame must be resent.
- *Sliding Window.* This method sends up to number of predetermined frames prior to receiving an acknowledgment. The receiver acknowledges which packets have been received, effectively "narrowing" the window that indicates that additional packets can be sent. If no acknowledgment or an error is received from the receiver, the sender retransmits those particular packets.

There are two sub-layers that work together to make up the link layer:

1. *Logical Link Control (LLC).* This establishes and controls the links between communicating hosts utilizing the above-described error and flow control methods. LLC Type 1, or LLC1, provides connectionless data transfer. LLC Type 2, or LLC2, provides for a connection-oriented data transfer.

2. *Media Access Control (MAC).* Below the LLC, the MAC sub-layer provides a mechanism for multiple hosts to share the same media channel. This MAC sub-layer also provides a means to uniquely address each device, commonly called a MAC or hardware address. Protocols such as Ethernet, FDDI, and Token Ring use a MAC address that is generally preassigned by the equipment manufacturer, but is sometimes user changeable.

When packets are transmitted on a non-switched local area network (LAN), all hosts on the LAN can receive them. Unless the host is operating in promiscuous mode (such as when it is acting as a network sniffer, which forces the data-link layer to process all packets), the host reads only the frame far enough to determine the destination address. If the destination address is the host's own address or a broadcast address, the rest of the frame is read and processed up the stack. If the destination address is not the host's own address and is not a broadcast address, the remainder of the frame is discarded.

Examples

It is important to note a particularly important example of a data-link layer protocol, the High Level Data Link Control (HDLC) protocol. HDLC is a bit-oriented, bit-stuffed protocol with a frame structure that includes:

- *Pre-defined bit pattern:* pre-defined bit patterns mark the start and end of each frame.
- *Address:* identifies the destination through a hardware-based address.
- *Control:* used to provide sequencing, acknowledgments, negative acknowledgments, and other error messaging.
- *Data:* the data being transferred between hosts.
- *Checksum:* a variation of CRC is used to calculate and verify checksums.

HDLC, a successor to Synchronous Data Link Control (SDLC), was originally designed by IBM and is often used to provide data communications equipment (DCE) to data terminal equipment (DTE) connectivity between network equipment and data terminals.

A bridge is a physical device that connects a LAN to another LAN at each gateway point, both of which use the same protocol. Because the bridge must know the destination address, the data-link layer's MAC address must be obtained prior to the bridge deciding whether or not the frame must be forwarded to another segment or simply discarded if the destination address is on the same LAN segment as the sender. In some sense, a bridge can perform a function similar to the repeater as it may be required to retransmit packets from its interface to a more distant LAN segment.

Network Layer Concepts

Overview

The network layer adds functionality to the OSI Reference Model, to include the concept of network addresses that can be used to communicate with devices on logically separated communications networks, thus forming an internet. The network layer is responsible for establishing, maintaining, controlling, and terminating connections between interconnected hosts.

The network layer introduces the concept of a logical network address such as an Internet Protocol (IP) address, a 32-bit, decimally represented number indicating the source or destination address. In addition, a logical socket or port is introduced that specifies the target or destination process for the communications traffic.

When a data packet travels from one network to a different network, multiple issues can be introduced that must be resolved. The type of addressing might be different, the size of the packet might be too large, or the destination might be unreachable.

As part of maintaining and controlling the connection, the network layer introduces error control and congestion/flow control intended to prevent flooding of the LAN. The error and congestion controls can be implemented by either:

- *Connection mode.* Error and congestion controls are provided throughout the route of the connection path. Either the transmitting host, receiving host, or any of the intermediary network devices can issue flow control commands to the endpoint hosts.

- *Connectionless mode.* Error and congestion controls are provided only at the endpoints (sending and receiving hosts) of the connection path.

The OSI Reference Model accommodates several types of routing algorithms used to transmit traffic between endpoints. These algorithms include:

- *Circuit switching.* Similar to a traditional telephone circuit, a constant and dedicated path is established and maintained for the duration of the data connection.
- *Message switching.* This algorithm establishes and utilizes a dedicated path for each message transferred. Commonly called a "store-and-forward" network, a message is completely received and stored by an intermediary device prior to forwarding to the next destination. Subsequent messages, including those between identical sending and receiving devices, can travel independently along separate paths.
- *Packet switching.* In a packet-switched network, messages are transmitted in packets and those packets can travel through different intermediary devices prior to reaching their destinations. As some packets may be received out of order, the network layer is responsible for reordering and reconstructing the message before passing it up the stack.

Examples

The Internet Protocol (IP) is a protocol that operates at the network layer. The IP is a connectionless protocol that is responsible for routing the traffic between hosts and the addressing of the hosts.

A router is a device that operates within the network layer and determines which packets should be delivered to which networks that it knows about. Located at gateways, where interconnected networks are joined, a router makes decisions based on its routing table and current network conditions using the network address it has extracted from the data packet.

Transport Layer Concepts

Overview

The transport layer interacts with the network layer and provides supplemental functionality for establishing and tearing down connection services. The transport layer provides a true end-to-end connection between devices through:

- *Error control.* When the transport layer does not receive packets, missing packets are requested. By computing checksums of received packets, the transport layer can also detect erroneous packets and request that they be resent.
- *Flow control.* End-to-end flow control, including acknowledgments of data received back to the sending system.
- *Packaging.* The transport layer is responsible for the fragmenting and reassembly of packets.
- *Quality of service (QoS).* It is the transport layer's responsibility to provide the QoS requested by the session layer, such as the maximum delay and priority of the packets.
- *Sequencing.* The transport layer is responsible for passing the data to the session layer in the same sequence it was transmitted.

Examples

In lieu of a device operating at this level, the Transmission Control Protocol (TCP) operates at this level. TCP, which uses the Internet Protocol of the network layer, is used to provide connection-oriented message delivery. TCP ensures that messages are properly fragmented and reassembled, and re-requests packets that do not arrive or arrive with errors.

Secure Shell, also referred to as Secure Socket Shell (SSH), is a protocol for secure remote log-in and other secure network services over insecure networks. SSH provides strong encryption, cryptographic-based host authentication, and data integrity protection at the transport layer, typically running on top of TCP.

The User Datagram Protocol (UDP) is a connectionless-oriented message delivery protocol typically used where speed and efficiency are preferred over complete data delivery. For uses such as streaming video and

voice-over-IP (VoIP), UDP is the preferred choice because it may be acceptable to lose a small percentage of packets while the others travel with less overhead and are processed more efficiently.

It is at the transport layer that TCP and UDP introduce the concept of a port. Because several different applications may be running on one system using a single network interface, TCP and UDP need to keep track of what data goes to which application. Assigning a port number to every connection as it is established does this. The port number need not be the same (and is often not the same) on the local and remote processes. When a TCP or UDP segment is received, the protocol knows which process to pass it to by looking at the port number in the packet header.

Session Layer Concepts

Overview

The session layer is responsible for establishing, maintaining, and terminating the dialogue between applications. Sessions can allow dialogue in any of three formats:

1. *Simplex.* Each session is established and provides for unidirectional data transfer within the session. For example, a doorbell sends a signal to the buzzer in the house, but receives no feedback because the signal travels only in one direction.
2. *Half duplex.* Each session provides for non-concurrent bi-directional data transfer. For example, recall how some people have conversations on radio transceivers. After a person finished their thought, they would add "Over" at the end to indicate they were finished and other people could talk.
3. *Full duplex.* Each session provides for concurrent bi-directional data transfer. A perfect example of a full-duplex conversation would be a telephone call that allows all participants to talk and listen to the others simultaneously.

Another service provided by the session layer is token operation management. Some protocols, such as IBM's Token Ring protocol, require network management to ensure that only one system attempts to inject data into an empty token at any given time. Additional token management issues such as token release (Give Token), request a token (Please Token), and synchronization are also managed by the session layer.

The session layer provides an essential mechanism to insert "fail-safes" or checkpoints into connection streams. These checkpoints can be used to resume communications in the event that the session was interrupted and will not require the data transmitted prior to the last checkpoint to be retransmitted.

Examples

While the previous examples of layers of the OSI Reference Model have included hardware or protocols such as TCP, the session layer is best described as an established connection between two devices. Protocols such as Domain Name Service (DNS) and Network File System (NFS) operate at the session layer.

Presentation Layer Concepts

Overview

The presentation layer is responsible for the conversion of implementation-specific data syntaxes from the application layer to the session layer. Although this layer is a formal layer of the OSI Reference Model, many applications today do not utilize the concepts of the presentation layer and communicate directly with the session layer.

The presentation layer is responsible for the translation of data into various character representations, such as American Standard Code for Information Interchange (ASCII) and Unicode Worldwide Character Standard, commonly referred to as Unicode and used extensively in Microsoft products. As part of that data translation, byte and bit order translations are also managed. For example, when transmitting a byte's worth of bits (eight bits in a byte), some computers consider the first bit to be the "most significant bit" (MSB), while other systems

consider the first bit to be the "least significant bit" (LSB). The presentation layer would manage the transmission of data to ensure that the bits are properly ordered and processed.

In a similar fashion, the presentation layer is responsible for ensuring the proper ordering of the bytes that are sent. Consider that Microsoft Windows systems running on Intel's 80 x 86 processors are littleendian (*least* significant byte stored at the lower memory address) and that Solaris running on Sun's SPARC processors are bigendian (*most* significant byte stored at the lower memory address). When transmitting IPv4 addresses, which require 4 bytes in the TCP header, the data is transmitted in "network byte order," which is bigendian, or most significant byte first. If the packet being processed by the presentation layer is littleendian, the IP address would need to be converted.

Examples

Abstract System Notation.1 (ASN.1) is a formal method for describing the messages to be sent across a network. ASN.1 is comprised of two separate components, each of which is an ISO standard. One component of ASN.1 specifies the syntax for describing the contents of each message and the other component specifies how the data items are encoded in each message. Because ASN.1 does not specify content, the notation provides an excellent method to encode data at the presentation layer. If the data format changes or new formats are required, ASN.1 can easily adapt to include those changes and insulate the rest of the network stack from those changes.

Application Layer Concepts

Overview

The application layer provides an interface for which applications and end users can utilize networked resources. This layer is not an application itself and does not provide services to any upper layers; rather, it provides the networked resources to applications and end users.

The application entity (AE) is the part of the application that is considered to reside within the OSI model. Application service elements (ASEs) provide an abstract interface layer to the lower layers for the AE. Because the ASE provides such varied services, it is divided into common application service elements (CASEs) and specific application service elements (SASEs).

The CASEs provide services to more than one application. An example of a CASE is the association control service element (ACSE), which each application must contain. Other examples include general elements such as the reliable transfer service element (RTSE), remote procedure calls (RPCs), and distributed transaction processing (DTP).

The SASEs provide services to specific applications. Consider the International Telecommunications Union (ITU) x.400 set of standards, which specifies a messaging standard that is an alternative to SMTP-based e-mail. SASEs such as message retrieval service elements (MRSEs) and message transfer service elements (MTSEs) provide specific elements applicable to x.400.

The application layer protocol defines:

- *Types of messages:* request for data, response messages, etc.
- *Syntax of messages:* specifies the required fields and data formats
- *Semantics of fields:* defines the required and optional fields
- *Processing rules:* defines how messages will be sent and how responses will be processed

Application program interfaces (APIs) are also part of the application layer. APIs provide interfaces or "hooks" into the underlying network or computing infrastructure to allow programs to access network and computer resources without requiring extensive system- and operating-specific details. For example, instead of requiring a programmer to understand how numerous systems implement sockets, a network API allows a programmer to create a listening socket with one command and some parameters.

Examples

Consider a favorite Telnet client application that you can use to connect to another machine on your LAN. The Telnet application uses the Telnet protocol, which sits at the application layer. Additionally, the File Transfer

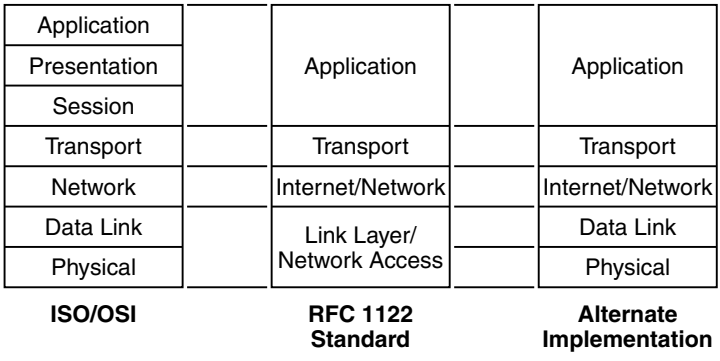


EXHIBIT 30.4 Comparison of OSI/ISO model, the RFC 1122 Standard model, and the five-layer Alternate model.

Protocol (FTP) uses the Telnet protocol to provide control communications between the FTP client and server. Finally, e-mail clients are not part of the application layer but may use the Simple Mail Transfer Protocol (SMTP), which is part of the application layer.

A Brief Introduction to the TCP/IP Protocol

While the concepts and the framework of the OSI Reference Model were being discussed in the late 1970s and early 1980s, the Defense Advanced Research Project Agency (DARPA) had already begun to define the TCP/IP protocols and architecture. In 1980, DARPA (since then, the "Defense" description has dropped and it is now referred to as ARPA) began to migrate machines connected to its research network to networks running the new TCP/IP protocol. Further solidifying the TCP/IP standardization was the U.S. Government's adoption of TCP/IP for all of its networks.

Unlike the OSI Reference Model, which originated through standards committees, the TCP/IP protocols developed through the efforts of the engineers to develop and implement the ARPANET (ARPA Network). Publicly available U.S. military standards were initially used to standardize the ARPANET and have since moved to Request For Comments (RFCs) as the ARPANET migrated to the Internet as we know it today.

According to RFC 1122, now part of the IETF Standards Track, the TCP/IP Model has four distinct layers in its stack:

1. Application Layer
2. Transport Layer
3. Internet Layer or Network Layer
4. Link Layer or Network Access Layer

It is worth noting that due to the rapid and widespread adaptation of the TCP/IP Protocol, several implementations contain a fifth layer in their design. In this implementation, the TCP/IP Link Layer contains a Data Link Layer and a Physical Layer. Figure 30.4 compares the OSI/ISO model, the RFC 1122 Standard model, and the five-layer alternate model.

The TCP/IP Link Layer works at the hardware level to define how the bits are physically transmitted across the network. The data is encapsulated into frames or packets and specifies how the data should be sent and received and includes provisions for encryption, quality of service, and flow and error control.

When considering the alternate model, the TCP/IP Physical Layer defines the physical characteristics of the medium used for communications. In addition, the Data Link Layer specifies details such as framing to manage how packets are transported over the physical layer.

The Internet (IP) layer provides the basic packet delivery service by encapsulating the data into packets of data called datagrams. Responsible for the routing of data, the Internet layer is a connectionless protocol and is solely responsible for the encapsulation and delivery of datagrams, which allows the data to traverse multiple networks through gateways.

The transport layer can utilize the TCP protocol, which initiates a three-way handshake between systems to establish a connection-based, reliable delivery service. Once the handshake has been established, data transfer

proceeds with the appropriate synchronization (SYN), acknowledgment (ACK), reset (RST), and other packets used to control the connection.

Alternatively, the transport layer may utilize the User Datagram Protocol (UDP). Using UDP, data is sent without establishing a connection between communicating hosts. While this method forces the application layer to provide any required sequencing, error detection, and error correction, efficiency is increased as UDP traffic generates less control overhead than TCP.

Whether the transport layer utilizes TCP or UDP, this layer specifies how data is to be communicated between different hosts.

Finally, the TCP/IP application layer provides to the system, application, or end user the interfaces to utilize requested networked resources. In some implementations, this layer may also provide services such as authentication and encryption.

Conclusion

The Internet has grown from a handful of machines sharing a small text file listing of connected hosts to a vast and global network of millions of machines across hundreds of countries. Years of work by some of the best scholars and engineers have been condensed to several pages in this chapter.

RFCs from the Internet Engineering Task Force and other documents help define the Internet as we know it today, and the reader is urged to review some of the common RFCs that help make up the protocols and services that most of us use every day. In addition, those RFCs and other documents from organizations such as the Institute of Electrical and Electronic Engineers (IEEE) and the Association of Computing Machinery (ACM) continue to shape the Internet as we will know it in years to come.

Integrity and Security of ATM

Steve Blanding

ATM (ASYNCHRONOUS TRANSFER MODE) IS A RAPIDLY GROWING AND QUICKLY MATURING, WIDE AREA NETWORK TECHNOLOGY. Many vendors, public carriers, private corporations, and government agencies are delivering ATM services in their product offerings today. The popularity of ATM has been driven by several industry developments over the past decade, including:

- the growing interest in merging telecommunication and information technology (IT) networking services
- the increasing demand for World Wide Web services

ATM is now considered the wide area network transport protocol of choice for broadband communications because of its ability to handle much larger data volumes when compared to other transport technologies. The demand for increased bandwidth has emerged as a result of the explosive growth of the World Wide Web and the trend toward the convergence of information networking and telecommunications.

The purpose of this chapter is to describe the key integrity and security attributes of ATM. The design and architectural design of ATM provide a basis for its integrity. However, because of its power and flexibility, opportunities for poorly controlled implementation of ATM also exists. The unique characteristics of ATM must be used to design a cost-effective ATM broadband transport network to meet Quality of Service (QoS) requirements under both normal and congested network conditions. The business case for ATM is reviewed first, followed by an analysis of transport service, control, signaling, traffic management, and network restoration.

THE BUSINESS CASE FOR ATM: COMPUTERS AND NETWORKING

There are three possible sectors that might use ATM technology in the computer and networking industry: ATM for the desktop, ATM for LANs,

and ATM for WANs. In general, ATM is winning the biggest place as a wide area networking solution, but there are serious challenges from existing and emerging LAN switching products (e.g., Fast Ethernet and Gigabit Ethernet) in the LAN and desktop environments.

The PC Desktop

Because of its cost, ATM is not currently perceived as an attractive option for the desktop environment when compared with existing and emerging technologies. Cost is not the only factor to consider when evaluating the use of ATM for the desktop. For example, most desktop applications today do not include the real-time multimedia for which ATM may be particularly suited. This increases the challenge of how to effectively bring ATM to the desktop. To overcome this challenge, the potential cost savings from eliminating private branch exchanges (PBXs) must be offset by the cost of upgrading every desktop with a new ATM network interface card.

To be competitive, ATM must be more cost affordable than switched Ethernet, which is regarded as the current standard in the industry. The most attractive approach would involve a solution that allows ATM to run over existing Ethernet. This approach would ignore higher-layer Ethernet protocols, reusing only the existing physical media, such as cabling and the Ethernet adapter. By adopting this solution, the need for any hardware upgrades to the desktop would be eliminated, requiring that workstation software be upgraded to include ATM signaling protocol, QoS, and flow control functionality.

LANs and WANs

The use of ATM technology for LANs will not become a widespread reality until application requirements force the traffic demand consistently into the gigabit-per-second range. The integration of voice, data, and video into a physical LAN would require the use of an ATM-type solution to meet the desired performance requirements. Currently, switched Ethernet and Gigabit Ethernet LANs are cost-effective solutions used to support most high traffic-intensive, client/server-based LAN applications.

The growth of high-demand WAN applications has driven the need for ATM as the transport technology solution of choice for wide area networking applications. Existing WAN transport technologies, such as Fiber Distributed Data Interface (FDDI), cannot support new applications that demand a QoS greater than FDDI's capability to deliver. ATM is considered the transport technology of choice although it is more expensive than FDDI and other similar transport solutions.

The recent explosive growth of the World Wide Web has also placed increased demands on higher bandwidth, wide area networks. As the Internet

becomes a greater source of video- and multimedia-based applications, the requirement for a more robust underlying transport infrastructure such as ATM becomes increasingly imperative. The design features of ATM and its explicit rate flow control functionality provide a basis to meet the increasing demands of the Internet.

THE BUSINESS CASE FOR ATM: TELECOMMUNICATIONS

The emerging broadband services provide the greatest incentive for the use of ATM in the telecommunications industry. Those services that require megabit-per-second speed bandwidth to meet QoS requirements are referred to as broadband services. These services can be divided into three major classes:

1. enterprise information networking services such as LAN interconnection and LAN emulation
2. video and image distribution services, including video on demand, interactive TV, multimedia applications, cable television, and home shopping services
3. high-speed data services, including frame relay services, switched multimegabit data service, ATM cell relay services, gigabit data service, and circuit emulation services

These emerging services would initially be supported by broadband ATM networks through permanent virtual connections (PVCs), which do not require processing functions, call control, or signaling. Switched virtual connection (SVC) service capabilities could be added as signaling standards are developed during the evolution of the network.

CHARACTERISTICS AND COMPONENTS OF ATM

ATM transmits information through uniform cells in a connection-oriented manner through the use of high-speed, integrated multiplexing and switching technology. This section describes the new characteristics of ATM, as opposed to synchronous transfer mode (STM), which includes bandwidth on demand, separation between path assignment and capacity assignment, higher operations and maintenance bandwidth, and nonhierarchical path and multiplexing structure.

Where ATM has been adopted by the International Telecommunication Union as the core transport technology, both narrowband and emerging broadband services will be supported by a Broadband Integrated Service Digital Network (B-ISDN). The telecommunication network infrastructure will continue to utilize ATM capability as demand for capacity increases. Different virtual channels (VCs) or virtual paths (VPs) with different QoS requirements are used within the same physical network to carry ATM services, control, signaling, and operations and maintenance messages in

order to maximize savings in this B-ISDN environment. To accomplish this, the integrated ATM transport model contains one service intelligent layer and two-layered transport networks. A control transport network and a service transport network make up the two-layered transport network. These correspond, respectively, to the control plan and user plan, and are coordinated by plane management and layer management systems.

B-ISDN Transport Network

The B-ISDN signal protocol reference model consists of three layers: physical, ATM, and ATM adaptation layer (AAL). The ATM transport platform is formed by the physical and ATM layers. The physical layer uses SONET standards and the AAL layer is a service-dependent layer. The SONET layer provides protection switching capability to ATM cells (when needed) while carrying the cells in a high-speed and transparent manner. Public network carriers have deployed SONET around the world for the last decade because of its cost-effective network architecture. The ATM layer provides, as its major function, fast multiplexing and routing for data transfer based on the header information. Two sublayers — the virtual path (VP) and virtual channel (VC) — make up the ATM layer. The unidirectional communication capability for the transport of ATM cells is described as the VC. Two types of VC are available: (1) permanent VC, which identifies the end-to-end connection established through provisioning, and (2) switched VC, which identifies the end-to-end connection established via near-real-time call setup.

A set of different VCs having the same source and destination can be accommodated by a VP. While VCs can be managed by users with ATM terminals, VPs are managed by network systems. To illustrate, a leased circuit may be used to connect a customer to another customer location using a VP and also be connected via a switched service using another VP to a central office. Several VCs for WAN and video conferencing traffic may be accommodated by each VP.

Virtual channel identifiers (VCIs) and virtual path identifiers (VPIs) are used to identify VCs and VPs, respectively. VCIs and VPIs are assigned on a per-link basis in large networks. As a result, intermediate ATM switches on an end-to-end VP or VC must be used to provide translation of the VPI or VCI.

Digital signals are provided by a SONET physical link bit stream. Multiple digital paths, such as Synchronous Transport Signal 3c (STS-3c), STS-12c, or STS-48c, may be included in a bit stream. STM using a hierarchical TSI concept is the switching method used for SONET's STS paths. A nonhierarchical ATM switching concept is the switching method used for VPs and VCs. Network rerouting through physical network reconfiguration is

performed by STM, and network rerouting using logical network reconfiguration through update of the routing table is performed by ATM.

Physical Path versus Virtual Path

The different characteristics of the corresponding path structures for SONET's STS paths (STM) and ATM VPs/VCs (ATM) result in the use of completely different switching principles. A physical path structure is used for the STS path and a logical path structure is used for the VP/VC path. A hierarchical structure with a fixed capacity for each physical path is characteristic of the physical path concept of the SONET STM system.

To illustrate, VT1.5s, with a capacity of 1.728 Mbps each, are multiplexed to an STS-1 and then to STS-12, and STS-48 with other multiplexed streams for optical transport over fiber. As a result, for each hierarchy of signals, a SONET transport node may equip a variety of switching equipment. The VP transport system is physically nonhierarchical, with a multiplexing structure that provides for a simplified nodal system design. Its capacity can be varied in a range from zero (for protection) up to the line rate, or STS-*N*c, depending on the application.

Channel Format

ATM switching is performed on a cell-by-cell basis based on routing information in the cell header. This is in contrast to the time slot channel format used in STM networks. Channels in ATM networks consist of a set of fixed-size cells and are identified through the channel indicator in the cell header.

The major function of the ATM layer is to provide fast multiplexing and routing for data transfer. This is based on information included in the 5-byte header part of the ATM cell. The remainder of the cell consists of a 48-byte payload. Other information contained in the header is used to (1) establish priority for the cell, (2) indicate the type of information contained in the cell payload, (3) facilitate header error control and cell delineation functions, and (4) assist in controlling the flow of traffic at the user-network interface (UNI).

Within the ATM layer, facility bandwidth is allocated as needed because ATM cells are independently labeled and transmitted on demand. This allocation is performed without the fixed hierarchical channel rates required for STM networks. Both constant and variable bit-rate services are supported at a broad range of bit rates because ATM cells are sent either periodically or in bursts (randomly). Call control, bandwidth management, and processing capabilities are not required through the permanent or semipermanent connections at the VP layer. Permanent, semipermanent, and switched connections are supported at the VC layer; however, switched

connections do require the signaling system to support its establishment, tear-down, and capacity management.

Adaptation Layer

The function of adapting services onto the ATM layer protocol is performed by the ATM adaptation layer (AAL). The functional requirements of a service are linked by the AAL to the ATM transport, which is characterized as generic and service independent. AAL can be terminated in the network or used by customer premise equipment (CPE) having ATM capability, depending on the service.

There are four basic AAL service models or classes defined by the ATM Forum, a group created by four computer and communications companies in 1991 to supplement the work of the ANSI standards group. These classes — Class A, B, C, and D — are defined based on the distinctions of three parameters: delay, bit rates, and connection modes. Class A identifies connection-oriented services with constant bit rates (CBRs) such as voice service. Within this class, the timing of the bit rates at the source and receiver are related. Connected-oriented services with variable bit rates (VBRs), and related source and receiver timing, are represented by Class B. These services are characterized as real-time, such as VBR video. Class C defines bursty connection-oriented services with variable bit rates that do not require a timing relationship between the source and the receiver. Connection-oriented data services such as file transfer and X.25 are examples of Class C service. Connectionless services similar to Class C are defined as Class D service. Switched multimegabit data service is an example of Class D service.

Available bit rate (ABR) and unspecified bit rate (UBR) are potential new ATM service classes within the AAL. ABR provides variable data rates based on whatever is available through its use of the end-to-end flow control system and is primarily used in LAN and TCP/IP environments. UBR, on the other hand, does not require the specification of a required bit rate, and cells are transported by the network whenever the network bandwidth is available.

Three types of AAL are also identified, which are in current use. These are AAL Type 1, Type 3/4, and Type 5. CBR applications are carried by AAL Type 1, which has an available cell payload of 47 bytes for data. The transparent transport of a synchronous DS1 through the asynchronous ATM network is an example of an application carried by AAL Type 1. Error-free transmission of VBR information is designed to be carried by AAL Type 3/4, which has an available payload of 44 bytes. Connectionless SMDS applications are carried by this AAL type. AAL Type 5, with an available cell payload of 48 bytes for data, is designed for supporting VBR data transfer with minimal overhead. Frame Relay Service and user network signaling

messages are transported over ATM using AAL Type 5. Other types of AAL include a null AAL and proprietary AALs for special applications. Null AALs are used to provide the basic capabilities of ATM switching and transport directly.

Comparing STM and ATM

STM and ATM use widely different switching concepts and methods. The major difference is that the path structure for STM is physical and hierarchical, whereas the structure for ATM is logical and nonhierarchical, due to its corresponding path multiplexing structure. With STM, the path capacity hierarchy is much more limited than with ATM. A relatively complex control system is required for ATM because of increased flexibility of bandwidth on demand, bandwidth allocation, and transmission system efficiency over the STM method. Network rerouting with STM may be slower than with ATM because rerouting requires physical switch reconfiguration as STM physically switches the signals.

BROADBAND SIGNALING TRANSPORT NETWORKS

Future broadband signaling needs must be addressed with a new, switched broadband service solution. These requirements demand a signaling network infrastructure that is much faster, more flexible, and more scalable than the older Signaling System #7 (SS7) signaling network solution. These new broadband signaling requirements can best be met through the implementation of an ATM signaling transport infrastructure. This section introduces the role of ATM technology in broadband signaling and potential ATM signaling network architectures.

New signaling requirements must be addressed in the areas of network services, intelligent networks, mobility management, mobility services, broadband services, and multimedia services. Broadband signaling enhancements needed to meet these requirements include: (1) increased signaling link speeds and processing capabilities, (2) increased service functionality, such as version identification, mediation, billing, mobility management, quality-of-service, traffic descriptors, and message flow control, (3) separate call control from connection control, and (4) reduced operational costs for services and signaling.

The Role of ATM in Broadband Signaling

The ATM signaling network has more flexibility in establishing connections and allocating needed bandwidth on demand when compared to the older SS7 signaling network solution. ATM is better suited to accommodate signaling traffic growth and stringent delay requirements due to flexible connection and bandwidth management capabilities. The ATM network is attractive for supporting services with unpredictable or unexpected traffic

patterns because of its bandwidth-on-demand feature. The bandwidth allocation for each ATM signaling connection can be 173 cells per second, up to approximately 1.5 Mbps, depending on the service or application being supported. Applications such as new broadband multimedia and Personal Communication Service (PCS) can best be addressed by an ATM signaling solution.

ATM Signaling

The family of protocols used for call and connection setup is referred to as signaling. The set of protocols used for call and connection setup over ATM interfaces is called ATM signaling. The North American and international standards groups have specified two ATM signaling design philosophies. These architectures are designed for public networks and for enterprise networks, which is called Private Network-to-Network Interface or Private Network Node Interface (PNNI). The different natures of public and enterprise networks have resulted in the different signaling network design philosophies between public and enterprise networks. Network size, stability frequency, nodal complexity, and intelligent residence are the major differences between the public networks and enterprise networks. An interoffice network is generally on the order of up to several hundred nodes in public networks. As a result, a cautious, long planning process for node additions and deletions is required. In contrast, frequent node addition and deletion is expected in an enterprise network containing thousands, or tens of thousands, of nodes. Within the public network node, the network transport, control, and management capabilities are much more complex, reliable, and expensive than in the enterprise network. Thus, intelligent capabilities reside in customer premise equipment within enterprise networks, whereas intelligence in the public networks is designed primarily in the network nodes.

Enterprise ATM Signaling Approach. A TCP/IP-like structure and hierarchical routing philosophy form the foundation of enterprise ATM network routing and signaling as specified in the Private Network Node Interface (PNNI) by the ATM Forum. The PNNI protocol allows the ATM enterprise network to be scaled to a large network, contains signaling for SVCs, and includes dynamic routing capabilities. This hierarchical, link-state routing protocol performs two roles: (1) to distribute topology information between switches and clusters of switches used to compute routing paths from the source node through the network and (2) to use the signaling protocol to establish point-to-point and point-to-multi-point connections across the ATM network and to enable dynamic alternative rerouting in the event of a link failure.

The topology distribution function has the ability to automatically configure itself in networks where the address structure reflects the topology

using a hierarchical mechanism to ensure network scalability. A connection's requested bandwidth and QoS must be supported by the path, which is based on parameters such as available bit rate, cell loss ratio, cell transfer delay, and maximum cell rate. Because the service transport path is established by signaling path tracing, the routing path for signaling and the routing path for service data are the same under the PNNI routing protocol.

The dynamic alternative rerouting function allows for reestablishment of the connection over a different route without manual intervention if a connection goes down. This signaling protocol is based on user-network interface (UNI) signaling with additional features that support crankback, source routing, and alternate routing of call setup requests when there has been a connection setup failure.

Public ATM Signaling Approach. Public signaling has developed in two major areas: the evolution of the signaling user ports and the evolution of the signaling transport in the broadband environment. Broadband signaling transport architectures and protocols are used within the ATM environment to provide reliable signaling transport while also making efficient use of the ATM broadband capabilities in support of new, vastly expanded signaling capabilities. Benefits of using an ATM transport network to carry the signaling and control messages include simplification of existing signaling transport protocols, shorter control and signaling message delays, and reliability enhancement via the self-healing capability at the VP level. Possible broadband signaling transport architectures include retention of signal transfer points (STPs) and the adoption of a fully distributed signaling transport architecture supporting the associated signaling mode only.

ATM NETWORK TRAFFIC MANAGEMENT

The primary role of network traffic management (NTM) is to protect the network and the end system from congestion in order to achieve network performance objectives while promoting the efficient use of network resources. The power and flexibility of bandwidth management and connection establishment in the ATM network has made it attractive for supporting a variety of services with different QoS requirements under a single transport platform. However, these powerful advantages could become disadvantages in a high-speed ATM network when it becomes congested. Many variables must be managed within an ATM network — bandwidth, burstiness, delay time, and cell loss. In addition, many cells have various traffic characteristics or quality requirements that require calls to compete for the same network resources.

Functions and Objectives

The ATM network traffic management facility consists of two major components: proactive ATM network traffic control and reactive ATM network

congestion control. The set of actions taken by the network to avoid congested conditions is ATM network traffic control. The set of actions taken by the network to minimize intensity, spread, and duration of congestion, where these actions are triggered by congestion in one or more network elements, is ATM network congestion control. The objective is to make the ATM network operationally effective. To accomplish this objective, traffic carried on the ATM network must be managed and controlled effectively while taking advantage of ATM's unique characteristics with a minimum of problems for users and the network when the network is under stress. The control of ATM network traffic is fundamentally related to the ability of the network to provide appropriately differentiated QoS for network applications.

Three sets of NTM functions are needed to provide the required QoS to customers:

1. *NTM surveillance functions* are used to gather network usage and traffic performance data to detect overloads as indicated by measures of congestion (MOC).
2. *Measures of congestion (MOC)* are defined at the ATM level based on measures such as cell loss, buffer fill, utilization, and other criteria.
3. *NTM control functions* are used to regulate or reroute traffic flow to improve traffic performance during overloads and failures in the network.

Effective management of ATM network traffic must address how users define their particular traffic characteristics so that a network can recognize and use them to monitor traffic. Other key issues include how the network avoids congestion, how the network reacts to network congestion to minimize effects, and how the network measures traffic to determine if the cell can be accepted or if congestion control should be triggered. The most important issue to be addressed is how quality-of-services is defined at the ATM layer.

The complexity of ATM traffic management design is driven by unique characteristics of ATM networks. These include the high-speed transmission speeds, which limit the available time for message processing at immediate nodes and result in a large number of cells outstanding in the network. Also, the traffic characteristics of various B-ISDN services are not well-understood and the VBR source generates traffic at significantly different rates with very different QoS requirements.

The following sections describe ATM network traffic and congestion control functions. The objectives of these control functions are:

- to obtain the optimum set of ATM layer traffic controls and congestion controls to minimize network and end-system complexity while maximizing network utilization

- to support a set of ATM layer QoS classes sufficient for all planned B-ISDN services
- to not rely on AAL protocols that are B-ISDN service specific, nor on higher-layer protocols that are application specific

ATM Network Traffic Control

The set of actions taken by the network to avoid congested conditions is called network traffic control. This set of actions, performed proactively as network conditions dynamically change, includes connection admission control, usage and network parameter control, traffic shaping, feedback control, and network resource management.

Connection Admission Control. The set of actions taken by the network at the call setup phase to determine whether a virtual channel connection (VCC) or a virtual path connection (VPC) can be accepted is called connection admission control (CAC). Acceptance of a connection request is only made when sufficient resources are available to establish the connection through the entire network at its required QoS. The agreed QoS of existing connections must also be maintained. CAC also applies during a call renegotiation of the connection parameters of an existing call. The information derived from the traffic contract is used by the CAC to determine the traffic parameters needed by usage parameter control (UPC), routing and allocation of network resources, and whether the connection can be accepted.

Negotiation of the traffic characteristics of the ATM connections can be made using the network at its connection establishment phase. Renegotiation of these characteristics may be made during the lifetime of the connection at the request of the user.

Usage/Network Parameter Control. The set of actions taken by the network to monitor and control traffic is defined as usage parameter control (UPC) and network parameter control (NPC). These actions are performed at the user-network interface (UNI) and the network-network interface (NNI), respectively. UPC and NPC detect violations of negotiated parameters and take appropriate action to maintain the QoS of already established connections. These violations can be characterized as either intentional or unintentional acts.

The functions performed by UPC/NPC at the connection level include connection release. In addition, UPC/NPC functions can also be performed at the cell level. These functions include cell passing, cell rescheduling, cell tagging, and cell discarding. Cell rescheduling occurs when traffic shaping and UPC are combined. Cell tagging takes place when a violation is detected. Cell passing and cell rescheduling are performed on cells that are identified by UPC/NPC as conforming. If UPC identifies the cell as nonconforming to at

least one element of the traffic contract, then cell tagging and cell discarding are performed.

The UPC/NPC function uses algorithms to carry out its actions. The algorithms are designed to ensure that user traffic complies with the agreed parameters on a real-time basis. To accomplish this, the algorithms must have the capability of detecting any illegal traffic situation, must have selectivity over the range of checked parameters, must exhibit rapid response time to parameter violations, and must possess simplicity for implementation. The algorithm design must also consider the accuracy of the UPC/NPC. UPC/NPC should be capable of enforcing a PCR at least 1 percent larger than the PCR used for the cell conformance evaluation for peak cell rate control.

Traffic Shaping. The mechanism that alters the traffic characteristics of a stream of cells on a VCC or a VPC is called traffic shaping. This function occurs at the source ATM endpoint. Cell sequence integrity on an ATM connection must be maintained through traffic shaping. Burst length limiting and peak cell rate reduction are examples of traffic shaping. Traffic shaping can be used in conjunction with suitable UPC functions as an option. The acceptable QoS negotiated at call setup must be attained, however, with the additional delay caused by the traffic shaping function. Customer equipment or terminals can also use traffic shaping to ensure that the traffic generated by the source or at the UNI is conforming to the traffic contract.

For typical applications, cells are generated at the peak rate during the active period and not at all during the silent period. At the time of connection, the amount of bandwidth reserved is between the average rate and the peak rate. Cells must be buffered before they enter the network so that the departure rate is less than the peak arrival rate of cells. This is the purpose of traffic shaping. Delay-sensitive services or applications, such as signaling, would not be appropriate for the use of traffic shaping.

As indicated previously, traffic can be reshaped at the entrance of the network. At this point, resources would be allocated in order to respect both the CDV and the fixed nodal processing delay allocated to the network. Two other options for traffic shaping are also available. One option is to dimension the network to accommodate the input CDV and provide for traffic shaping at the output. The other option is to dimension the network both to accommodate the input CDV and comply with the output CDV without any traffic shaping.

Feedback Control. The set of actions taken by the network and by users to regulate the traffic submitted to ATM connections according to the state of network elements is known as feedback control. The coordination of available network resource and user traffic volume for the purpose of

avoiding network congestion is the responsibility of the feedback control mechanism.

Network Resource Management. Resource management is defined as the process of allocating network resources to separate traffic flows according to service characteristics. Network resource management is heavily dependent on the role of VPCs. One objective of using VPCs is to reduce the requirement of establishing individual VCCs by reserving capacity. By making simple connection admission decisions at nodes where VPCs are terminated, individual VPCs can be established. The trade-off between increased capacity costs and reduced control costs determines the strategies for reservation of capacity on VPCs. The performances of the consecutive VPCs used by a VCC and how it is handled in virtual channel connection-related functions determine the peer-to-peer network performance on a given VCC.

The basic control feature for implementation of advanced applications, such as ATM protection switching and bandwidth on demand, is VP bandwidth control. There are two major advantages of VP bandwidth control: (1) reduction of the required VP bandwidth, and (2) bandwidth granularity. The bandwidth of a VP can be precisely tailored to meet the demand with no restriction due to path hierarchy. Much higher utilization of the link capacity can be achieved when compared with digital, physical-path bandwidth control in STM networks.

ATM Network Congestion Control

The state of network elements and components (e.g., hubs, switches, routers, etc.) where the network cannot meet the negotiated network performance objectives for the established connections is called network congestion. The set of actions taken by the ATM network to minimize the intensity, spread, and duration of congestion is defined as ATM network congestion control. Network congestion does not include instances where buffer overflow causes cell losses but still meets the negotiated QoS.

Network congestion is caused by unpredictable statistical fluctuations of traffic flows under normal conditions or just simply having the network come under fault conditions. Both software faults and hardware failures can result in fault conditions. The unattended rerouting of network traffic, resulting in the exhaustion of some particular subset of network resources, is typically caused by software faults. Network restoration procedures are used to overcome or correct hardware failures. These procedures can include restoration or shifting of network resources from unaffected traffic areas or connections within an ATM network. Congestion measurement and congestion control mechanisms are the two major areas that make up the ATM network congestion control system.

Measure of Congestion. Performance parameters, such as percentage of cells discarded (cell loss ratio) or the percentage of ATM modules in the ATM NT that are congested, are used to define measures of congestion of an ATM network element (NE). ATM switching fabric, intraswitching links, and modules associated with interfaces are ATM modules within an ATM NE. ATM module measures of congestion include buffer occupancy, utilization, and cell loss ratio.

Buffer occupancy is defined as the number of cells in the buffer at a sampling time, divided by the cell capacity of the buffer. Utilization is defined as the number of cells actually transmitted during the sample interval, divided by the cell capacity of the module during the sampling interval. The cell loss ratio is defined as the number of cells dropped during the sampling interval, divided by the number of cells received during the sampling interval.

Congestion Control Functions. Recovery from network congestion occurs through the implementation of two processes. In the first method, low-priority cells are selectively discarded during the congestion. This method allows for the network to still meet network performance objectives for aggregate and high-priority flows. In the second method, an explicit forward congestion indication (EFCI) threshold is used to notify end users to lower their access rates. In other words, an EFCI is used as a congestion notification mechanism to assist the network in avoiding and recovering from a congested state.

Traffic control indication can also be performed by EFCI. When a network element begins to reach an impending state of congestion, an EFCI value may be set in the cell header for examination by the destination customer premise equipment (CPE). A state in which the network is operating around its maximum capacity level is defined as an impending congested state. Controls can be programmed into the CPE that would implement protocols to lower the cell rate of the connection during congestion or impending congestion.

Currently, three types of congestion control mechanisms can be used in ATM networks. These mechanisms include link-by-link credit-based congestion control, end-to-end rate-based congestion control, and priority control and selective cell discard. These congestion control methods can be used collectively within an ATM network; the most popular method is to use the priority control and selective discard method in conjunction with either the rate-based congestion control or credit-based congestion control.

The mechanism based on credits allocated to the node is called credit-based congestion control. This is performed on a link-by-link basis requiring that each virtual channel (VC) have a credit before a data cell can be sent. As a result, credits are consistently sent to the upstream node to maintain a continuous flow of data when cells are transmitted on a VC.

The other congestion control mechanism that utilizes an approach that is adaptive to network load conditions is called rate-based congestion control. This control mechanism adjusts the access rate based on end-to-end or segmented network status information. The ATM node notifies the traffic sources to adjust their rates based on feedback received from the network. The traffic source slows the rate at which it transmits data to the network upon receiving a congestion notification.

The simplest congestion control mechanism is the priority control and selective cell discard mechanism. Users can generate different priority traffic flows by using the cell loss priority (CLP), bit, allowing a congested network element to selectively discard cells with low priority. This mechanism allows for maximum protection of network performance for high-priority cells. For example, assume CLP=0 is assigned for low-priority flow, CLP=1 is assigned for high-priority flow, and CLP=0+1 is assigned for multiplexed flow. Network elements may selectively discard cells of the CLP=1 flow and still meet network performance objectives on both the CLP=0 and CLP=0+1 flow. The Cell Loss Ratio objective for CLP=0 cells should be greater than or equal to the CLR objective for the CLP=1 flow for any specified ATM connection.

ATM Network Restoration Controls

Network restoration is one of greatest area of control concerns in an ATM network. Loss of high-speed, high-capacity ATM broadband services due to disasters or catastrophic failures would be devastating to customers dependent on those services. While this area is one of most significant areas that must be addressed, providing protection against broadband network failures could be very expensive due to the high costs associated with transport equipment and the requirement for advanced control capability. An extremely important challenge in today's emerging ATM network environment is providing for an acceptable level of survivability while maintaining reasonable network operating costs. Growing technological advancements will have a major impact on the challenges of maintaining this critical balance.

Currently, there are three types of network protection and restoration schemes that can be utilized to minimize the effects of broadband ATM services when a network failure occurs. These control mechanisms include protection switching, rerouting, and self-healing. The term "network restoration" refers to the rerouting of new and existing connections around the failure area when a network failure occurs.

Protection Switching. The establishment of a preassigned replacement connection using equipment but without a network management control function is called protection switching. ATM protection switching systems can use one of two different design approaches: one based on fault

management and the other based on signaling capability. The design of the fault management system is independent of the routing design for the working system. The signaling capability design uses the existing routing capability to implement the protection switching function. This design can minimize development costs but may only be applicable to some particular networks using the same signaling messaging system.

Rerouting. The establishment of a replacement connection by the network management control connection is defined as rerouting. The replacement connection is routed depending on network resources available at the time the connection failure occurs. An example of rerouting is the centralized control DCS network restoration. Network protection mechanisms developed for automatic protection switching or for self-healing can also be used for network rerouting. As a result, network rerouting is generally considered as either centralized control automatic protection switching or as self-healing.

Self-healing. The establishment of a replacement connection by a network without utilizing a network management control function is called self-healing. In the self-healing technique, the replacement connection is found by the network elements (NE) and rerouted depending on network resources available at the time a connection failure occurs.

SUMMARY

This chapter has reviewed the major integrity and security areas associated with ATM transport network technology. These areas — transport service, control, and signaling, traffic management, and network restoration — form the foundation required for building and maintaining a well-controlled ATM network. The design and infrastructure of ATM must be able to support a large-scale, high-speed, high-capacity network while providing an appropriae multi-grade QoS requirement. The cost of ATM must also be balanced with performance and recoverability, which is a significant challenge to ATM network designers. Continuing technological changes and increasing demands for higher speeds and bandwidth will introduce new challenges for maintaining integrity and security of the ATM network environment.

31

Enclaves: The Enterprise as an Extranet

Bryan T. Koch, CISSP

Even in the most secure organizations, information security threats and vulnerabilities are increasing over time. Vulnerabilities are increasing with the complexity of internal infrastructures; complex structures have more single points of failure, and this in turn increases the risk of multiple simultaneous failures. Organizations are adopting new, untried, and partially tested products at ever-increasing rates. Vendors and internal developers alike are relearning the security lessons of the past — one at a time, painful lesson by painful lesson.

Given the rapid rate of change in organizations, minor or incremental improvements in security can be offset or undermined by "organizational entropy." The introduction of local area networks (LANs) and personal computers (PCs) years ago changed the security landscape, but many security organizations continued to function using centralized control models that have little relationship to the current organizational or technical infrastructures. The Internet has brought new threats to the traditional set of organizational security controls. The success of the Internet model has created a push for electronic commerce (E-commerce) and electronic business (E-business) initiatives involving both the Internet itself and the more widespread use of Internet Protocol (IP)-based extranets (private business-to-business networks).

Sophisticated, effective, and easy-to-use attack tools are widely available on the Internet. The Internet has implicitly linked competing organizations with each other, and linked these organizations to communities that are opposed to security controls of any kind. There is no reason to assume that attack tools developed in the Internet cannot or will not be used within an organization.

External threats are more easily perceived than internal threats, while surveys and studies continue to show that the majority of security problems are internal. With all of this as context, the need for a new security paradigm is clear.

The time has come to apply the lessons learned in Internet and extranet environments to one's own organization. This chapter proposes to apply Internet/extranet security architectural concepts to internal networks by creating protected *enclaves* within organizations. Access between enclaves and the enterprise is managed by *network guardians*. Within enclaves, the security objective is to apply traditional controls consistently and well. Outside of enclaves, current practice (i.e., security controls at variance with formal security policies) is tolerated (one has no choice). This restructuring can reduce some types of network security threats by orders of magnitude. Other threats remain and these must be addressed through traditional security analysis and controls, or accepted as part of normal risk/reward trade-offs.

Security Context

Security policies, procedures, and technologies are supposed to combine to yield acceptable risk levels for enterprise systems. However, the nature of security threats, and the probability that they can be successfully deployed against enterprise systems, have changed. This is partly a result of the diffusion of computer technology and computer networking into enterprises, and partly a result of the Internet.

For larger and older organizations, security policies were developed to address security vulnerabilities and threats in legacy mainframe environments. Legacy policies have been supplemented to address newer threats such as computer viruses, remote access, and e-mail. In this author's experience, it is rare for current policy frameworks to effectively address network-based threats. LANs and PCs were the first steps in what has become a marathon of increasing complexity and inter-relatedness; intranet (internal networks and applications based on IP), extranet, and Internet initiatives are the most common examples of this.

The Internet has brought network technology to millions. It is an enabling infrastructure for emerging E-business and E-commerce environments. It has a darker side, however, because it also:

- Serves as a "proving ground" for tools and procedures that test for and exploit security vulnerabilities in systems
- Serves as a distribution medium for these tools and procedures
- Links potential users of these tools with anonymously available repositories

Partly because it began as an "open" network, and partly due to the explosion of commercial use, the Internet has also been the proving ground for security architectures, tools, and procedures to protect information in the Internet's high-threat environment. Examples of the tools that have emerged from this environment include firewalls, virtual private networks, and layered physical architectures. These tools have been extended from the Internet into extranets.

In many sectors — most recently telecommunications, finance, and healthcare — organizations are growing primarily through mergers and acquisitions. Integration of many new organizations per year is challenging enough on its own. It is made more complicated by external network connectivity (dial-in for customers and employees, outbound Internet services, electronic commerce applications, and the like) within acquired organizations. It is further complicated by the need to integrate dissimilar infrastructure components (e-mail, calendaring, and scheduling; enterprise resource planning (ERP); and human resources (HR) tools). The easiest solution — to wait for the dust to settle and perform long-term planning — is simply not possible in today's "at the speed of business" climate.

An alternative solution, the one discussed here, is to accept the realities of the business and technical contexts, and to create a "network security master plan" based on the new realities of the internal threat environment. One must begin to treat enterprise networks as if they are an extranet or the Internet and secure them accordingly.

The One Big Network Paradigm

Network architects today are being tasked with the creation of an integrated network environment. One network architect described this as a mandate to "connect everything to everything else, with complete transparency." The author refers to this as the One Big Network paradigm. In this author's experience, some network architects aim to keep security at arm's length — "we build it, you secure it, and we don't have to talk to each other." This is untenable in the current security context of rapid growth from mergers and acquisitions.

One Big Network is a seductive vision to network designers, network users, and business executives alike. One Big Network will — in theory — allow new and better business interactions with suppliers, with business customers, and with end-consumers. Everyone connected to One Big Network can — in theory — reap great benefits at minimal infrastructure cost. Electronic business-to-business and electronic-commerce will be — in theory — ubiquitous.

However, one critical element has been left out of this brave new world: security. Despite more than a decade of networking and personal computers, many organizational security policies continue to target the legacy environment, not the network as a whole. These policies assume that it is possible to secure stand-alone "systems" or "applications" as if they have an existence independent of the rest of the enterprise. They assume that attackers will target applications rather than the network infrastructure that links the various parts of the distributed application together. Today's automated attack tools target the network as a whole to identify and attack weak applications and systems, and then use these systems for further attacks.

One Big Network changes another aspect of the enterprise risk/reward equation: it globalizes risks that had previously been local. In the past, a business unit could elect to enter into an outsource agreement for its

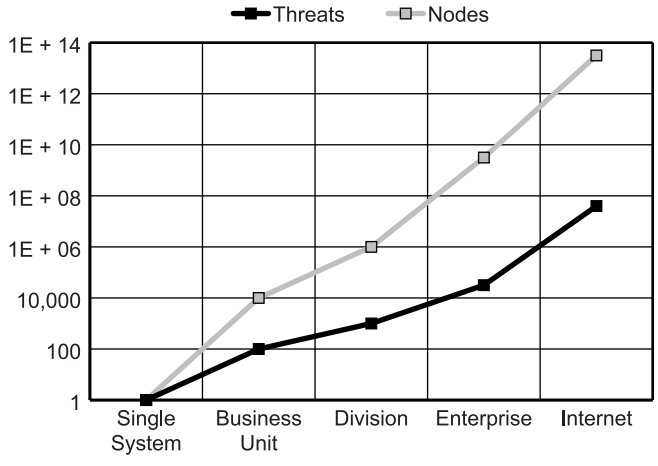


EXHIBIT 31.1 Network threats (log scale).

applications, secure in the knowledge that the risks related to the agreement affected it alone. With One Big Network, the risk paradigm changes. It is difficult, indeed inappropriate, for business unit management to make decisions about risk/reward trade-offs when the risks are global while the benefits are local.

Finally, One Big Network assumes consistent controls and the loyalty of employees and others who are given access. Study after study, and survey after survey, confirm that neither assumption is viable.

Network Security and the One Big Network Paradigm

It is possible that there was a time when One Big Network could be adequately secured. If it ever existed, that day is long past. Today's networks are dramatically bigger, much more diverse, run many more applications, connect more divergent organizations, all in a more hostile environment where the "bad guys" have better tools than ever before. The author believes that it is not possible to secure, to any reasonable level of confidence, any enterprise network for any large organization where the network is managed as a single "flat" network with "any-to-any" connectivity.

In an environment with no effective internal network security controls, each network node creates a threat against every other node. (In mathematical terms, where there are n network nodes, the number of threats is approximately n^2.) Where the organization is also on the Internet without a firewall, the effective number of threats becomes essentially infinite (see Exhibit 31.1).

Effective enterprise security architecture must augment its traditional, applications-based toolkit with *network-based tools* aimed at addressing network-based threats.

Internet Security Architecture Elements

How does one design differently for Internet and extranet than one did for enterprises? What are Internet/extranet security engineering principles?

- *Simplicity.* Complexity is the enemy of security. Complex systems have more components, more single points of failure, more points at which failures can cascade upon one another, and are more difficult to certify as "known good" (even when built from known good components, which is rare in and of itself).
- *Prioritization and valuation.* Internet security systems know what they aim to protect. The sensitivity and vulnerability of each element is understood, both on its own and in combination with other elements of the design.

- *Deny by default, allow by policy.* Internet security architectures begin with the premise that all traffic is to be denied. Only traffic that is explicitly required to perform the mission is enabled, and this through defined, documented, and analyzed pathways and mechanisms.
- *Defense in depth, layered protection.* Mistakes happen. New flaws are discovered. Flaws previously believed to be insignificant become important when exploits are published. The Internet security architecture must, to a reasonable degree of confidence, fail in ways that result in continued security of the overall system; the failure (or misconfiguration) of a single component should not result in security exposures for the entire site.
- *End-to-end, path-by-path analysis.* Internet security engineering looks at all components, both on the enterprise side and on the remote side of every transaction. Failure or compromise of any component can undermine the security of the entire system. Potential weak points must be understood and, if possible, managed. Residual risks must be understood, both by the enterprise and by its business partners and customers.
- *Encryption.* In all Internet models, and most extranet models, the security of the underlying network is not assumed. As a result, some mechanism — encryption — is needed to preserve the confidentiality of data sent between the remote users and enterprise servers.
- *Conscious choice, not organic growth.* Internet security architectures are formally created through software and security engineering activities; they do not "just happen."

The Enclave Approach

This chapter proposes to treat the enterprise as an extranet. The extranet model invokes an architecture that has security as its first objective. It means identifying what an enterprise genuinely cares about: what it lives or dies by. It identifies critical and securable components and isolates them into protected *enclaves*. Access between enclaves and the enterprise is managed by *network guardians*. Within enclaves, the security objective is to apply traditional controls consistently and well. Outside of enclaves, current practice (i.e., security controls at variance with formal security policies), while not encouraged, is acknowledged as reality. This restructuring can reduce some types of network security threats by orders of magnitude. Taken to the extreme, all business-unit-to-business-unit interactions pass through enclaves (see Exhibit 31.2).

Enclaves

The enclaves proposed here are designed to contain high-value securable elements. Securable elements are systems for which security controls consistent with organizational security objectives can be successfully designed, deployed, operated, and maintained at any desired level of confidence. By contrast, nonsecurable

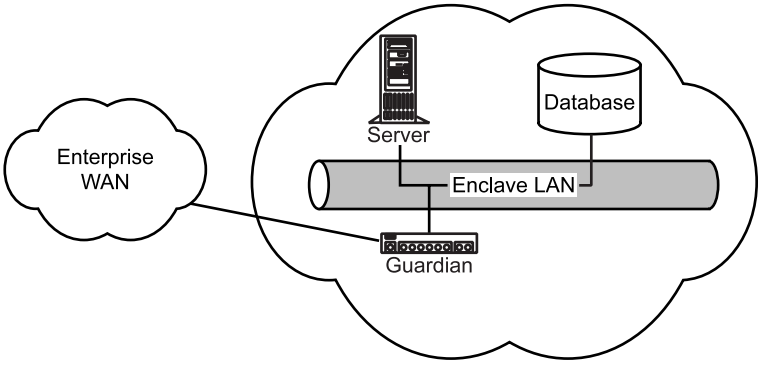


EXHIBIT 31.2 Relationship of an enclave to the enterprise.

elements might be semi-autonomous business units, new acquisitions, test labs, and desktops (as used by telecommuters, developers, and business partners) — elements for which the cost, time, or effort required to secure them exceeds their value to the enterprise.

Within a secure enclave, every system and network component will have security arrangements that comply with the enterprise security policy and industry standards of due care. At enclave boundaries, security assurance will be provided by network guardians whose rule sets and operational characteristics can be enforced and audited. In other words, there is some level of assurance that comes from being part of an enclave. This greatly simplifies the security requirements that are imposed on client/server architectures and their supporting applications programming interfaces (APIs). Between enclaves, security assurance will be provided by the application of cryptographic technology and protocols.

Enclave membership is earned, not inherited. Enclave networks may need to be created from the ground up, with existing systems shifted onto enclave networks when their security arrangements have been adequately examined.

Enclaves could potentially contain the elements listed below:

1. Mainframes
2. Application servers
3. Database servers
4. Network gateways
5. PKI certificate authority and registration authorities
6. Network infrastructure components (domain name and time servers)
7. Directories
8. Windows "domain controllers"
9. Approved intranet Web servers
10. Managed network components
11. Internet proxy servers

All these are shared and securable to a high degree of confidence.

Network Guardians

Network guardians mediate and control traffic flow into and out of enclaves. Network guardians can be implemented initially using network routers. The routers will isolate enclave local area network traffic from LANs used for other purposes (development systems, for example, and user desktops) within the same physical space. This restricts the ability of user desktops and other low-assurance systems to monitor traffic between remote enclave users and the enclave. (Users will still have the ability to intercept traffic on their own LAN segment, although the use of switching network hubs can reduce the opportunity for this exposure as well.)

The next step in the deployment of network guardians is the addition of access control lists (ACLs) to guardian routers. The purpose of the ACLs is similar to the functionality of "border routers" in Internet firewalls — screening incoming traffic for validity (anti-spoofing), screening the destination addresses of traffic within the enclave, and to the extent possible, restricting enclave services visible to the remainder of the enterprise to the set of intended services.

Decisions to implement higher levels of assurance for specific enclaves or specific enclave-to-enclave or enclave-to-user communications can be made based on later risk assessments. Today and for the near future, simple subnet isolation will suffice.

Enclave Benefits

Adopting an enclave approach reduces network-based security risks by orders of magnitude. The basic reason is that in the modern enterprise, the number of nodes (n) is very large, growing, and highly volatile. The number of enclaves (e) will be a small, stable number. With enclaves, overall risk is on the order of $n \times e$, compared with $n \times n$ without enclaves. For large n, $n \times e$ is much smaller than $n \times n$.

Business units can operate with greater degrees of autonomy than they might otherwise be allowed, because the only data they will be placing at risk is their own data on their own networks. Enclaves allow the realignment of risk with reward. This gives business units greater internal design freedom.

Because they require documentation and formalization of network data flows, the presence of enclaves can lead to improved network efficiency and scalability. Enclaves enforce an organization's existing security policies, at a network level, so by their nature they tend to reduce questionable, dubious, and erroneous network traffic and provide better accounting for allowed traffic flows. This aids capacity planning and disaster planning functions.

By formalizing relationships between protected systems and the remainder of the enterprise, enclaves can allow faster connections to business partners. (One of the significant sources of delay this author has seen in setting up extranets to potential business partners is collecting information about the exact nature of network traffic, required to configure network routers and firewalls. The same delay is often seen in setting up connectivity to newly acquired business units.)

Finally, enclaves allow for easier allocation of scarce security resources where they can do the most good. It is far easier to improve the security of enclave-based systems by, say, 50 percent than it is to improve the overall security of all desktop systems in the enterprise by a similar amount, given a fixed resource allocation.

Limitations of Enclaves

Enclaves protect only the systems in them; and by definition, they exclude the vast majority of the systems on the enterprise network and all external systems. Some other mechanism is needed to protect data in transit between low-assurance (desktops, external business partner) systems and the high-assurance systems within the enclaves. The solution is a set of confidentiality and authentication services provided by encryption. Providing an overall umbrella for encryption and authentication services is one role of public key infrastructures (PKIs).

From a practical perspective, management is difficult enough for externally focused network guardians (those protecting Internet and extranet connectivity). Products allowing support of an enterprisewide set of firewalls are just beginning to emerge. Recent publicity regarding Internet security events has increased executive awareness of security issues, without increasing the pool of trained network security professionals, so staffing for an enclave migration may be difficult.

Risks remain, and there are limitations. Many new applications are not "firewall friendly" (e.g., Java, CORBA, video, network management). Enclaves may not be compatible with legacy systems. Application security is just as important — perhaps more important than previously — because people connect to the application. Applications, therefore, should be designed securely. Misuse by authorized individuals is still possible in this paradigm, but the enclave system controls the path they use. Enclave architecture is aimed at network-based attacks, and it can be strengthened by integrating virtual private networks (VPNs) and switching network hubs.

Implementation of Enclaves

Enclaves represent a fundamental shift in enterprise network architecture. Stated differently, they re-apply the lessons of the Internet to the enterprise. Re-architecting cannot happen overnight. It cannot be done on a cookie-cutter, by-the-book basis. The author's often-stated belief is that "security architecture" is a verb; it describes a *process*, rather than a destination. How can an organization apply the enclave approach to its network security problems? In a word, planning. In a few more words, information gathering, planning, prototyping, deployment, and refinement. These stages are described more fully below.

Information Gathering

Information is the core of any enclave implementation project. The outcome of the information-gathering phase is essentially an inventory of critical systems with a reasonably good idea of the sensitivity and criticality of these systems. Some readers will be fortunate enough to work for organizations that already have information

systems inventories from the business continuity planning process, or from recent Year 2000 activities. A few will actually have accurate and complete information. The rest will have to continue on with their research activities.

The enterprise must identify candidate systems for enclave membership and the security objectives for candidates. A starting rule-of-thumb would be that no desktop systems, and no external systems, are candidates for enclave membership; all other systems are initially candidates. Systems containing business-critical, business-sensitive, legally protected, or highly visible information are candidates for enclave membership. Systems managed by demonstrably competent administration groups, to defined security standards, are candidates.

External connections and relationships, via dial-up, dedicated, or Internet paths, must be discovered, documented, and inventoried.

The existing enterprise network infrastructure is often poorly understood and even less well-documented. Part of the information-gathering process is to improve this situation and provide a firm foundation for realistic enclave planning.

Planning

The planning process begins with the selection of an enclave planning group. Suggested membership includes senior staff from the following organizations: information security (with an emphasis on network security and business continuity specialists), network engineering, firewall management, mainframe network operations, distributed systems or client/server operations, E-commerce planning, and any outsource partners from these organizations. Supplementing this group would be technically well-informed representatives from enterprise business units.

The planning group's next objective is to determine the scope of its activity, answering a set of questions including at least:

- Is one enclave sufficient, or is more than one a better fit with the organization?
- Where will the enclaves be located?
- Who will manage them?
- What level of protection is needed within each enclave?

What is the simplest representative sample of an enclave that could be created within the current organization?

The purpose of these questions is to apply standard engineering practices to the challenge of carving out a secure enclave from the broader enterprise, and to use the outcome of these practices to make a case to enterprise management for the deployment of enclaves.

Depending on organizational readiness, the planning phase can last as little as a month or as long as a year, involving anywhere from days to years of effort.

Prototyping

Enclaves are not new; they have been a feature of classified government environments since the beginning of computer technology (although typically within a single classification level or compartment). They are the basis of essentially all secure Internet electronic commerce work. However, the application of enclave architectures to network security needs of large organizations is, if not new, at least not widely discussed in the professional literature. Further, as seen in Internet and extranet environments generally, significant misunderstandings can often delay deployment efforts, and efforts to avoid these delays lead either to downward functionality adjustments, or acceptance of additional security risks, or both.

As a result, prudence dictates that any attempt to deploy enclaves within an enterprise be done in a stepwise fashion, compatible with the organization's current configuration and change control processes. The author recommends that organizations considering the deployment of the enclave architecture first evaluate this architecture in a prototype or laboratory environment. One option for doing this is an organizational test environment. Another option is the selection of a single business unit, district, or regional office.

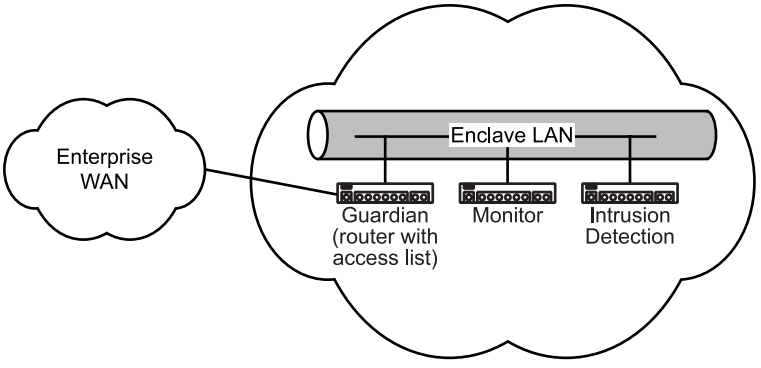


EXHIBIT 31.3 Initial enclave guardian configuration.

Along with the selection of a locale and systems under evaluation, the enterprise must develop evaluation criteria: what does the organization expect to learn from the prototype environment, and how can the organization capture and capitalize on learning experiences?

Deployment

After the successful completion of a prototype comes general deployment. The actual deployment architecture and schedule depends on factors too numerous to mention in any detail here. The list includes:

- *The number of enclaves.* (The author has worked in environments with as few as one and as many as a hundred potential enclaves.)
- *Organizational readiness.* Some parts of the enterprise will be more accepting of the enclave architecture than others. Early adopters exist in every enterprise, as do more conservative elements. The deployment plan should make use of early adopters and apply the lessons learned in these early deployments to sway or encourage the more change-resistant organizations.
- *Targets of opportunity.* The acquisition of new business units through mergers and acquisitions may well present targets of opportunity for early deployment of the enclave architecture.

Refinement

The enclave architecture is a concept and a process. Both will change over time: partly through organizational experience and partly through the changing technical and organizational infrastructure within which they are deployed.

One major opportunity for refinement is the composition and nature of the network guardians. Initially, this author expects network guardians to consist simply of already-existing network routers, supplemented with network monitoring or intrusion detection systems. The router will initially be configured with a minimal set of controls, perhaps just anti-spoofing filtering and as much source and destination filtering as can be reasonably considered. The network monitoring system will allow the implementers to quickly learn about "typical" traffic patterns, which can then be configured into the router. The intrusion detection system looks for known attack patterns and alerts network administrators when they are found (see Exhibit 31.3).

In a later refinement, the router may well be supplemented with a firewall, with configuration rules derived from the network monitoring results, constrained by emerging organizational policies regarding authorized traffic (see Exhibit 31.4).

Still later, where the organization has more than one enclave, encrypted tunnels might be established between enclaves, with selective encryption of traffic from other sources (desktops, for example, or selected business partners) into enclaves. This is illustrated in Exhibit 31.5.

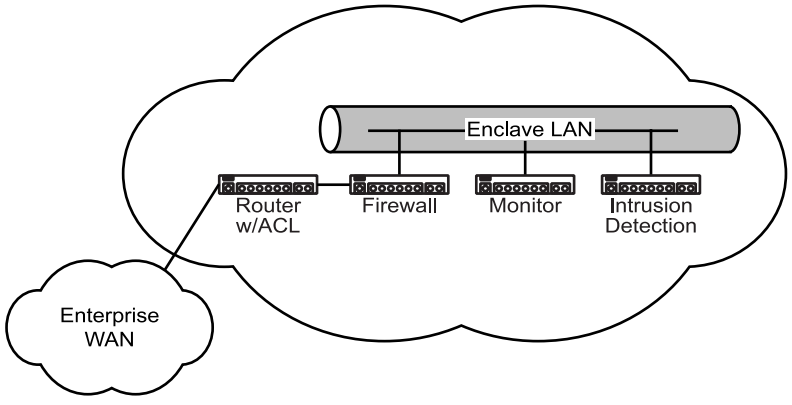


EXHIBIT 31.4 Enclave with firewall guardian.

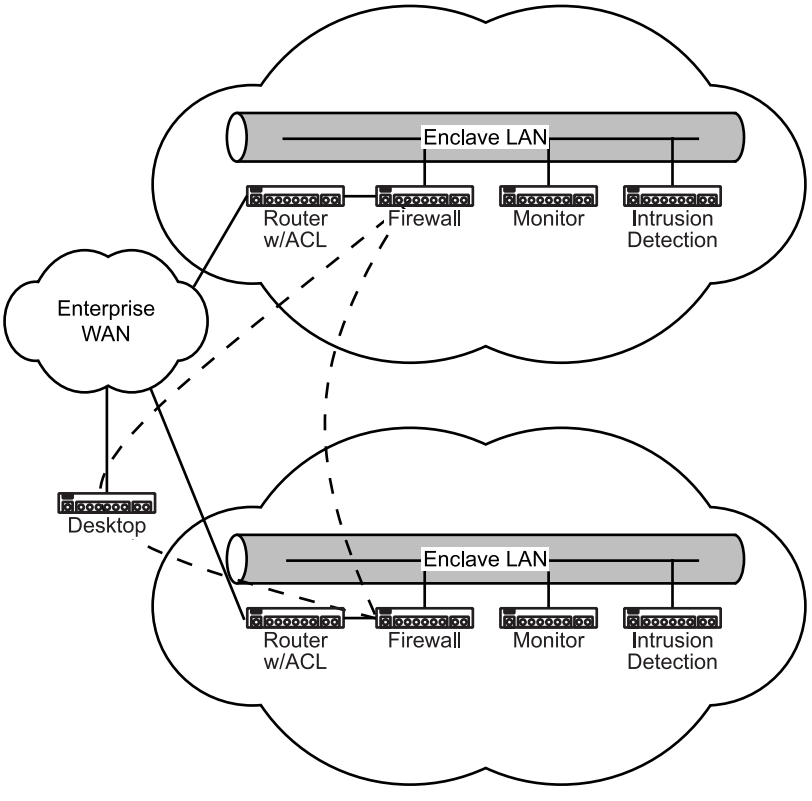


EXHIBIT 31.5 Enclaves with encrypted paths (dashed lines).

Conclusion

The enterprise-as-extranet methodology gives business units greater internal design freedom without a negative security impact on the rest of the corporation. It can allow greater network efficiency and better network disaster planning because it identifies critical elements and the pathways to them. It establishes security triage. The net results are global threat reduction by orders of magnitude and improved, effective real-world security.

32

IPSec Virtual Private Networks

James S. Tiller, CISA, CISSP

The Internet has graduated from simple sharing of e-mail to business-critical applications that involve incredible amounts of private information. The need to protect sensitive data over an untrusted medium has led to the creation of virtual private networks (VPNs). A VPN is the combination of tunneling, encryption, authentication, access control, and auditing technologies and services used to transport traffic over the Internet or any network that uses the TCP/IP protocol suite to communicate.

This chapter:

- Introduces the IPSec standard and the RFCs that make up VPN technology
- Introduces the protocols of the IPSec suite and key management
- Provides a technical explanation of the IPSec communication technology
- Discusses implementation considerations and current examples
- Discusses the future of IPSec VPNs and the industry's support for growth of the standard

History

In 1994, the Internet Architecture Board (IAB) issued a report on "Security in the Internet Architecture" (Request For Comment [RFC] 1636). The report stated the general consensus that the Internet needs more and better security due to the inherent security weaknesses in the TCP/IP protocol suite, and it identified key areas for security improvements. The IAB also mandated that the same security functions become an integral part of the next generation of the IP protocol, IPv6. So, from the beginning, this evolving standard will continually be compatible with future generations of IP and network communication technology.

VPN infancy started in 1995 with the AIAG (Automotive Industry Action Group), a nonprofit association of North American vehicle manufacturers and suppliers, and their creation of the ANX (Automotive Network eXchange) project. The project was spawned to fulfill a need for a TCP/IP network comprised of trading partners, certified service providers, and network exchange points. The requirement demanded efficient and secure electronic communications among subscribers, with only a single connection over unsecured channels. As this technology grew, it became recognized as a solution for any organization wishing to provide secure communications with partners, clients, or any remote network. However, the growth and acceptance had been stymied by the lack of standards and product support issues.

In today's market, VPN adoption has grown enormously as an alternative to private networks. Much of this has been due to many performance improvements and the enhancement of the set of standards. VPN connections must be possible between any two or more types of systems. This can be further defined in three groups:

1. Client to gateway
2. Gateway to gateway
3. Client to client

This process of broad communication support is only possible with detailed standards. IPSec (IP Security protocol) is an ever-growing standard to provide encrypted communications over IP. Its acceptance and robustness have fortified IPSec as the VPN technology standard for the foreseeable future. There are several RFCs that define IPSec, and currently there are over 40 Internet Engineering Task Force (IETF) RFC drafts that address various aspects of the standard's flexibility and growth.

Building Blocks of a Standard

The IPSec standard is used to provide privacy and authentication services at the IP layer. Several RFCs are used to describe this protocol suite. The interrelationship and organization of the documents are important to understand to become aware of the development process of the overall standard.

As Exhibit 32.1 shows, there are seven groups of documents that allow for the association of separate aspects of the IPSec protocol suite to be developed independently while a functioning relationship is attained and managed.

The Architecture is the main description document that covers the overall technology concepts and security considerations. It provides the access point for an initial understanding of the IPSec protocol suite.

The ESP (Encapsulating Security Payload) protocol (RFC 2406) and AH (Authentication Header) protocol (RFC 2402) document groups detail the packet formats and the default standards for packet structure that include implementation algorithms.

The Encryption Algorithm documents are a set of documents that detail the use of various encryption techniques utilized for the ESP. Examples of documents include DES (Data Encryption Standard RFC 1829) and Triple DES (draft-simpson-desx-02) algorithms and their application in the encryption of the data.

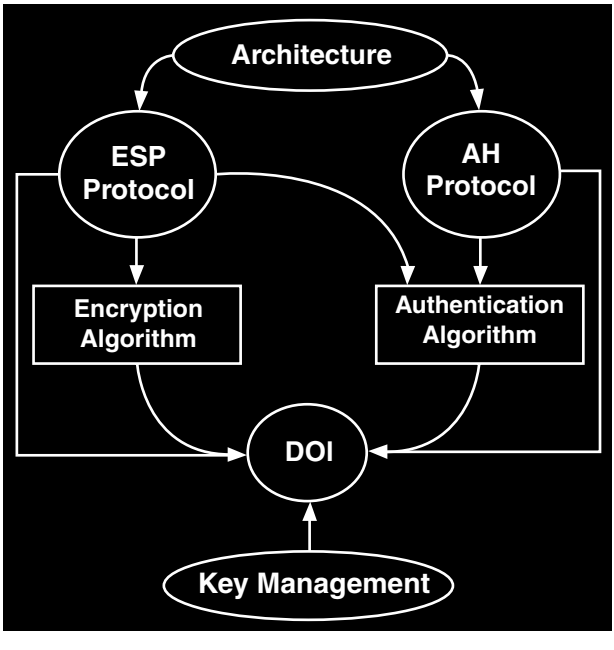


EXHIBIT 32.1 IETF IPSec DOI model.

The Authentication Algorithms are a group of documents describing the process and technologies used to provide an authentication mechanism for the AH and ESP protocols. Examples would be HMAC-MD5 (RFC 2403) and HMAC-SHA-1 (RFC 2404).

All of these documents specify values that must be consolidated and defined for cohesiveness into the DOI, or Domain of Interpretation (RFC 2407). The DOI document is part of the IANA assigned numbers mechanism and is a constant for many standards. It provides the central repository for values for the other documents to relate to each other. The DOI contains parameters that are required for the other portions of the protocol to ensure that the definitions are consistent.

The final group is Key Management, which details and tracks the standards that define key management schemes. Examples of the documents in this group are the Internet Security Association and Key Management Protocol (ISAKMP) and Public Key Infrastructure (PKI). This chapter unveils each of these protocols and the technology behind each that makes it the standard of choice in VPNs.

Introduction of Function

IPSec is a suite of protocols used to protect information, authenticate communications, control access, and provide non-repudiation. Of this suite there are two protocols that are the driving elements:

1. Authentication Header (AH)
2. Encapsulating Security Payload (ESP)

AH was designed for integrity, authentication, sequence integrity (replay resistance), and non-repudiation — but not for confidentiality for which the ESP was designed. There are various applications where the use of only an AH is required or stipulated. In applications where confidentiality is not required or not sanctioned by government encryption restrictions, an AH can be employed to ensure integrity, which in itself can be a powerful foe to potential attackers. This type of implementation does not protect the information from dissemination but will allow for verification of the integrity of the information and authentication of the originator. AH also provides protection for the IP header preceding it and selected options. The AH includes the following fields:

- IP Version
- Header Length
- Packet Length
- Identification
- Protocol
- Source and Destination Addresses
- Selected Options

The remainder of the IP header is not used in authentication with AH security protocol. ESP authentication does not cover any IP headers that precede it.

The ESP protocol provides encryption as well as some of the services of the AH. These two protocols can be used separately or combined to obtain the level of service required for a particular application or environmental structure. The ESP authenticating properties are limited compared to the AH due to the non-inclusion of the IP header information in the authentication process. However, ESP can be more than sufficient if only the upper layer protocols need to be authenticated. The application of only ESP to provide authentication, integrity, and confidentiality to the upper layers will increase efficiency over the encapsulation of ESP in the AH. Although authentication and confidentiality are both optional operations, one of the security protocols must be implemented. It is possible to establish communications with just authentication and without encryption or null encryption (RFC 2410). An added feature of the ESP is payload padding, which conceals the size of the packet being transmitted and further protects the characteristics of the communication.

The authenticating process of these protocols is necessary to create a security association (SA), the foundation of an IPSec VPN. An SA is built from the authentication provided by the AH or ESP protocol and becomes the primary function of key management to establish and maintain the SA between systems. Once the SA is achieved, the transport of data can commence.

Understanding the Foundation

Security associations are the infrastructure of IPSec. Of all the portions of IPSec protocol suite, the SA is the focal point for vendor integration and the accomplishment of heterogeneous virtual private networks. SAs are common among all IPSec implementations and must be supported to be IPSec compliant. An SA is nearly synonymous with VPN, but the term "VPN" is used much more loosely. SAs also exist in other security protocols. As described later, much of the key management used with IPSec VPNs is existing technology without specifics defining the underlying security protocol, allowing the key management to support other forms of VPN technology that use SAs.

SAs are simplex in nature in that two SAs are required for authenticated, confidential, bi-directional communications between systems. Each SA can be defined by three components:

1. Security parameter index (SPI)
2. Destination IP address
3. Security protocol identifier (AH or ESP)

An SPI is a 32-bit value used to distinguish among different SAs terminating at the same destination and using the same IPSec protocol. This data allows for the multiplexing of SAs to a single gateway. Interestingly, the destination IP address can be unicast, multicast, or broadcast; however, the standard for managing SAs currently applies to unicast applications or point-to-point SAs. Many vendors will use several SAs to accomplish a point-to-multipoint environment.

The final identification — the security protocol identifier — is the security protocol being utilized for that SA. Note that only one security protocol can be used for communications provided by a single SA. In the event that the communication requires authentication and confidentiality by use of both the AH and ESP security protocols, two or more SAs must be created and added to the traffic stream.

Finding the Gateway

Prior to any communication, it is necessary for a map to be constructed and shared among the community of VPN devices. This acts to provide information regarding where to forward data based on the required ultimate destination. A map can contain several pieces of data that exist to provide connection point information for a specific network and to assist the key management process. A map typically will contain a set of IP addresses that define a system, network, or groups of each that are accessible by way of a gateway's IP address.

An example of a map that specifies how to get to network 10.1.0.0 by a tunnel to 251.111.27.111 and use a shared secret with key management, might look like:

```
begin static -map
target "10.1.0.0/255.255.0.0"
mode "ISAKMP-Shared"
tunnel "251.111.27.111"
end
```

Depending on the vendor implemented, keying information and type may be included in the map. A shared secret or password may be associated with a particular destination. An example is a system that wishes to communicate with a remote network via VPN and needs to know the remote gateway's IP address and the expected authentication type when communication is initiated. To accomplish this, the map may contain mathematical representations of the shared secret in the map to properly match the secret with the destination gateway. A sample of this is a Diffie–Hellman key, explained in detail later.

Modes of Communication

The type of operation for IPSec connectivity is directly related to the role the system is playing in the VPN or the SA status. There are two modes of operation, as shown in Exhibit 32.2, for IPSec VPNs: transport mode and tunnel mode.

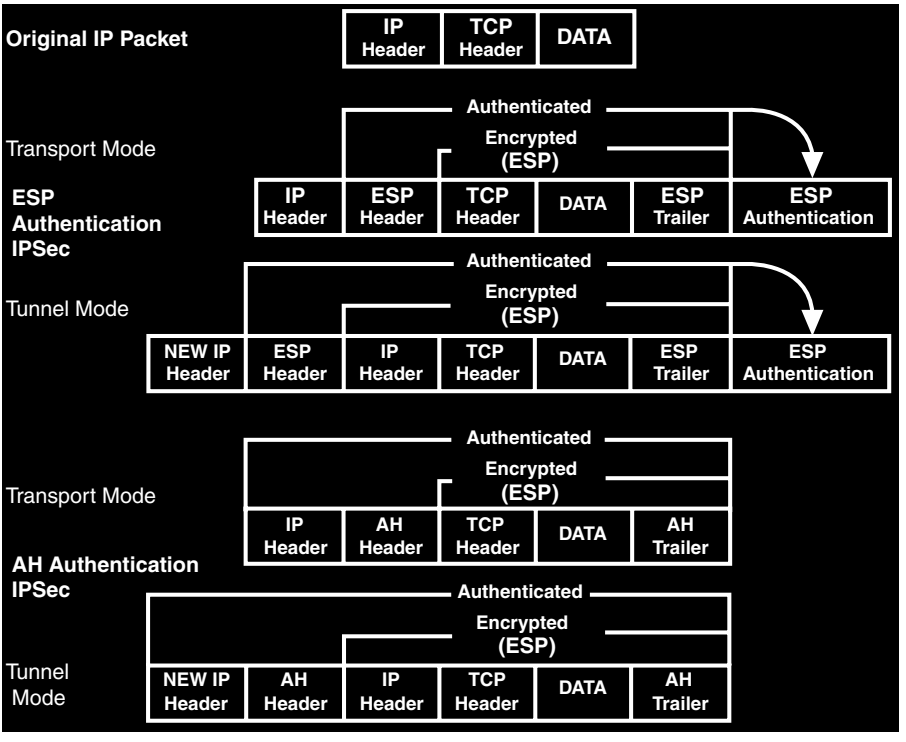


EXHIBIT 32.2 Tunnel and transport mode packet structure.

Transport mode is used to protect upper layer protocols and only affects the data in the IP packet. A more dramatic method, tunnel mode, encapsulates the entire IP packet to tunnel the communications in a secured communication.

Transport mode is established when the endpoint is a host, or when communications are terminated at the endpoints. If the gateway in gateway-to-host communications was to use transport mode, it would act as a host system, which can be acceptable for direct protocols to that gateway. Otherwise, tunnel mode is required for gateway services to provide access to internal systems.

Transport Mode

In transport mode, the IP packet contains the security protocol (AH or ESP) located after the original IP header and options and before any upper layer protocols contained in the packet, such as TCP and UDP. When ESP is utilized for the security protocol, the protection, or hash, is only applied to the upper layer protocols contained in the packet. The IP header information and options are not utilized in the authentication process. Therefore, the originating IP address cannot be verified for integrity against the data. With the use of AH as the security protocol, the protection is extended forward into the IP header to provide integrity of the entire packet by use of portions of the original IP header in the hashing process.

Tunnel Mode

Tunnel mode is established for gateway services and is fundamentally an IP tunnel with authentication and encryption. This is the most common mode of operation. Tunnel mode is required for gateway-to-gateway and host-to-gateway communications. Tunnel mode communications have two sets of IP headers — inside and outside.

The outside IP header contains the destination IP address of the VPN gateway. The inside IP header contains the destination IP address of the final system behind the VPN gateway. The security protocol appears after the

outer IP header and before the inside IP header. As with transport mode, extended portions of the IP header are utilized with AH that are not included with ESP authentication, ultimately providing integrity only of the inside IP header and payload.

The inside IP header's TTL (Time To Live) is decreased by one by the encapsulating system to represent the hop count as it passes through the gateway. However, if the gateway is the encapsulating system, as when NAT is implemented for internal hosts, the inside IP header is not modified. In the event the TTL is modified, the checksum must be recreated by IPSec and used to replace the original to reflect the change, maintaining IP packet integrity.

During the creation of the outside IP header, most of the entries and options of the inside header are mapped to the outside. One of these is ToS (Type of Service), which is currently available in IPv4.

Protecting and Verifying Data

The AH and ESP protocols can provide authentication or integrity for the data, and the ESP can provide encryption support for the data. The security protocol's header contains the necessary information for the accompanying packet. Exhibit 32.3 shows each header's format.

Authentication and Integrity

Security protocols provide authentication and integrity of the packet by use of a message digest of the accompanying data. By definition, the security protocols must use HMAC-MD5 or HMAC-SHA-1 for hashing functions to meet the minimum requirements of the standard. The security protocol uses a hashing algorithm to produce a unique code that represents the original data that was hashed and reduces the result into a reasonably sized element called a digest. The original message contained in the packet accompanying the hash can be hashed by the recipient and then compared to the original delivered by the source. By comparing the

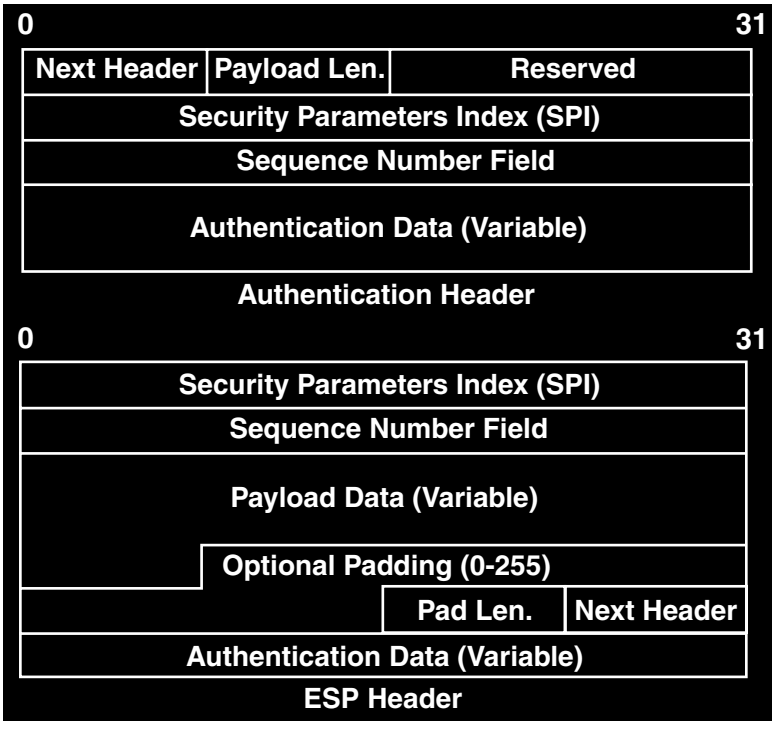


EXHIBIT 32.3 AH and ESP header format.

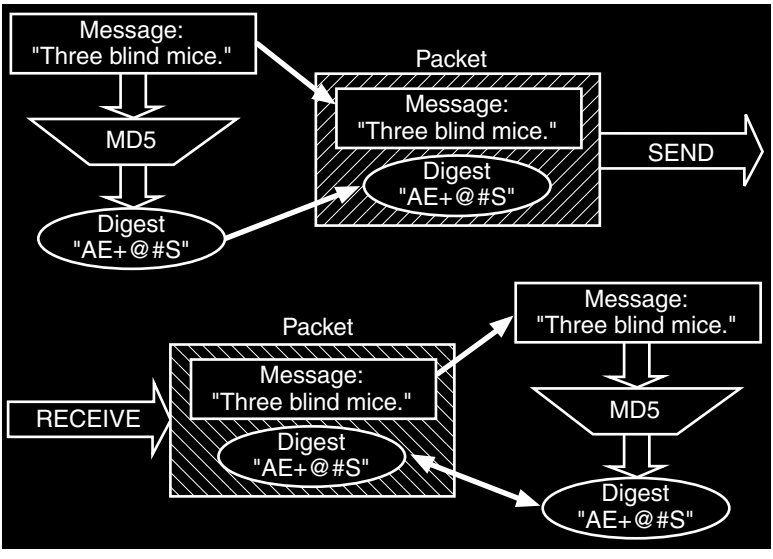


EXHIBIT 32.4 Message digest flow.

hashed results, it is possible to determine if the data was modified in transit. If they match, then the message was not modified. If the message hash does not match, then the data has been altered from the time it was hashed. Exhibit 32.4 shows the communication flow and comparison of the hash digest.

Confidentiality and Encryption

The two modes of operation affect the implementation of the ESP and the process of encrypting portions of the data being communicated. There is a separate RFC defining each form of encryption and the implementation of encryption for the ESP and the application in the two modes of communication. The standard requires that DES be the default encryption of the ESP. However, many forms of encryption technologies with varying degrees of strength can be applied to the standard. The current list is relatively limited due to the performance issues of high-strength algorithms and the processing required. With the advent of dedicated hardware for encryption processes and the advances in small, strong encryption algorithms such as ECC (Elliptic Curve Cryptosystems), the increase in VPN performance and confidentiality is inevitable.

In transport mode, the data of the original packet is encrypted and becomes the ESP. In tunnel mode, the entire original packet is encrypted and placed into a new IP packet in which the data portion is the ESP containing the original encrypted packet.

Managing Connections

As mentioned earlier, SAs furnish the primary purpose of the IPSec protocol suite and the relationship between gateways and hosts. Several layers of application and standards provide the means for controlling, managing, and tracking SAs.

Various applications may require the unification of services, demanding combined SAs to accomplish the required transport. An example would be an application that requires authentication and confidentiality by utilizing AH and ESP and requires that further groups of SAs provide hierarchical communication. This process is called an SA Bundle, which can provide a layered effect of communications. SA bundles can be utilized by applications in two formats: fine granularity and coarse granularity.

Fine granularity is the assignment of SAs for each communication process. Data transmitted over a single SA is protected by a single security protocol. The data is protected by an AH or ESP, but not both because SAs can have only one security protocol.

Coarse granularity is the combination of services from several applications or systems into a group or portion of an SA bundle. This affords the communication two levels of protection by way of more than one SA.

Exhibit 32.5 conveys the complexity of SAs, and the options available become apparent considering that SAs in a SA bundle can terminate at different locations.

Consider the example of a host on the Internet that established a tunnel-mode SA with a gateway and a transport-mode SA to the final destination internal host behind the gateway. This implementation affords the protection of communications over an untrusted medium and further protection once on the internal network for point-to-point secured communications. It also requires an SA bundle that terminates at different destinations.

There are two implementations of SA Bundles:

1. Transport adjacency
2. Iterated tunneling

Transport adjacency involves applying more than one security protocol to the same IP datagram without implementing tunnel mode for communications. Using both AH and ESP provides a single level of protection and no nesting of communications because the endpoint of the communication is the final destination. This

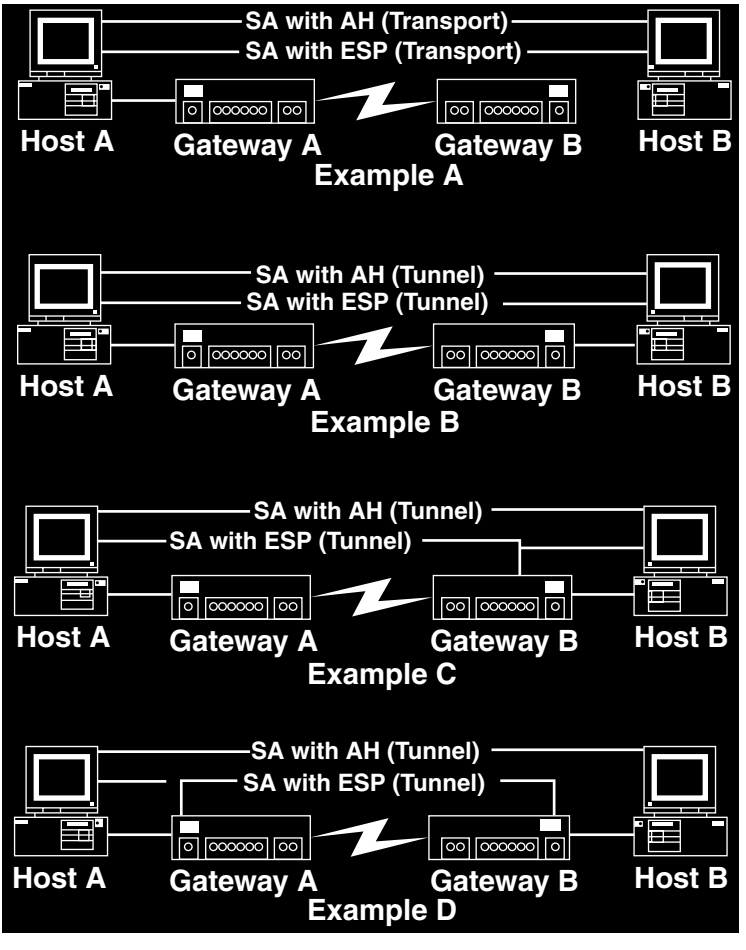


EXHIBIT 32.5 SA types.

application of transport adjacency is applied when transport mode is implemented for communication between two hosts, each behind a gateway. (See Exhibit 32.5: Example A.)

In contrast, iterated tunneling is the application of multiple layers of security protocols within a tunnel-mode SA(s). This allows for multiple layers of nesting because each SA can originate or terminate at different points in the communication stream. There are three occurrences of iterated tunneling:

- Endpoints of each SA are identical
- One of the endpoints of the SAs is identical
- Neither endpoint of the SAs is identical

Identical endpoints can refer to tunnel-mode communications between two hosts behind a set of gateways where SAs terminate at the hosts and AH (or ESP) is contained in an ESP providing the tunnel. (See Exhibit 32.5: Example B.)

With only one of the endpoints being identical, an SA can be established between the host and gateway and between the host and an internal host behind the gateway. This was used earlier as an example of one of the applications of SA Bundling. (See Exhibit 32.5: Example C.)

In the event of neither SA terminating at the same point, an SA can be established between two gateways and between two hosts behind the gateways. This application provides multi-layered nesting and communication protection. An example of this application is a VPN between two gateways that provide tunnel mode operations for their corresponding networks to communicate. Hosts on each network are provided secured communication based on client-to-client SAs. This provides for several layers of authentication and data protection. (See Exhibit 32.5: Example D.)

Establishing a VPN

Now that the components of a VPN have been defined, it is necessary to discuss the form that they create when combined. To be IPSec compliant, four implementation types are required of the VPN. Each type is merely a combination of options and protocols with varying SA control. The four detailed here are only the required formats, and vendors are encouraged to build on the four basic models.

The VPNs shown in Exhibit 32.6 can use either security protocol. The mode of operation is defined by the role of the endpoint — except in client-to-client communications, which can be transport or tunnel mode.

In Example A, two hosts can establish secure peer communications over the Internet. Example B illustrates a typical gateway-to-gateway VPN with the VPN terminating at the gateways to provide connectivity for internal hosts. Example C combines Examples A and B to allow secure communications from host to host in an existing gateway-to-gateway VPN. Example D details the situation when a remote host connects to an ISP, receives an IP address, and then establishes a VPN with the destination network's gateway. A tunnel is established to the gateway, and then a tunnel- or transport-mode communication is established to the internal system. In this example, it is necessary for the remote system to apply the transport header prior to the tunnel header. Also, it will be necessary for the gateway to allow IPSec connectivity and key management protocols from the Internet to the internal system.

Keeping Track

Security associations and the variances of their applications can become complicated; levels of security, security protocol implementation, nesting, and SA Bundling all conspire to inhibit interoperability and to decrease management capabilities. To ensure compatibility, fundamental objectives are defined to enable coherent management and control of SAs. There are two primary groups of information, or databases, that are required to be maintained by any system participating in an IPSec VPN Security Policy Database (SPD) and Security Association Database (SAD).

The SPD is concerned with the status, service, or character provided by the SA and the relationships provided. The SAD is used to maintain the parameters of each active association. There are a minimum of two of each database — one for tracking inbound and another for outbound communications.

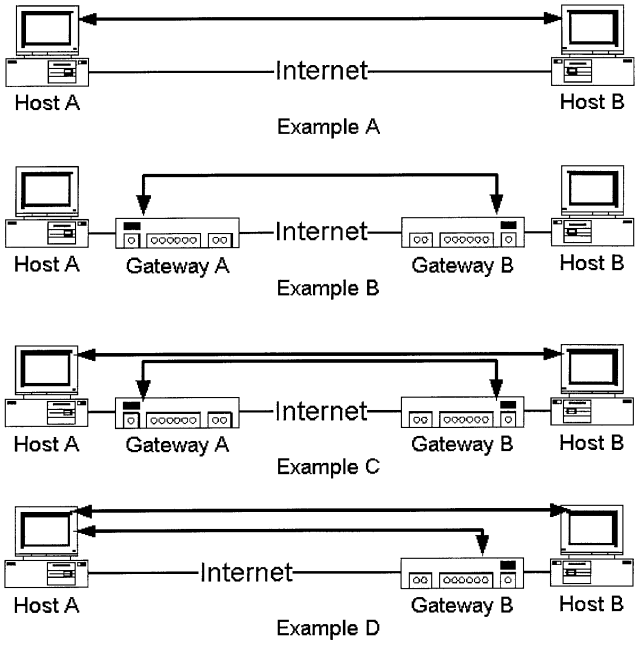


EXHIBIT 32.6 VPN TYPES.

Communication Policies

The SPD is a security association management constructed to enforce a policy in the IPSec environment. Consequently, an essential element of SA processing is an underlying security policy that specifies what services are to be offered to IP datagrams and in what fashion they are implemented. SPD is consulted for all IP and IPSec communications, inbound and outbound, and therefore is associated with an interface. An interface that provides IPSec, and ultimately is associated with an SPD, is called a "black" interface. An interface where IPSec is not being performed is called a "red" interface and no data is encrypted for this network by that gateway. The number of SPDs and SADs are directly related to the number of black and red interfaces being supported by the gateway. The SPD must control traffic that is IPSec based and traffic that is not IPSec related. There are three modes of this operation:

1. Forward and do not apply IPSec
2. Discard packet
3. Forward and apply IPSec

In the policy, or database, it is possible to configure traffic that is only IPSec to be forwarded, hence providing a basic firewall function by allowing only IPSec protocol packets into the black interface. A combination will allow multi-tunneling, a term that applies to gateways and hosts. It allows the system to discriminate and forward traffic based on destination, which ultimately determines if the data is encrypted or not. An example is to allow basic browsing from a host on the Internet while providing a secured connection to a remote gateway on the same connection. A remote user may dial an ISP and establish a VPN with the home office to get their mail. While receiving the mail, the user is free to access services on the Internet using the local ISP connection to the Internet.

If IPSec is to be applied to the packet, the SPD policy entry will specify a SA or SA bundle to be employed. Within the specification are the IPSec protocols, mode of operation, encryption algorithms, and any nesting requirements.

A *selector* is used to apply traffic to a policy. A security policy may determine several SAs be applied for an application in a defined order, and the parameters of this bundled operation must be detailed in the SPD. An example policy entry may specify that all matching traffic be protected by an ESP using DES, nested inside an AH using SHA-1. Each selector is employed to associate the policy to SAD entries.

The policies in the SPD are maintained in an ordered list. Each policy is associated with one or more selectors. Selectors define the IP traffic that characterizes the policy. Selectors have several parameters that define the communication to policy association, including:

- Destination IP address
- Source IP address
- Name
- Data sensitivity
- Transport protocol
- Source and destination TCP ports

Destination address may be unicast, multicast, broadcast, a range of addresses, or a wildcard address. Broadcast, range, and wildcard addresses are used to support more than one destination using the same SA. The destination address defined in the selector is not the destination that is used to define an SA in the SAD (SPI, destination IP address, and IPSec protocol). The destination from the SA identifier is used as the packet arrives to identify the packet in the SAD. The destination address within the selector is obtained from the encapsulating IP header. Once the packet has been processed by the SA and un-encapsulated, its selector is identified by the IP address and associated to the proper policy in the inbound SPD. This issue does not exist in transport mode because only one IP header exists. The source IP address can be any of the types allowed by the destination IP address field.

There are two sets of names that can be included in the Name field: User ID and System Name.

User ID can be a user string associated with a fully qualified domain name (FQDN), as with person@company.com. Another accepted form of user identification is X.500 distinguished name. An example of this type of name could be: C=US,O=Company,OU=Finance,CN=Person. System Name can be a FQDN, box.company.com, or an X.500 distinguished name.

Data sensitivity defines the level of security applied to that packet. This is required for all systems implemented in an environment that uses data labels for information security flow.

Transport protocol and port are obtained from the header. These values may not be available because of the ESP header or not mapped due to options being utilized in the originating IP header.

Security Association Control

The SPD is policy driven and is concerned with system relationships. However, the SAD is responsible for each SA in the communications defined by the SPD. Each SA has an entry in the SAD. The SA entries in the SAD are indexed by the three SA properties: destination IP address, IPSec protocol, and SPI. The SAD database contains nine parameters for processing IPSec protocols and the associated SA:

1. Sequence number counter for outbound communications
2. Sequence number overflow counter that sets an option flag to prevent further communications utilizing the specific SA
3. A 32-bit anti-replay window that is used to identify the packet for that point in time traversing the SA and provides the means to identify that packet for future reference
4. Lifetime of the SA that is determined by a byte count or timeframe, or a combination of the two
5. The algorithm used in the AH
6. The algorithm used in the authenticating the ESP
7. The algorithm used in the encryption of the ESP
8. IPSec mode of operation: transport or tunnel mode
9. Path MTU (PMTU) (this is data that is required for ICMP data over an SA)

Each of these parameters is referenced in the SPD for assignment to policies and applications. The SAD is responsible for the lifetime of the SA, which is defined in the security policy. There are two lifetime settings for each SA: soft lifetime and hard lifetime.

Soft lifetime determines a point when to initiate the process to create a replacement SA. This is typical for rekeying procedures. Hard lifetime is the point where the SA expires. If a replacement SA has not been established, the communications will discontinue.

Providing Multi-Layered Security Flow

There are many systems that institute multi-layered security (MLS), or data labeling, to provide granularity of security based on the data and the systems it may traverse while on the network. This model of operation can be referred to as Mandatory Access Control (MAC). An example of this security model is the Bell–LaPadula model, designed to protect against the unauthorized transmission of sensitive information. Because the data itself is tagged for review while in transit, several layers of security can be applied. Other forms of security models such as Discretionary Access Control (DAC) that may employ access control lists or filters are not sufficient to support multi-layer security. The AH and ESP can be combined to provide the necessary security policy that may be required for MLS systems working in a MAC environment.

This is accomplished using the authenticating properties of the AH security protocol to bind security mappings in the original IP header to the payload. Using the AH in this manner allows the authentication of the data against the header. Currently, IPv4 does not validate the payload with the header. The sensitivity of the data is assumed only by default of the header.

To accomplish this process each SA, or SA Bundle, must be discernable from other levels of secured information being transmitted. An example is: "SENSITIVE" labeled data will be mapped to a SA or a SA Bundle, while "CLASSIFIED" labeled data will be mapped to others. The SAD and SPD contain a parameter called *Sensitivity Information* that can be accessed by various implementations to ensure that the data being transferred is afforded the proper encryption level and forwarded to the associated SAs.

There are two forms of processing when MAC is implemented:

1. Inbound operation
2. Outbound operation

When a packet is received and passed to the IPSec functions, the MLS must verify the sensitivity information level prior to passing the datagram to upper layer protocols or forwarding. The sensitivity information level is then bound to the associated SA and stored in the SPD to properly apply policies for that level of secured data.

Outbound requirements of the MLS are to ensure that the selection of a SA, or SA Bundle, is appropriate for the sensitivity of the data, as defined in the policy. The data for this operation is contained in the SAD and SPD, which is modified by defined policies and the previous inbound operations.

Implementations of this process are vendor driven. Defining the level of encryption, type of authentication, key management scheme, and other security-related parameters associated with a data label are available for vendors to implement. The mechanism for defining policies that can be applied is accessible and vendors are beginning to become aware of these options as comfort and maturity of the IPSec standard are realized.

A Key Point

Key management is an important aspect of IPSec or any encrypted communication that uses keys to provide information confidentiality and integrity. Key management and the protocols utilized are implemented to set up, maintain, and control secure relationships and ultimately the VPN between systems. During key management, there are several layers of system insurance prior to the establishment of an SA, and there are several mechanisms used to accommodate these processes.

Key History

Key management is far from obvious definition, and lackadaisical conversation with interchanged acronyms only adds to the perceived misunderstandings. The following is an outline of the different protocols that are used to get keys and data from one system to another.

The Internet Security Association and Key Management Protocol (ISAKMP) (RFC 2408) defines the procedures for authenticating a communicating peer and key generation techniques. All of these are necessary to establish and maintain an SA in an Internet environment. ISAKMP defines payloads for exchanging key and authentication data. As shown Exhibit 32.7, these formats provide a consistent framework that is independent of the encryption algorithm, authentication mechanism being implemented, and security protocol, such as IPSec.

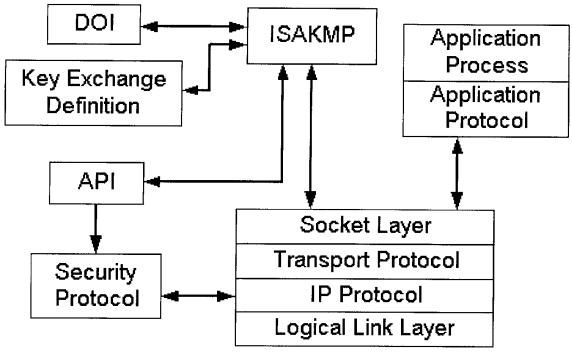


EXHIBIT 32.7 ISAKMP structure.

The Internet Key Exchange (IKE) protocol (RFC 2409) is a hybrid containing three primary, existing protocols that are combined to provide an IPSec-specific key management platform. The three protocols are:

1. ISAKMP
2. Oakley
3. SKEME (Secure Key Exchange Mechanism)

Different portions of each of these protocols work in conjunction to securely provide keying information specifically for the IETF IPSec DOI. The terms IKE and ISAKMP are used interchangeably by various vendors, and many use ISAKMP to describe the keying function. While this is correct, ISAKMP addresses the procedures and not the technical operations as they pertain to IPSec. IKE is the term that best represents the IPSec implementation of key management.

Public Key Infrastructure (PKI) is a suite of protocols that provide several areas of secure communication based on trust and digital certificates. PKI integrates digital certificates, public key cryptography, and certificate authorities into a total, enterprisewide network security architecture that can be utilized by IPSec.

IPSec IKE

As described earlier, IKE is a combination of several existing key management protocols that are combined to provide a specific key management system. IKE is considerably complicated, and several variations are available in the establishment of trust and providing keying material.

Oakley and ISAKMP protocols, which are included in IKE, each define separate methods of establishing an authenticated key exchange between systems. Oakley defines *modes* of operation to build a secure relationship path, and ISAKMP defines *phases* to accomplish much the same process in a hierarchical format. The relationship between these two is represented by IKE with different exchanges as modes, which operate in one of two phases. Implementing multiple phases may add overhead in processing, resulting in performance degradation, but several advantages can be realized. Some of these are:

- First phase creation assisted by second phase
- First phase key material used in second phase
- First phase trust used for second phase

The first phase session can be disbursed among several second phase operations to provide the construction of new ISAKMP security associations (ISA for purposes of clarity in this document) without the renegotiation process between the peers. This allows for the first phase of subsequent ISAs to be preempted via communications in the second phase.

Another benefit is that the first phase process can provide security services for the second phase in the form of encryption keying material. However, if the first phase does not meet the requirements of the second phase, no data can be exchanged or provided from the first to the second phase.

With the first phase providing peer identification, the second phase may provide the creation of the security protocol SAs without the concern for authentication of the peer. If the first phase were not available, each new SA would need to authenticate the peer system. This function of the first phase is an important feature for IPSec communications. Once peers are authenticated by means of certificates or shared secret, all communications of the second phase and internal to the IPSec SAs are authorized for transport. The remaining authentication is for access control. By this point, the trusted communication has been established at a higher level.

Phases and Modes

Phase one takes place when the two ISAKMP peers establish a secure, authenticated channel with which to communicate. Each system is verified and authenticated against its peer to allow for future communications. Phase two exists to provide keying information and material to assist in the establishment of SAs for an IPSec communication.

Within phase one, there are two modes of operation defined in IKE: main mode and aggressive mode. Each of these accomplishes a phase one secure exchange, and these two modes only exist in phase one. Within phase two, there are two modes: Quick Mode and New Group Mode.

Quick Mode is used to establish SAs on behalf of the underlying security protocol. New Group Mode is designated as a phase two mode only because it must exist in phase two; however, the service provided by New Group Mode is to benefit phase one operations. As described earlier, one of the advantages of a two-phase approach is that the second phase can be used to provide additional ISAs, which eliminates the reauthorization of the peers.

Phase one is initiated using ISAKMP-defined cookies. The initiator cookie (I-cookie) and responder cookie (R-cookie) are used to establish an ISA, which provides end-to-end authenticated communications. That is, ISAKMP communications are bi-directional and, once established, either peer may initiate a Quick Mode to establish SA communications for the security protocol. The order of the cookies is crucial for future second phase operations. A single ISA can be used for many second phase operations, and each second phase operation can be used for several SAs or SA Bundles. Main Mode and Aggressive Mode each use Diffie–Hellman keying material to provide authentication services.

While Main Mode must be implemented, Aggressive Mode is not required. Main Mode provides several messages to authenticate. The first two messages determine a communication policy; the next two messages exchange Diffie–Hellman public data; and the last two messages authenticate the Diffie–Hellman Exchange. Aggressive Mode is an option available to vendors and developers that provides much more information with fewer messages and acknowledgments. The first two messages in Aggressive Mode determine a communication policy and exchange Diffie–Hellman public data. In addition, a second message authenticates the responder, thus completing the negotiation.

Phase two is much simpler in nature in that it provides keying material for the initiation of SAs for the security protocol. This is the point where key management is utilized to maintain the SAs for IPSec communications. The second phase has one mode designed to support IPSec: Quick Mode. Quick Mode verifies and establishes the keying process for the creation of SAs. Not related directly to IPSec SAs is the New Group Mode of operation; New Group provides services for phase one for the creation of additional ISAs.

System Trust Establishment

The first step in establishing communications is verification of the remote system. There are three primary forms of authenticating a remote system:

1. Shared secret
2. Certificate
3. Public/private key

Of these methods, shared secret is currently used widely due to the relatively slow integration of Certificate Authority (CA) systems and the ease of implementation. However, shared secret is not scalable and can become unmanageable very quickly due to the fact that there can be a separate secret for each communication. Public and private key use is employed in combination with Diffie–Hellman to authenticate and provide keying material. During the system authentication process, hashing algorithms are utilized to protect the authenti-

cating shared secret as it is forwarded over untrusted networks. This process of using hashing to authenticate is nearly identical to the authentication process of an AH security protocol. However, the message — in this case a password — is not sent with the digest. The map previously shared or configured with participating systems will contain the necessary data to be compared to the hash.

An example of this process is a system, called system A, that requires a VPN to a remote system, called system B. By means of a preconfigured map, system A knows to sends its hashed shared secret to system B to access a network supported by system B. System B will hash the expected shared secret and compare it to the hash received from system A. If the two hashes match, an authenticated trust relationship is established.

Certificates are a different process of trust establishment. Each device is issued a certificate from a CA. When a remote system requests communication establishment, it will present its certificate. The recipient will query the CA to validate the certificate. The trust is established between the two systems by means of an ultimate trust relationship with the CA and the authenticating system. Seeing that certificates can be made public and are centrally controlled, there is no need to attempt to hash or encrypt the certificate.

Key Sharing

Once the two systems are confident of each other's identity, the process of sharing or swapping keys must take place to provide encryption for future communications. The mechanisms that can be utilized to provide keying are related to the type of encryption to be utilized for the ESP. There are two basic forms of keys: symmetrical and asymmetrical.

Symmetrical key encryption occurs when the same key is used for the encryption of information into human unintelligible data (or ciphertext) and the decryption of that ciphertext into the original information format. If the key used in symmetrical encryption is not carefully shared with the participating individuals, an attacker can obtain the key, decrypt the data, view or alter the information, encrypt the data with the stolen key, and forward it to the final destination. This process is defined as a man-in-the-middle attack and, if properly executed, can affect data confidentiality and integrity, rendering the valid participants in the communication oblivious to the exposure and the possible modification of the information.

Asymmetrical keys consist of a key-pair that is mathematically related and generated by a complicated formula. The concept of asymmetrical comes from the fact that the encryption is one way with either of the key-pair, and data that is encrypted with one key can only be decrypted with the other key of the pair. Asymmetrical key encryption is incredibly popular and can be used to enhance the process of symmetrical key sharing. Also, with the use of two keys, digital signatures have evolved and the concept of trust has matured to certificates, which contribute to a more secure relationship.

One Key

Symmetrical keys are an example of DES encryption, where the same keying information is used to encrypt and decrypt the data. However, to establish communications with a remote system, the key must be made available to the recipient for decryption purposes. In early cases, this may have been a phone call, e-mail, fax, or some form of nonrelated communication medium. However, none of these options are secure or can communicate strong encryption keys that require a sophisticated key that is nearly impossible to convey in a password or phrase.

In 1976, two mathematicians, Bailey W. Diffie at Berkeley and Martin E. Hellman at Stanford, defined the Diffie–Hellman agreement protocol (also known as exponential key agreement) and published it in a paper entitled "New Directions in Cryptography." The protocol allows two autonomous systems to exchange a secret key over an untrusted network without any prior secrets. Diffie and Hellman postulated that the generation of a key could be accomplished by fundamental relationships between prime numbers. Some years later, Ron Rivest, Adi Shamir, and Leonard Adelman, who developed the RSA Public and Private key cryptosystem based on large prime numbers, further developed the Diffie–Hellman formula (i.e., the nuts and bolts of the protocol). This allowed communication of a symmetrical key without transmitting the actual key, but rather a mathematical portion or fingerprint.

An example of this process is system A and system B require keying material for the DES encryption for the ESP to establish an SA. Each system acquires the Diffie–Hellman parameters, a large prime number p and

a base number *g*, which must be smaller than $p - 1$. The generator, *g*, is a number that represents every number between 1 and *p* to the power of *k*. Therefore, the relationship is $g^k = \text{n mod } p$.

Both of these numbers must be hardcoded or retrieved from a remote system. Each system then generates a number X, which must be less than $p - 2$. The number X is typically created by a random string of characters entered by a user or a passphrase that can be combined with date and time to create a unique number. The hardcoded numbers will not be exceeded because most, if not all, applications employ a limit on the input.

As shown in Exhibit 32.8, a new key is generated with these numbers, $g^X \text{mod } p$. The result Y, or fingerprint, is then shared between the systems over the untrusted network. The formula is then exercised again using the shared data from the other system and the Diffie–Hellman parameters. The results will be mathematically equivalent and can be used to generate a symmetrical key. If each system executes this process successfully, they will have matching symmetrical keys without transmitting the key itself. The Diffie–Hellman protocol was finally patented in 1980 (U.S. Patent 4200770) and is such a strong protocol that there are currently 128 other patents that reference Diffie–Hellman.

To complicate matters, Diffie–Hellman is vulnerable to man-in-the-middle attacks because the peers are not authenticated using Diffie–Hellman. The process is built on the trust established prior to keying material creation. To provide added authentication properties within the Diffie–Hellman procedure, the Station-to-Station (STS) protocol was created. Diffie, Oorschot, and Wiener completed STS in 1992 by allowing the two parties to authenticate themselves to each other by the use of digital signatures created by a public and private key relationship.

An example of this process, as shown in Exhibit 32.9, transpires when each system is provided a public and private key-pair. System A will encrypt the Y value (in this case Ya) with the private key. When system B receives the signature, it can only be decrypted with the system A public key. The only plausible result is that system A encrypted the Ya value authenticating system A. The STS protocol allows for the use of certificates to further authorize the public key of system A to ensure that the man-in-the-middle has not compromised the key-pair integrity.

Many Keys

Asymmetrical keys, such as PGP (Pretty Good Privacy) and RSA, can be used to share the keying information. Asymmetrical keys were specifically designed to have one of the keys in a pair published. A sender of data can

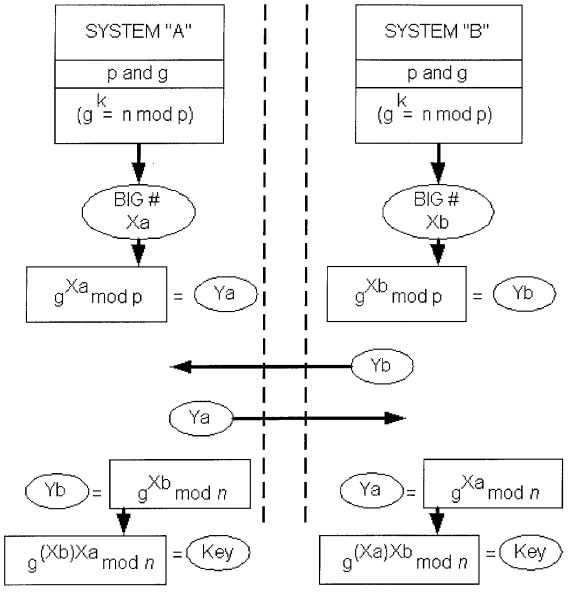


EXHIBIT 32.8 Diffie–Hellman exchange protocol.

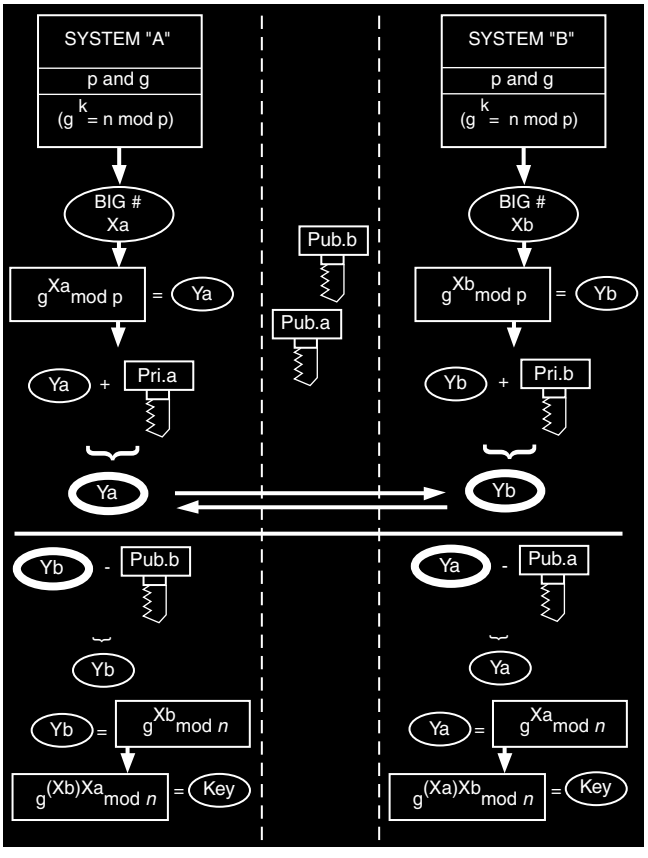


EXHIBIT 32.9 Diffie–Hellman exchange protocol with STS.

obtain the public key of the preferred recipient to encrypt data that can only be decrypted by the holder of the corresponding private key. The application of asymmetrical keys in the sharing of information does not require the protection of the public key in transit over an untrusted network.

Key Establishment

The IPSec standard mandates that key management must support two forms of key establishment: manual and automatic.

The other IPSec protocols (AH and ESP) are not typically affected by the type of key management. However, there may be issues with implementing anti-replay options, and the level of authentication can be related to the key management process supported. Indeed, key management can also be related to the ultimate security of the communication. If the key is compromised, the communication can be in danger of attack. To thwart the eventuality of such an attack, there are re-keying mechanisms that attempt to ensure that if a key is compromised its validity is limited either by time, amount of data encrypted, or a combination of both.

Manual Keying

Manual key management requires that an administrator provide the keying material and necessary security association information for communications. Manual techniques are practical for small environments with limited numbers of gateways and hosts. Manual key management does not scale to include many sites in a

meshed or partially meshed environment. An example is a company with five sites throughout North America. This organization wants to use the Internet for communications, and each office site must be able to communicate directly with any other office site. If each VPN relationship had a unique key, the number of keys can be calculated by the formula $n(n-1)/2$, where n is the number of sites. In this example, the number of keys is 10. Apply this formula to 25 sites (i.e., five times the number of sites in the previous example) and the number of keys skyrockets to 300, not 50. In reality, the management is more difficult than it may appear by the examples. Each device must be configured, and the keys must be shared with all corresponding systems. The use of manual keying conspires to reduce the flexibility and options of IPSec. Anti-replay, on-demand re-keying, and session-specific key management are not available in manual key creation.

Automatic Keying

Automatic key management responds to the limited manual process and provides for widespread, automated deployment of keys. The goal of IPSec is to build off existing Internet standards to accommodate a fluid approach to interoperability. As described earlier, the IPSec default automated key management is IKE, a hybrid based in ISAKMP. However, based on the structure of the standard, any automatic key management can be employed. Automated key management, when instituted, may create several keys for a single SA. There are various reasons for this, including:

- Encryption algorithm requires more than one key
- Authentication algorithm requires more than one key
- Encryption and authentication are used for a single SA
- Re-keying

The encryption and authentication algorithms' use of multiple keys, or if both algorithms are used, then multiple keys will need to be generated for the SA. An example of this would be if Triple-DES is used to encrypt the data. There are several types of applications of Triple-DES (DES-EEE3, DES-EDE3, and DES-EEE2) and each uses more than one key (DES-EEE2 uses two keys, one of which is used twice).

The process of re-keying is to protect future data transmissions in the event a key is compromised. This process requires the rebuilding of an existing SA. The concept of re-keying during data transmission provides a relatively unpredictable communication flow. Being unpredictable is considered a valuable security method against an attacker.

Automatic key management can provide two primary methods of key provisioning:

1. Multiple string
2. Single string

Multiple strings are passed to the corresponding system in the SA for each key and for each type. For example, the use of Triple-DES for the ESP will require more than one key to be generated for a single type of algorithm, in this case, the encryption algorithm. The recipient will receive a string of data representing a single key; once the transfer has been acknowledged, the next string representing another key will be transmitted.

In contrast, the single string method sends all the required keys in a single string. As one might imagine, this requires a stringent set of rules for management. Great attention is necessary to ensure that the systems involved properly map the corresponding bits to the same key strings for the SA being established. To ensure that IPSec-compliant systems properly map the bit to keys, the string is read from the left, highest bit order first for the encryption key(s) and the remaining string is used for the authentication. The number of bits used is determined by the encryption algorithm and the number of keys required for the encryption being utilized for that SA.

Technology Turned Mainstream

VPNs are making a huge impact on the way communications are viewed. They are also providing ample fodder for administrators and managers to have seemingly endless discussions about various applications. On one side are the possible money savings, and the other are implementation issues. There are several areas of serious concern, including:

- Performance
- Interoperability
- Scalability
- Flexibility

Performance

Performance of data flow is typically the most common concern, and IPSec is very processor intensive. The performance costs of IPSec are the encryption being performed, integrity checking, packet handling based on policies, and forwarding, all of which become apparent in the form of latency and reduced throughput. IPSec VPNs over the Internet increase the latency in the communication that conspires with the processing costs to discourage VPN as a solution for transport-sensitive applications. Process time for authentication, key management, and integrity verification will produce delay issues with SA establishment, authentication, and IPSec SA maintenance. Each of these results in poor initialization response and, ultimately, disgruntled users.

The application of existing hardware encryption technology to IPSec vendor products has allowed these solutions to be considered more closely by prospective clients wishing to seize the monetary savings associated with the technology. The creation of a key and its subsequent use in the encryption process can be offloaded onto a dedicated processor that is designed specifically for these operations. Until the application of hardware encryption for IPSec, all data was managed through software computation that was also responsible for many other operations that may be running on the gateway.

Hardware encryption has released IPSec VPN technology into the realm of viable communication solutions. Unfortunately, the client operating system participating in a VPN is still responsible for the IPSec process. Publicly available mobile systems that provide hardware-based encryption for IPSec communications are becoming available, but are some time away from being standard issue for remote users.

Interoperability

Interoperability is a current issue that will soon become antiquated as vendors recognize the need to become fully IPSec compliant — or consumers will not implement their product based simply on its incompatibility. Shared secret and ISAKMP key management protocol are typically allowing multi-vendor interoperability. As Certificate Authorities and the technology that supports them become fully adopted technology, they will only add to the cross-platform integration. However, complex and large VPNs will not be manageable using different vendor products in the near future. Given the complexity, recentness of the IPSec standard, and the various interpretations of that standard, the time to complete interoperability seems great.

Scalability

Scalability is obtained by the addition of equipment and bandwidth. Some vendors have created products focused on remote access for roaming users, while others have concentrated on network-to-network connectivity without much attention to remote users. The current ability to scale the solution will be directly related to the service required. The standard supporting the technology allows for great flexibility in the addition of services. It will be more common to find limitations in equipment configurations than in the standard as it pertains to growth capabilities. Scalability ushers in a wave of varying issues, including:

- Authentication
- Management
- Performance

Authentication can be provided by a number of processes, although the primary focus has been on RADIUS (Remote Access Dial-In User Security), Certificates, and forms of two-factor authentication. Each of these can be applied to several supporting databases. RADIUS is supported by nearly every common authenticating system, from Microsoft Windows NT to NetWare's NDS. Authentication, when implemented properly, should not become a scalability issue for many implementations, because the goal is to integrate the process with existing or planned enterprise authenticating services.

A more interesting aspect of IPSec vendor implementations and the scalability issues that might arise is management. As detailed earlier, certain implementations do not scale, due to the shear physics of shared secrets and manual key management. In the event of the addition of equipment or increased bandwidth to support remote applications, the management will need to take multiplicity into consideration. Currently, VPN management of remote users and networks leaves a great deal to be desired. As vendors and organizations become more acquainted with what can be accomplished, sophisticated management capabilities will become increasingly available.

Performance is an obvious issue when considering the increase of an implementation. Typically, performance is the driving reason, followed by support for increased numbers. Both of these issues are volatile and interrelated with the hardware technology driving the implementation. Performance capabilities can be controlled by the limitation of supported SAs on a particular system — a direct limitation in scalability. A type of requested encryption might not be available on the encryption processor currently available. Forcing the calculation of encryption onto the operating system ultimately limits the performance. A limitation may resonate in the form of added equipment to accomplish the link between the IPSec equipment and the authenticating database. When users authenticate, the granularity of control over the capabilities of that user may be directly related to the form of authentication. The desired form of authentication may have limitations in various environments due to restrictions in various types of authenticating databases. Upgrade issues, service pack variations, user limitations, and protocol requirements also combine to limit growth of the solution.

The Market for VPN

Several distinct qualities of VPN are driving the investigation by many organizations to implement VPN as a business interchange technology. VPNs attempt to resolve a variety of current technological limitations that represent themselves as costs in equipment and support or solutions where none had existed prior. Three areas that can be improved by VPNs are:

1. Remote user access and remote office connectivity
2. Extranet partner connectivity
3. Internal departmental security

Remote Access

Providing remote user access via a dial-up connection can become a costly service for any organization to provide. Organizations must consider costs for:

- Telephone lines
- Terminating equipment
- Long-distance
- Calling card
- 800/877 number support

Telephone connections must be increased to support the number of proposed simultaneous users that will be dialing in for connectivity to the network. Another cost that is rolled up into the telephone line charge is the possible need for equipment to allow the addition of telephone lines to an existing system. Terminating equipment, such as modem pools, can become expenses that are immediate savings once the VPN is utilized. Long-distance charges, calling cards that are supplied to roaming users, and toll-free lines require initial capital and continuous financial support. In reality, an organization employing conventional remote access services is nothing more than a service provider for its employees. Taking this into consideration, many organizations tend to overlook the use of the Internet connection by remote users. As the number of simultaneous users access the network, the more bandwidth is utilized for the existing Internet service.

The cost savings are realized by redirecting funds, originally to support telephone communications, in an Internet service provider (ISP) and its ability to support a greater area of access points and technology. This allows an organization to eliminate support for all direct connectivity and focus on a single connection and technology for all data exchange — ultimately saving money. With the company access point becoming a single point of entry, access controls, authenticating mechanisms, security policies, and system redundancy become focused and common among all types of access regardless of the originator's communication technology.

The advent of high-speed Internet connectivity by means of cable modems and ADSL (Asynchronous Digital Subscriber Line) is an example of how a VPN becomes an enabler to facilitate the need for high-speed, individual remote access where none existed before. Existing remote access technologies are generally limited to 128K ISDN (Integrated Services Digital Network) or, more typically, 56K modem access. Given the inherent properties of the Internet and IPSec functioning at the network layer, the communication technology utilized to access the Internet only needs to be supported at the immediate connection point to establish an IP session with the ISP. Using the Internet as a backbone for encrypted communications allows for equal IP functionality with increased performance and security over conventional remote access technology.

Currently, cable modem and ADSL services are expanding from the home-user market into the business industry for remote office support. A typical remote office will have a small Frame Relay connection to the home office. Any Internet traffic from the remote office is usually forwarded to the home office's Internet connection, where access controls can be centrally managed and Internet connection costs are eliminated at the remote office. However, as the number of remote offices and the distances increase, so does the financial investment. Each Frame Relay connection, PVC (permanent virtual circuit), has costs associated with it. Committed Information Rate (CIR), port speed (e.g., 128K), and sometimes a connection fee add to the overall investment. A PVC is required for any connection; so, as remote offices demand direct communication to their peers, a PVC will need to be added to support this decentralized communication. Currently within the United States, the cost of Frame Relay is very low and typically outweighs the cost of an ISP and Internet connectivity. As the distance increases and moves beyond the United States, the costs can increase exponentially and will typically call for more than one telecommunications vendor. With VPN technology, a local connection to the Internet can be established. Adding connectivity to peers is accomplished by configuration modifications; this allows the customer to control communications without the inclusion of the carrier in the transformation.

The current stability of remote, tier three, and lower ISPs is an unknown variable. The arguable service associated with multiple and international ISP connectivity has become the Achilles' heel for VPN acceptance for business-critical and time-critical services. As the reach of tier one and tier two ISPs increases, they will be able to provide contiguous connectivity over the Internet to remote locations using an arsenal of available technologies.

Extranet Access

The single, most advantageous characteristic of VPNs is to provide protected and controlled communication with partnering organizations. Years ago, prior to VPN becoming a catchword, corporations were beginning to feel the need for dedicated Internet access. Dedicated access is becoming increasingly utilized for business purposes, whereas before it was viewed as a service for employees and research requirements.

The Internet provides the ultimate bridge between networks that was relatively nonexistent before VPN technology. Preceding VPNs, a corporation needing to access a partner's site was typically provided a Frame Relay connection to a common Frame Relay cloud where all the partners claimed access. Other options were ISDN and dial-on-demand routing. As this requirement grows, several limitations begin to surface. Security issues, partner support, controlling access, disallowing unwanted interchange between partners, and connectivity support for partners without supported access technologies all conspire to expose the huge advantages of VPNs over the Internet. Utilizing VPNs, an organization can maintain a high granularity of control over the connectivity per partner or per user on a partner network.

Internal Protection

As firewalls became more predominant as protection against the Internet, they were increasingly being utilized for internal segmentation of departmental entities. The need for protecting vital departments within an organization originally spawned this concept of using firewalls internally. As the number of departments increase, the management, complexity, and cost of the firewalls increase as well. Also, any attacker with access to the protected network can easily obtain sensitive information due to the fact that the firewall applies only perimeter security.

VLANs (virtual local area networks) with access control lists became a minimized replacement for conventional firewalls. However, the same security issue remained, in that the perimeter security was controlled and left the internal network open for attack.

As IPSec became accepted as a viable secure communication technology and applied in MAC environments, it also became the replacement for other protection technologies. Combined with strategically placed firewalls,

VPN over internal networks allows secure connectivity between hosts. IPSec encryption, authentication, and access control provide protection for data between departments and within a department.

Consideration for VPN Implementation

The benefits of VPN technology can be realized in varying degrees, depending on the application and the requirements it has been applied to. Considering the incredible growth in technology, the advantages will only increase. Nevertheless, the understandable concerns with performance, reliability, scalability, and implementation issues must be investigated.

System Requirements

The first step is determining the foreseeable amount of traffic and its patterns to ascertain the adjacent system requirements or augmentations. In the event that existing equipment is providing all or a portion of the service the VPN is replacing, the costs can be compared to discover initial savings in the framework of money, performance, or functionality.

Security Policy

It will be necessary to determine if the VPN technology and how it is planned to be implemented meet the current security policy. In case the security policy does not address the area of remote access, or in the event a policy or remote access does not exist, a policy must address the security requirements of the organization and its relationship with the service provided by VPN technology.

Application Performance

As previously discussed, performance is the primary reason VPN technology is not the solution for many organizations. It will be necessary to determine the speed at which an application can execute the essential processes. This is related to the type of data within the VPN. Live traffic or user sessions are incredibly sensitive to any latency in the communication. Pilot tests and load simulation should be considered strongly prior to large-scale VPN deployment or replacement of existing services and equipment.

Data replication or transient activity that is not associated with human or application time sensitivity is a candidate for VPN connectivity. The application's resistance to latency must be measured to determine the minimum requirements for the VPN. This is not to convey that VPNs are only good for replication traffic and cannot support user applications. It is necessary to determine the application needs and verify the requirements to properly gauge the performance provisioning of the VPN. The performance "window" will allow the proper selection of equipment to meet the needs of the proposed solution; otherwise, the equipment and application may present poor results compared to the expected or planned results. Or, more importantly, the acquired equipment is under-worked or does not scale in the direction needed for a particular organization's growth path. Each of these results in poor investment realization and makes it much more difficult to persuade management to use VPN again.

Training

User and administrator training is an important part of the implementation process. It is necessary to evaluate a vendor's product from the point of the users, as well as evaluating the other attributes of the product. In the event that user experience is poor, it will reach management and ultimately weigh heavily on the administrators and security practitioners. It is necessary to understand the user intervention that is required in the every-day process of application use. Comprehending the user knowledge requirements will allow for the creation of a training curriculum that best represents what the users are required to accomplish to operate the VPN as per the security policy.

Future of IPSec VPNs

Like it or not, VPN is here to stay. IP version 6 (IPv6) has the IPSec entrenched in its very foundation; and as the Internet grows, Ipv6 will become more prevalent. The current technological direction of typical networks will become the next goals for IPSec; specifically, Quality of Service (QoS). ATM was practically invented to accommodate the vast array of communication technologies at high speeds; but to do it efficiently, it must control who gets in and out of the network.

Ethernet Type of Service (ToS) (802.1p) allows for three bits of data in the frame to be used to add ToS information and then be mapped into ATM cells. IP version 4, as currently applied, has support for a ToS field in the IP Header similar to Ethernet 802.1p; it provides three bits for extended information. Currently, techniques are being applied to map QoS information from one medium to another. This is very exciting for service organizations that will be able sell end-to-end QoS. As the IPSec standard grows and current TCP/IP applications and networks begin to support the existing IP ToS field, IPSec will quickly conform to the requirements.

The IETF and other participants, in the form of RFCs, are continually addressing the issues that currently exist with IPSec. Packet sizes are typically increased due to the added headers and sometimes trailer information associated with IPSec. The result is an increased possibility of packet fragmentation. IPSec addresses fragmentation and packet loss; the overhead of these processes constitutes the largest concern.

IPSec can only be applied to the TCP/IP protocol. Therefore, multi-protocol networks and environments that employ IPX/SPX, NetBEUI, and others will not take direct advantage of the IPSec VPN. To allow non-TCP/IP protocols to communicate over an IPSec VPN, an IP gateway must be implemented to encapsulate the original protocol into an IP packet and then be forwarded to the IPSec gateway. IP gateways have been in use for some time and are proven technology. For several organizations that cannot eliminate non-TCP/IP protocols and wish to implement IPSec as the VPN of choice, a protocol gateway is imminent.

As is obvious, performance is crucial to IPSec VPN capabilities and cost. As encryption algorithms become increasingly sophisticated and hardware support for those algorithms becomes readily available, this current limitation will be surpassed.

Another perceived limitation of IPSec is the export and import restrictions of encryption. There are countries that the United States places restrictions on to hinder the ability of those countries to encrypt possibly harmful information into the United States. In 1996, the International Traffic in Arms Regulation (ITAR) governing the export of cryptography was reconditioned. Responsibility for cryptography exports was transferred to the Department of Commerce from the Department of State. However, the Department of Justice is now part of the export review process. In addition, the National Security Agency (NSA) remains the final arbiter of whether to grant encryption products export licenses.

The NSA staff is assigned to the Commerce Department and many other federal agencies that deal with encryption policy and standards. This includes the State Department, Justice Department, National Institute for Standards and Technology (NIST), and the Federal Communications Commission. As one can imagine, the laws governing the export of encryption are complicated and are under constant revision. Several countries are completely denied access to encrypted communications to the United States; other countries have limitations due to government relationships and political posture. The current list of (as of this writing) embargoed countries include:

- Syria
- Iran
- Iraq
- North Korea
- Libya
- Cuba
- Sudan
- Serbia

As one reads the list of countries, it is easy to see why the United States is reluctant to allow encrypted communications with these countries. Past wars, conflict of interests, and terrorism are the primary ingredients to become exiled by the United States.

Similar rosters exist for other countries that have the United States listed as "unfriendly," due to their perception of communication with the United States.

As one can certainly see, the concept of encryption export and import laws is vague, complex, and constantly in litigation. In the event a VPN is required for international communication, it will be necessary to obtain the latest information available to properly implement the communication as per the current laws.

Conclusion

VPN technology, based on IPSec, will become more prevalent in our every-day existence. The technology is in its infancy; the standards and support for them are growing every day. Security engineers will see an interesting change in how security is implemented and maintained on a daily basis. It will generate new types of policies and firewall solutions — router support for VPN will skyrocket.

This technology will finally confront encryption export and import laws, forcing the hand of many countries. Currently, there are several issues with export and import restrictions that affect how organizations deploy VPN technology. As VPNs become more prevalent in international communications, governments will be forced to expedite the process. With organizations sharing information, services, and product, the global economy will force computer security to become the primary focus for many companies.

For VPNs, latency is the center for concern and, once hardware solutions and algorithms collaborate to enhance overall system performance, the technology will become truly accepted. Once this point is reached, every packet on every network will be encrypted. Browsers, e-mail clients, and the like will have VPN software embedded, and only authenticated communications will be allowed. Clear Internet traffic will be material for campfire stories. It is a good time to be in security.

33

Firewalls: An Effective Solution for Internet Security

E. Eugene Schultz, Ph.D., CISSP

The Internet has presented a new, complex set of challenges that even the most sophisticated technical experts have not been able to solve adequately. Achieving adequate security is one of the foremost of these challenges. The major security threats that the Internet community faces are described in this chapter. It also explains how firewalls — potentially one of the most effective solutions for Internet security — can address these threats, and it presents some practical advice for obtaining the maximum advantages of using firewalls.

Internet Security Threats

The vastness and openness that characterizes the Internet presents an extremely challenging problem — security. Although many claims about the number and cost of Internet-related intrusions are available, valid, credible statistics about the magnitude of this problem will not be available until scientific research is conducted. Exacerbating this dilemma is that most corporations that experience intrusions from the Internet and other sources do not want to make these incidents known for fear of public relations damage and, worse yet, many organizations fail to even detect most intrusions. Sources, such as Carnegie Mellon University's Computer Emergency Response Team, however, suggest that the number of Internet-related intrusions each year is very high and that the number of intrusions reported to CERT (which is one of dozens of incident response teams) is only the tip of the iceberg. No credible statistics concerning the total amount of financial loss resulting from security-related intrusions are available; but judging from the amount of money corporations and government agencies are spending to implement Internet and other security controls, the cost must be extremely high.

Many types of Internet security threats exist. One of the most serious methods is IP spoofing. In this type of attack, a perpetrator fabricates packets that bear the address of origination of a client host and sends these packets to the server for this client. The server acknowledges receiving these packets by returning packets with a certain sequence number. If the attacker can guess this packet sequence number and incorporate it into another set of fabricated packets that is then sent back to the server, the server can be tricked into setting up a connection with a fraudulent client. The intruder can subsequently use attack methods, such as use of trusted host relationships, to intrude into the server machine.

A similar threat is domain name service (DNS) spoofing. In this type of attack, an intruder subverts a host within a network and sets up this machine to function as an apparently legitimate name server. The host then provides bogus data about host identities and certain network services, enabling the intruder to break into other hosts within the network.

Session hijacking is another Internet security threat. The major tasks for the attacker who wants to hijack an ongoing session between remote hosts are locating an existing connection between two hosts and fabricating packets that bear the address of the host from which the connection has originated. By sending these packets to the destination host, the originating host's connection is dropped, and the attacker picks up the connection.

Another Internet security threat is network snooping, in which attackers install programs that copy packets traversing network segments. The attackers periodically inspect files that contain the data from the captured packets to discover critical log-on information, particularly user IDs and passwords for remote systems. Attackers subsequently connect to the systems for which they possess the correct log-on information and log on with no trouble. Attackers targeting networks operated by Internet service providers (ISPs) have made this problem especially serious, because so much information travels these networks. These attacks demonstrate just how vulnerable network infrastructures are; successfully attacking networks at key points, where router, firewalls, and server machines are located, is generally the most efficient way to gain information allowing unauthorized access to multitudes of host machines within a network.

A significant proportion of attacks exploit security exposures in programs that provide important network services. Examples of these programs include sendmail, Network File System (NFS), and Network Information Service (NIS). These exposures allow intruders to gain access to remote hosts and to manipulate services supported by these hosts or even to obtain superuser access. Of increasing concern is the susceptibility of World Wide Web services and the hosts that house these services to successful attack. The ability of intruders to exploit vulnerabilities in the HTTP and in Java, a programming language used to write WWW applications, seems to be growing at an alarming rate.

Until a short time ago, most intruders attempted to cover up indications of their activity, often by installing programs that selectively eliminated data from system logs. These also avoided causing system crashes or causing massive slowdowns or disruption. However, a significant proportion of the perpetrator community has apparently shifted its strategy by increasingly perpetrating denial-of-service attacks. For example, many types of hosts crash or perform a core dump when they are sent a packet internet groper or ping packet that exceeds a specified size limit or when they are flooded with synchronize (SYN) packets that initiate host-to-host connections. (Packet internet groper, or ping, is a service used to determine whether a host on a network is up and running.) These denial-of-service attacks make up an increasing proportion of observed Internet attacks. They represent a particularly serious threat because many organizations require continuity of computing and networking operations to maintain their business operations.

Not to be overlooked is another type of security threat called social engineering. Social engineering is fabricating a story to trick users, system administrators, or help desk personnel into providing information required to access systems. Intruders usually solicit passwords for user accounts, but information about the network infrastructure and the identity of individual hosts can also be the target of social engineering attacks.

Internet Security Controls

As previously mentioned, Internet security threats pose a challenge because of their diversity and severity. An added complication is an abundance of potential solutions.

Encryption

Encryption is a process of using an algorithm to transform cleartext information into text that cannot be read without the proper key. Encryption protects information stored in host machines and transmitted over networks. It is also useful in authenticating users to hosts or networks. Although encryption is an effective solution, its usefulness is limited by the difficulty in managing encryption keys (i.e., of assigning keys to users and recovering keys if they are lost or forgotten), laws limiting the export and use of encryption, and the lack of adherence to encryption standards by many vendors.

One-Time Passwords

Using one-time passwords is another way in which to challenge security threats. One-time passwords captured while in transit over networks become worthless because each password can only be used once. A captured password has already been used by the legitimate user who has initiated a remote log-on session by the time the captured password can be employed. Nevertheless, one-time passwords address only a relatively small proportion of the total range of Internet security threats. They do not, for example, protect against IP spoofing or exploitation of vulnerabilities in programs.

Installing fixes for vulnerabilities in all hosts within an Internet-capable network does not provide an entirely suitable solution because of the cost of labor, and, over the last few years, vulnerabilities have surfaced at a rate far faster than that at which fixes have become available.

Firewalls

Although no single Internet security control measure is perfect, the firewall has, in many respects, proved more useful overall than most other controls. Simply, a firewall is a security barrier between two networks that screens traffic coming in and out of the gate of one network to accept or reject connections and service requests according to a set of rules. If configured properly, it addresses a large number of threats that originate from outside a network without introducing any significant security liabilities. Because most organizations are unable to install every patch that CERT advisories describe, these organizations can nevertheless protect hosts within their networks against external attacks that exploit vulnerabilities by installing a firewall that prevents users from outside the network from reaching the vulnerable programs in the first place. A more sophisticated firewall also controls how any connection between a host external to a network and an internal host occurs. Moreover, an effective firewall hides information, such as names and addresses of hosts within the network, as well as the topology of the network which it is employed to protect.

Firewalls can defend against attacks on hosts (including spoofing attacks), application protocols, and applications. In addition, firewalls provide a central method for administering security on a network and for logging incoming and outgoing traffic to allow for accountability of user actions and for triggering incident response activity if unauthorized activity occurs.

Firewalls are typically placed at gateways to networks to create a security perimeter, as shown in Exhibit 33.1, primarily to protect an internal network from threats originating from an external one (particularly from the Internet). This scheme is successful to the degree that the security perimeter is not accessible through unprotected avenues of access. The firewall acts as a choke component for security purposes. Exhibit 33.1 displays routers that are located in front and in back of the firewall. The first router (shown above the firewall) is an external one used initially to route incoming traffic, to direct outgoing traffic to external networks, and to broadcast information that enables other network routers (as well as the router on the other side of the firewall) to know how to reach the host network. The other internal router (shown below the firewall) sends incoming packets to their destination within the internal network, directs outgoing packets to the external

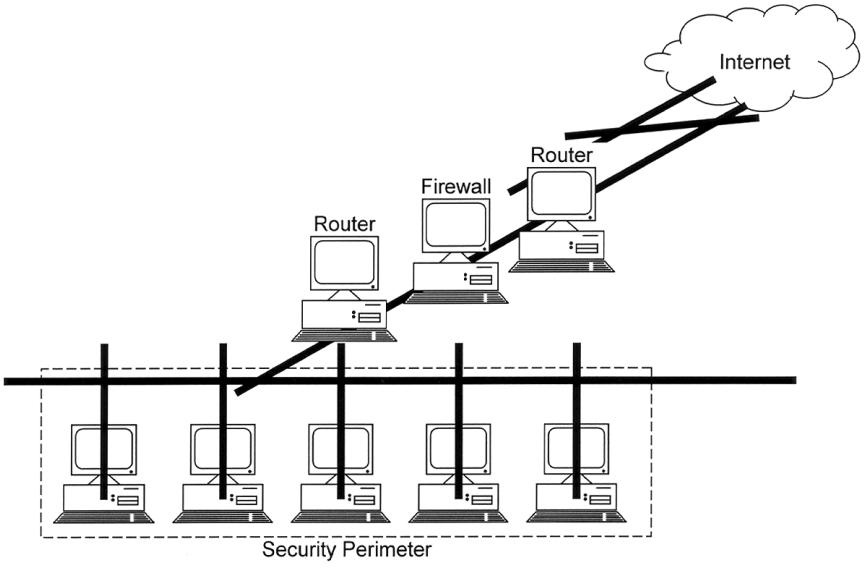


EXHIBIT 33.1 A typical gate-based firewall architecture.

router, and broadcasts information on how to reach the internal network and the external router. This belt-and-suspenders configuration further boosts security by preventing the broadcast of information about the internal network outside the network the firewall protects. An attacker finding this information can learn IP addresses, subnets, servers, and other information that is useful in perpetrating attacks against the network. Hiding information about the internal network is much more difficult if the gate has only one router.

Another way in which firewalls are deployed (although less frequently) is within an internal network — at the entrance to a subnet within a network — rather than at the gateway to the entire network. The purpose of this configuration (shown in Exhibit 33.2) is to segregate a subnetwork (a screened subnet) from the internal network at large, a wise strategy if the subnet has tighter security requirements than the rest of the security perimeter. This type of deployment more carefully controls access to data and services within a subnet than is otherwise allowed within the network. The gate-based firewall, for example, may allow File Transfer Protocol (FTP) access to an internal network from external sources. However, if a subnet contains hosts that store information, such as lease bid data or salary data, then allowing FTP access to this subnet is less advisable. Setting up the subnet as a screened subnet may provide suitable security control; that is, the internal firewall that provides security screening for the subnet is configured to deny all FTP access, regardless of whether the access requests originated from outside or inside the network.

Simply having a firewall, no matter how it is designed and implemented, does not necessarily protect against externally originated security threats. The benefits of firewalls depend to a large degree on the type used and how it is deployed and maintained.

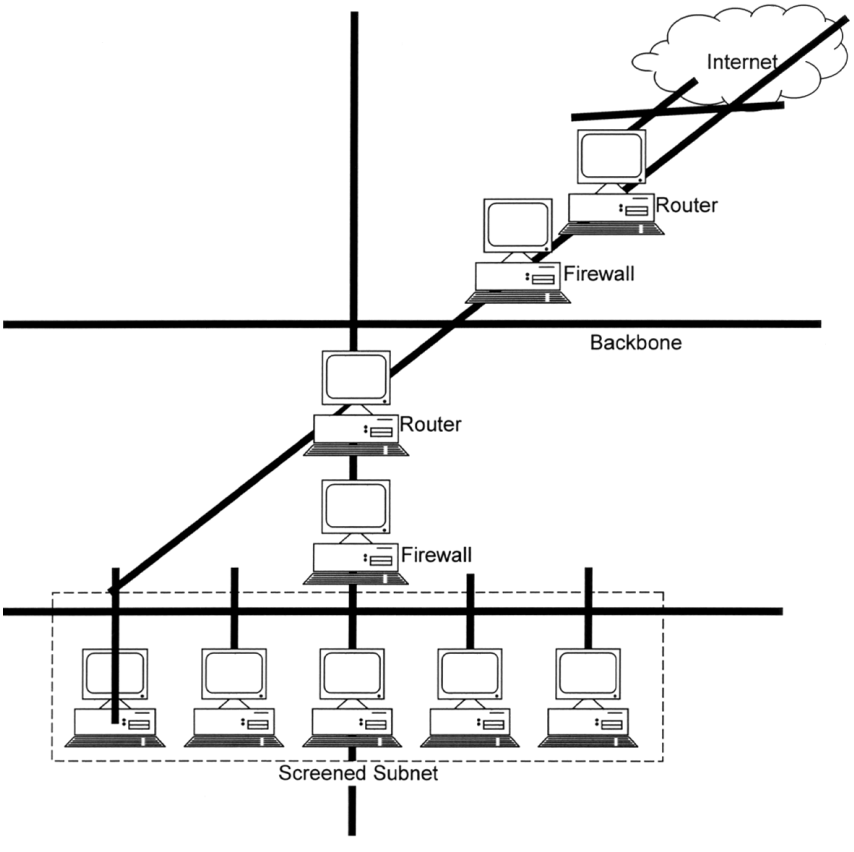


EXHIBIT 33.2 A screened subnet

Using Firewalls Effectively

To ensure that firewalls perform their intended function, it is important to choose the appropriate firewall and to implement it correctly. Establishing a firewall policy is also a critical step in securing a system, as is regular maintenance of the entire security structure.

Choosing the Right Firewall

Each type of firewall offers its own set of advantages and disadvantages. Combined with the vast array of vendor firewall products and the possibility of custom-building a firewall, this task can be potentially overwhelming. Establishing a set of criteria for selecting an appropriate firewall is an effective aid in narrowing down the choices.

One of the most important considerations is the amount and type of security needed. For some organizations with low to moderate security needs, installing a packet-filtering firewall that blocks only the most dangerous incoming service requests often provides the most satisfactory solution because the cost and effort are not likely to be great. For other organizations, such as banks and insurance corporations, packet-filtering firewalls do not generally provide the granularity and control against unauthorized actions usually needed for connecting customers to services that reside within a financial or insurance corporation's network.

Additional factors, such as the reputation of the vendor, the arrangements for vendor support, the verifiability of the firewall's code (i.e., to confirm that the firewall does what the vendor claims it does), the support for strong authentication, the ease of administration, the ability of the firewall to withstand direct attacks, and the quality and extent of logging and alarming capabilities, should also be strong considerations in choosing a firewall.

The Importance of a Firewall Policy

The discussion to this point has focused on high-level technical considerations. Although these considerations are extremely important, too often security professionals overlook other considerations that, if neglected, can render firewalls ineffective. The most important consideration in effectively using firewalls is developing a firewall policy.

A firewall policy is a statement of how a firewall should work — the rules by which incoming and outgoing traffic should be allowed or rejected. A firewall policy, therefore, is a type of security requirements document for a firewall. As security needs change, firewall policies must change accordingly. Failing to create and update a firewall policy for each firewall almost inevitably results in gaps between expectations and the actual function of the firewall, resulting in uncontrolled security exposures in firewall functionality. For example, security administrators may think that all incoming HTTP requests are blocked, but the firewall may actually allow HTTP requests from certain IP addresses, leaving an unrecognized avenue of attack.

An effective firewall policy should provide the basis for firewall implementation and configuration; needed changes in the way the firewall works should always be preceded by changes in the firewall policy. An accurate, up-to-date firewall policy should also serve as the basis for evaluating and testing a firewall.

Security Maintenance

Many organizations that employ firewalls feel a false sense of security once the firewalls are in place. Properly designing and implementing firewalls can be difficult, costly, and time consuming. It is critical to remember, however, that firewall design and implementation are simply the beginning points of having a firewall. Firewalls that are improperly maintained soon lose their value as security control tools.

One of the most important facets of firewall maintenance is updating the security policy and rules by which each firewall operates. Firewall functionality invariably must change as new services and applications are introduced in (or sometimes removed from) a network. Undertaking the task of daily inspections of firewall logs to discover attempted and possibly successful attacks on both the firewall and the internal network that it protects should be an extremely high priority. Evaluating and testing the adequacy of firewalls for unexpected access avenues to the security perimeter and vulnerabilities that lead to unauthorized access to the firewall should also be a frequent, high-priority activity.

Firewall products have improved considerably over the past several years and are likely to continue to improve. Several vendor products, for example, are not network addressable, which makes breaking into these platforms by someone who does not have physical access to them virtually impossible. At the same time, however, recognizing the limitations of firewalls and ensuring that other appropriate Internet security controls are in place is becoming increasingly important because of such problems as third-party connections to organizations' networks that bypass gate-based security mechanisms altogether. Therefore, an Internet security strategy that includes firewalls in addition to host-based security mechanisms is invariably the most appropriate direction for achieving suitable levels of Internet security.

Conclusion

Internet connectivity can be extremely valuable to an organization, but it involves many security risks. A firewall is a key tool in an appropriate set of security control measures to protect Internet-capable networks. Firewalls can be placed at the gateway to a network to form a security perimeter around the networks that they protect or at the entrance to subnets to screen the subnets from the rest of the internal network.

Developing an accurate and complete firewall policy is the most important step in using firewalls effectively. This policy should be modified and updated as new applications are added within the internal network protected by the firewall and as new security threats emerge. Maintaining firewalls properly and regularly examining the log data that they provide are almost certainly the most neglected aspects of using firewalls. Yet, these activities are among the most important in ensuring that the defenses are adequate and that incidents are quickly detected and handled. Performing regular security evaluations and testing the firewall to identify any exploitable vulnerabilities or misconfigurations are also essential activities. Establishing a regular security procedure minimizes the possibility of system penetration by an attacker.

34

Internet Security: Securing the Perimeter

Douglas G. Conorich, CISSP

The Internet has become the fastest growing tool organizations have ever had that can help them become more productive. In spite of its usefulness, there have been many debates as to whether the Internet can be used, in light of the many security issues. Today, more than ever before, computing systems are vulnerable to unauthorized access. Given the right combination of motivation, expertise, resources, time, and social engineering, an intruder will be able to access any computer that is attached to the Internet.

The corporate community has, in part, created this problem for itself. The rapid growth of the Internet with all the utilities now available to Web surf, combined with the number of users who now have easy access through all the various Internet providers, make every desktop — including those in homes, schools, and libraries — a place where an intruder can launch an attack. Surfing the Internet began as a novelty. Users were seduced by the vast amounts of information they could find. In many cases, it has become addictive.

Much of the public concern with the Internet has focused on the inappropriate access to Web sites by children from their homes or schools. A business is concerned with the bottom line. How profitable a business is can be directly related to the productivity of its employees. Inappropriate use of the Internet in the business world can decrease that productivity in many ways. The network bandwidth — how much data can flow across a network segment at any time — is costly to increase because of the time involved and the technology issues. Inappropriate use of the Internet can slow the flow of data and create the network approximation of a log jam.

There are also potential legal and public relations implications of inappropriate employee usage. One such issue is the increasing prevalence of "sin surfing" — browsing the pornographic Web sites. One company reported that 37 percent of its Internet bandwidth was taken up by "sin surfing." Lawsuits can be generated and, more importantly, the organization's image can be damaged by employees using the Internet to distribute inappropriate materials. To legally curtail the inappropriate use of the Internet, an organization must have a policy that defines what is acceptable, what is not, and what can happen if an employee is caught.

As part of the price of doing business, companies continue to span the bridge between the Internet and their own intranets with mission-critical applications. This makes them more vulnerable to new and unanticipated security threats. Such exposures can place organizations at risk at every level — down to the very credibility upon which they build their reputations.

Making the Internet safe and secure for business requires careful management by the organization. Companies will have to use existing and new, emerging technologies, security policies tailored to the business needs of the organization, and training of the employees in order to accomplish this goal. IBM has defined four phases of Internet adoption by companies as they do business on the Internet: access, presence, integration, and E-business. Each of these phases has risks involved.

1. *Access.* In this first phase of adoption, a company has just begun to explore the Internet and learn about its potential benefits. A few employees are using modems connected to their desktop PCs, to dial into either a local Internet service provider or a national service such as America Online. In this phase, the company is using the Internet as a resource for getting information only; all requests for access are in the outbound direction, and all information flow is in the inbound direction. Exchanging electronic mail and browsing the Web make up the majority of activities in this phase.

2. *Presence.* In this phase, the company has begun to make use of the Internet not only as a resource for getting information, but also as a means of providing information to others. Direct connection of the company's internal network means that all employees now have the ability to access the Internet (although this may be restricted by policy), allowing them to use it as an information resource, and also enabling processes such as customer support via e-mail. The creation of a Web server, either by the company's own staff or through a content hosting service, allows the company to provide static information such as product catalogs and data sheets, company background information, software updates, etc. to its customers and prospects.
3. *Integration.* In this phase, the company has begun to integrate the Internet into its day-to-day business processes by connecting its Web server directly (through a firewall or other protection system) to its back-office systems. In the previous phase, updates to the Web server's data were made manually, via tape or other means. In this phase, the Web server can obtain information on demand, as users request it. To use banking as an example, this phase enables the bank's customers to obtain their account balances, find out when checks cleared, and other information retrieval functions.
4. *E-business.* In the final phase, the company has enabled bi-directional access requests and information flow. This means that not only can customers on the Internet retrieve information from the company's back-office systems, but they can also add to or change information stored on those systems. At this stage, the company is conducting business electronically; customers can place orders, transfer money (via credit cards or other means), check on shipments, etc. business partners can update inventories, make notes in customer records, etc. In short, the entire company has become accessible via the Internet.

While companies may follow this road to the end, as described by IBM, they are most likely somewhere on it, either in one of the phases or in transition between them.

Internet Protocols

Communication between two people is made possible by their mutual agreement to a common mode of transferring ideas from one person to the other. Each person must know exactly how to communicate with the other if this is to be successful. The communication can be in the form of a verbal or written language, such as English, Spanish, or German. It can also take the form of physical gestures such as sign language. It can even be done through pictures or music. Regardless of the form of the communication, it is paramount that the meaning of an element, say a word, has the same meaning to both parties involved. The medium used for communication is also important. Both parties must have access to the same communication medium. One cannot talk to someone else via telephone if only one person has a telephone.

With computers, communications over networks is made possible by what are known as protocols. A protocol is a well-defined message format. The message format defines what each position in the message means. One possible message format could define the first 4 bits as the version number, the next 4 bits as the length of the header, and then 8 bits for the service being used. As long as both computers agree on this format, communication can take place.

Network communications use more than one protocol. Sets of protocols used together are known as protocol suites or layered protocols. One well-known protocol suite is the Transport Control Protocol/ Internet Protocol (TCP/IP) suite. It is based on the International Standards Organization (ISO) Open Systems Interconnection (OSI) Reference Model (see Exhibit 34.1).

The ISO Reference Model is divided into seven layers:

1. The Physical Layer is the lowest layer in the protocol stack. It consists of the "physical" connection. This may be copper wire or fiber-optic cables and the associated connection hardware. The sole responsibility of the Physical Layer is to transfer the bits from one location to another.
2. The second layer is the Data-Link Layer. It provides for the reliable delivery of data across the physical link. The Data-Link Layer creates a checksum of the message that can be used by the receiving host to ensure that the entire message was received.
3. The Network Layer manages the connections across the network for the upper four layers and isolates them from the details of addressing and delivery of data.
4. The Transport Layer provides the end-to-end error detection and correction function between communicating applications.
5. The Session Layer manages the sessions between communicating applications.

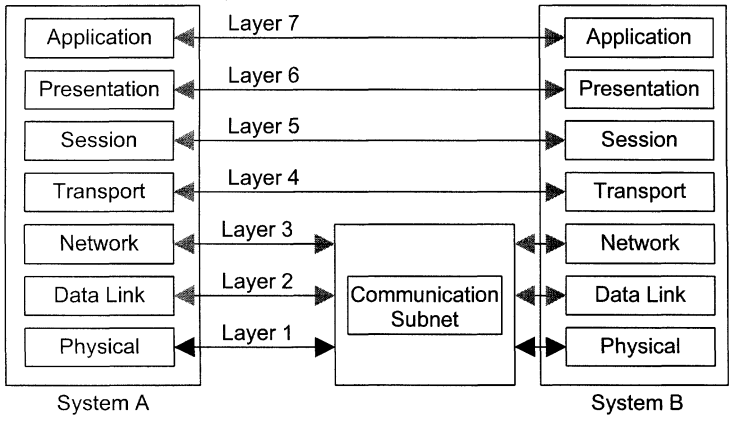


EXHIBIT 34.1 The ISO model.

6. The Preparation Layer standardizes the data presentation to the application level.
7. The Application Layer consists of application programs that communicate across the network. This is the layer with which most users interact.

Network devices can provide different levels of security, depending on how far up the stack they can read. Repeaters are used to connect two Ethernet segments. The repeater simply copies the electrical transmission and sends it on to the next segment of the network. Because the repeater only reads up through the Data-Link Layer, no security can be added by its use.

The bridge is a computer that is used to connect two or more networks. The bridge differs from the repeater in that it can store and forward entire packets, instead of just repeating electrical signals. Because it reads up through the Network Layer of the packet, the bridge can add some security. It could allow the transfer of only packets with local addresses. A bridge uses physical addresses — not IP addresses. The physical address, also know as the Ethernet address, is the actual address of the Ethernet hardware. It is a 48-bit number.

Routers and gateways are computers that determine which of the many possible paths a packet will take to get to the destination device. These devices read up through the Transport Layer and can read IP addresses, including port numbers. They can be programmed to allow, disallow, and reroute IP datagrams determined by the IP address of the packet.

As previously mentioned, TCP/IP is based on the ISO model, but it groups the seven layers of the ISO model into four layers, as displayed in Exhibit 34.2.

The Network Access Layer is the lowest layer of the TCP/IP protocol stack. It provides the means of delivery and has to understand how the network transmits data from one IP address to another. The Network Access Layer basically provides the functionality of the first three layers of the ISO model.

Application Layer consists of applications and processes that use the network.
Host-to-Host Transport Layer provides end-to-end data delivery service.
Internet Layer Defines the datagram and handles the routing of data.
Network Access Layer consists of routines for accessing physical networks.

EXHIBIT 34.2 The TCP/IP protocol architecture.

TCP/IP provides a scheme of IP addressing that uniquely defines every host connected to the Internet. The Network Access Layer provides the functions that encapsulate the datagrams and maps the IP addresses to the physical addresses used by the network.

The Internet Layer has at its core the Internet Protocol (RFC 791). IP provides the basic building blocks of the Internet. It provides:

- Datagram definition scheme
- Internet addressing scheme
- Means of moving data between the Network Access Layer and the Host-to-Host Layer
- Means for datagrams to be routed to remote hosts
- Function of breaking apart and reassembling packets for transmission

IP is a connectionless protocol. This means that it relies on other protocols within the TCP/IP stack to provide the connection-oriented services. The connection-oriented services (i.e., TCP) take care of the handshake — the exchange of control information. The IP Layer contains the Internet Control Message Protocol (ICMP).

The Host-to-Host Transport Layer houses two protocols: the Transport Control Protocol (TCP) and the User Datagram Protocol (UDP). Its primary function is to deliver messages between the Application Layer and the Internet Layer. TCP is a reliable protocol. This means that it guarantees that the message will arrive as sent. It contains error detection and correction features. UDP does not have these features and is, therefore, unreliable. For shorter messages, where it is easier to resend the message than worry about the overhead involved with TCP, UDP is used.

The Application Layer contains the various services that users will use to send data. The Application Layer contains such user programs as the Network Terminal Protocol (Telnet), File Transfer Protocol (FTP), and Simple Mail Transport Protocol (SMTP). It also contains protocols not directly used by users, but required for system use (e.g., Domain Name Service (DNS), Routing Information Protocol (RIP), and Network File System (NFS)).

Attacks

As previously noted, TCP is a reliable messaging protocol. This means that TCP is a connection-oriented protocol. TCP uses what is known as a "three-way handshake." A handshake is simply the exchange of control information between the two computers. This information enables the computers to determine which packets go where and ensure that all the information in the message has been received.

When a connection is desired between two systems, Host A and Host B, using TCP/IP, a three-way handshake must occur. The initiating host, Host A (the client), sends the receiving host, Host B (the server), a message with the SYN (synchronize sequence number) bit set. The SYN contains information needed by Host B to set up the connection. This message contains the IP address of the both Host A and Host B and the port numbers they will talk on. The SYN tells Host B what sequence number the client will start with, seq = x. This number is important to keep all the data transmitted in the proper order and can be used to notify Host B that a piece of data is missing. The sequence number is found starting at bit 32 to 63 of the header.

When Host B receives the SYN, it sends the client an ACK (acknowledgment message). This message contains the sequence number that Host B will start with, SYN, seq = y, and the sequence number of Host A incremented, the ACK, x + 1. The acknowledgment number is bits 64 through 95 of the header.

The three-way handshake is completed when Host A receives the ACK from Host B and sends an ACK, y + 1, in return. Now data can flow back and forth between the two hosts. This connection is now known as a socket. A socket is usually identified as Host_A_IP:Port_Number, Host_B_IP:Port_Number.

There are two attacks that use this technology: SYN flood and sequence predictability.

SYN Flood Attack

The SYN flood attack uses a TCP connection request (SYN). The SYN is sent to the target computer with the source IP address in the packet "spoofed," or replaced with an address that is not in use on the Internet or that belongs to another computer. When the target computer receives the connection request, it allocates resources to handle and track the new connection. A SYN_RECEIVED state is stored in a buffer register awaiting the return response (ACK) from the initiating computer, which would complete the three-way handshake. It then sends out an SYN-ACK. If the response is sent to the "spoofed," nonexistent IP address,

there will never be a response. If the SYN-ACK is sent to a real computer, it checks to see if it has a SYN in the buffer to that IP address. Because it does not, it ignores the request. The target computer retransmits the SYN-ACK a number of times. After a finite amount of wait time, the original SYN request is purged from the buffer of the target computer. This condition is known as a half-open socket.

As an example, the default configuration for a Windows NT 3.5x or 4.0 computer is to retransmit the SYN-ACK five times, doubling the timeout value after each retransmission. The initial timeout value is 3 seconds, so retries are attempted at 3, 6, 12, 24, and 48 seconds. After the last retransmission, 96 seconds are allowed to pass before the computer gives up on receiving a response and deallocates the resources that were set aside earlier for the connection. The total elapsed time that resources are in use is 189 seconds.

An attacker will send many of these TCP SYNs to tie up as many resources as possible on the target computer. Because the buffer size for the storage of SYNs is a finite size, numerous attempts can cause a buffer overflow. The effect of tying up connection resources varies, depending on the TCP/IP stack and applications listening on the TCP port. For most stacks, there is a limit on the number of connections that can be in the half-open SYN_RECEIVED state. Once the limit is reached for a given TCP port, the target computer responds with a reset to all further connection requests until resources are freed. Using this method, an attacker can cause a denial-of-service on several ports.

Finding the source of a SYN flood attack can be very difficult. A network analyzer can be used to try to track down the problem, and it may be necessary to contact the Internet service provider for assistance in attempting to trace the source. Firewalls should be set up to reject packets from the external network with any IP address from the internal network.

Sequence Predictability

The ability to guess sequence numbers is very useful to intruders because they can create a short-lived connection to a host without having to see the reply packets. This ability, taken in combination with the fact that many hosts have trust relationships that use IP addresses as authentication; that packets are easily spoofed; and that individuals can mount denial of service attacks, means one can impersonate the trusted systems to break into such machines without using source routing.

If an intruder wants to spoof a connection between two computers so that the connection seems as if it is coming from computer B to computer A, using your computer C, it works like this:

1. First, the intruder uses computer C to mount a SYN Flood attack on the ports on computer B where the impersonating will take place.
2. Then, computer C sends a normal SYN to a port on computer A.
3. Computer A returns a SYN-ACK to computer C containing computer A's current Initial Sequence Number (ISN).
4. Computer A internally increments the ISN. This incrementation is done differently in different operating systems (OSs). Operating systems such as BSD, HPUX, Irix, SunOS (not Solaris), and others usually increment by $FA00 for each connection and double each second.

   With this information, the intruder can now guess the ISN that computer A will pick for the next connection. Now comes the spoof.
5. Computer C sends a SYN to computer A using the source IP spoofed as computer B.
6. Computer A sends a SYN-ACK back to computer B, containing the ISN. The intruder on computer C does not see this, but the intruder has guessed the ISN.
7. At this point, computer B would respond to computer A with an RST. This occurs because computer B does not have a SYN_RECEIVED from computer A. Since the intruder used a SYN Flood attack on computer B, it will not respond.
8. The intruder on computer C sends an ACK to computer A, using the source IP spoofed as computer B, containing the guessed ISN+1.

   If the guess was correct, computer A now thinks there has been a successful three-way handshake and the TCP connection between computer A and computer B is fully set up. Now the spoof is complete. The intruder on computer C can do anything, but blindly.
9. Computer C sends `echo + + >>/.rhosts` to port 514 on computer A.
10. If root on computer A had computer B in its /.rhosts file, the intruder has root.
11. Computer C now sends a FIN to computer A.
12. Computer C could be brutal and send an RST to computer A just to clean up things.
13. Computer C could also send an RST to the synflooded port on B, leaving no traces.

To prevent such attacks, one should NEVER trust anything from the Internet. Routers and firewalls should filter out any packets that are coming from the external (sometimes known as the red) side of the firewall that has an IP address of a computer on the internal (sometimes known as the blue) side. This only stops Internet trust exploits; it will not stop spoofs that build on intranet trusts. Companies should avoid using rhosts files wherever possible.

ICMP

A major component of the TCP/IP Internet Layer is the Internet Control Message Protocol (ICMP). ICMP is used for flow control, detecting unreachable destinations, redirection routes, and checking remote hosts. Most users are interested in the last of these functions. Checking a remote host is accomplished by sending an ICMP Echo Message. The PING command is used to send these messages.

When a system receives one of these ICMP Echo Messages, it places the message in a buffer and then retransmits the message from the buffer back to the source. Due to the buffer size, the ICMP Echo Message size cannot exceed 64K. UNIX hosts, by default, will send an ICMP Echo Message that is 64 bytes long. They will not allow a message of over 64K. With the advent of Microsoft Windows NT, longer messages can be sent. The Windows NT hosts do not place an upper limit on these messages. Intruders have been sending messages of 1 MB and larger. When these messages are received, they cause a buffer overflow on the target host. Different operating systems will react differently to this buffer overflow. The reactions range from rebooting to a total system crash.

Firewalls

The first line of defense between the Internet and an intranet should be a firewall. A firewall is a multi-homed host that is placed in the Internet route, such that it stops and can make decisions about each packet that wants to get through. A firewall performs a different function from a router. A router can be used to filter out certain packets that meet a specific criteria (e.g., an IP address). A router processes the packets up through the IP Layer. A firewall stops all packets. All packets are processed up through the Application Layer. Routers cannot perform all the functions of a firewall. A firewall should meet, at least, the following criteria:

- For an internal or external host to connect to the other network, it must log in on the firewall host.
- All electronic mail is sent to the firewall, which in turn distributes it.
- Firewalls should not mount file systems via NFS, nor should any of its file systems be mounted.
- Firewalls should not run NIS (Network Information Systems).
- Only required users should have accounts on the firewall host.
- The firewall host should not be trusted, nor trust any other host.
- The firewall host is the only machine with anonymous FTP.
- Only the minimum service should be enabled on the firewall in the file inetd.conf.
- All system logs on the firewall should log to a separate host.
- Compilers and loaders should be deleted on the firewall.
- System directories permissions on the firewall host should be 711 or 511.

The DMZ

Most companies today are finding that it is imperative to have an Internet presence. This Internet presence takes on the form of anonymous FTP sites and a World Wide Web (WWW) site. In addition to these, companies are setting up hosts to act as a proxy server for Internet mail and a Domain Name Server (DNS). The host that sponsors these functions cannot be on the inside of the firewall. Therefore, companies are creating what has become known as the demilitarized zone (DMZ) or perimeter network, a segment between the router that connects to the Internet and the firewall.

Proxy Servers

A proxy host is a dual-homed host that is dedicated to a particular service or set of services, such as mail. All external requests to that service directed toward the internal network are routed to the proxy. The proxy host

then evaluates the request and either passes the request on to the internal service server or discards it. The reverse is also true. Internal requests are passed to the proxy from the service server before they are passed on to the Internet.

One of the functions of the proxy hosts is to protect the company from advertising its internal network scheme. Most proxy software packages contain network address translation (NAT). Take, for example, a mail server. The mail from Albert_Smith@starwars.abc.com would be translated to smith@proxy.abc.com as it went out to the Internet. Mail sent to smith@proxy.abc.com would be sent to the mail proxy. Here it would be readdressed to Albert_Smith@starwars.abc.com and sent to the internal mail server for final delivery.

Testing the Perimeter

A company cannot use the Internet without taking risks. It is important to recognize these risks and it is important not to exaggerate them. One cannot cross the street without taking a risk. But by recognizing the dangers, and taking the proper precautions (such as looking both ways before stepping off the curb), millions of people cross the street safely every day.

The Internet and intranets are in a state of constant change — new protocols, new applications, and new technologies — and a company's security practices must be able to adapt to these changes. To adapt, the security process should be viewed as forming a circle. The first step is to assess the current state of security within one's intranet and along the perimeter. Once one understands where one is, then one can deploy a security solution. If you do not monitor that solution by enabling some detection and devising a response plan, the solution is useless. It would be like putting an alarm on a car, but never checking it when the alarm goes off. As the solution is monitored and tested, there will be further weaknesses — which brings us back to the assessment stage and the process is repeated. Those new weaknesses are then learned about and dealt with, and a third round begins. This continuous improvement ensures that corporate assets are always protected.

As part of this process, a company must perform some sort of vulnerability checking on a regular basis. This can be done by the company, or it may choose to have an independent group do the testing. The company's security policy should state how the firewall and the other hosts in the DMZ are to be configured. These configurations need to be validated and then periodically checked to ensure that they have not changed. The vulnerability test may find additional weaknesses with the configurations and then the policy needs to be changed.

Security is achieved through the combination of technology and policy. The technology must be kept up-to-date and the policy must outline the procedures. An important part of a good security policy is to ensure that there are as few information leaks as possible.

One source of information can be DNS records. There are two basic DNS services: lookups and zone transfers. Lookup activities are used to resolve IP addresses into host names or to do the reverse. A zone transfer happens when one DNS server (a secondary server) asks another DNS server (the primary server) for all the information that it knows about a particular part of the DNS tree (a zone). These zone transfers only happen between DNS servers that are supposed to be providing the same information. Users can also request a zone transfer.

A zone transfer is accomplished using the nslookup command in interactive mode. The zone transfer can be used to check for information leaks. This procedure can show hosts, their IP addresses, and operating systems. A good security policy is to disallow zone transfers on external DNS servers. This information can be used by an intruder to attack or spoof other hosts. If this is not operationally possible, as a general rule, DNS servers outside of the firewall (on the red side) should not list hosts within the firewall (on the blue side). Listing internal hosts only helps intruders gain network mapping information and gives them an idea of the internal IP addressing scheme.

In addition to trying to do a zone transfer, the DNS records should be checked to ensure that they are correct and that they have not changed. Domain Information Gofer (DIG) is a flexible command-line tool that is used to gather information from the Domain Name System servers.

The ping command, as previously mentioned, has the ability to determine the status of a remote host using the ICMP Echo Message. If a host is running and is reachable by the message, the PING program will return an "alive" message. If the host is not reachable and the host name can be resolved by DNS, the program returns a "host not responding" message; otherwise, an "unknown host" message is obtained. An intruder can use the PING program to set up a "war dialer." This is a program that systematically goes through the IP addresses one after another, looking for "alive" or "not responding" hosts. To prevent intruders from mapping internal

networks, the firewall should screen out ICMP messages. This can be done by not allowing ICMP messages to go through to the internal network or go out from the internal network. The former is the preferred method. This would keep intruders from using ICMP attacks, such as the Ping 'O Death or Loki tunneling.

The traceroute program is another useful tool one can use to test the corporate perimeter. Because the Internet is a large aggregate of networks and hardware connected by various gateways, traceroute is used to check the "time-to-live" (ttl) parameter and routes. traceroute sends a series of three UDP packets with an ICMP packet incorporated during its check. The ttl of each packet is similar. As the ttl expires, it sends the ICMP packet back to the originating host with the IP address of the host where it expired. Each successive broadcast uses a longer ttl. By continuing to send longer ttls, traceroute pieces together the successive jumps. Checking the various jumps not only shows the routes, but it can show possible problems that may give an intruder information or leads. This information might show a place where an intruder might successfully launch an attack. A "*" return shows that a particular hop has exceeded the three-second timeout. These are hops that could be used by intruders to create DoSs. Duplicate entries for successive hops are indications of bugs in the kernel of that gateway or looping within the routing table.

Checking the open ports and services available is another important aspect of firewall and proxy server testing. There are a number of programs — like the freeware program strobe, IBM Network Services Auditor (NSA), ISS Internet Scanner™, and AXENT Technologies NetRecon™ — that can perform a selective probe of the target UNIX or Windows NT network communication services, operating systems and key applications. These programs use a comprehensive set of penetration tests. The software searches for weaknesses most often exploited by intruders to gain access to a network, analyzes security risks, and provides a series of highly informative reports and recommended corrective actions.

There have been numerous attacks in the past year that have been directed at specific ports. The teardrop, newtear, oob, and land.c are only a few of the recent attacks. Firewalls and proxy hosts should have only the minimum number of ports open. By default, the following ports are open as shipped by the vendor, and should be closed:

- echo on TCP port 7
- echo on UDP port 7
- discard on TCP port 9
- daytime on TCP port 13
- daytime on UDP port 13
- chargen on TCP port 19
- chargen on UDP port 19
- NetBIOS-NS on UDP port 137
- NetBIOS-ssn on TCP port 139

Other sources of information leaks include Telnet, FTP, and Sendmail programs. They all, by default, advertise the operating system or service type and version. They also may advertise the host name. This feature can be turned off and a more appropriate warning messages should be put in its place.

Sendmail has a feature that will allow the administrator to expand or verify users. This feature should not be turned on on any host in the DMZ. An intruder would only have to Telnet to the Sendmail port to obtain user account names. There are a number of well-known user accounts that an intruder would test. This method works even if the finger command is disabled.

VRFY and EXPN allow an intruder to determine if an account exists on a system and can provide a significant aid to a brute-force attack on user accounts. If you are running Sendmail, add the lines Opnovrfy and Opnoexpn to your Sendmail configuration file, usually located in /etc/sendmail.cf. With other mail servers, contact the vendor for information on how to disable the verify command.

```
# telnet xxx.xxx.xx.xxx
Trying xxx.xxx.xx.xxx...
Connected to xxx.xxx.xx.xxx.
Escape character is '^]'.
220 proxy.abc.com Sendmail 4.1/SMI-4.1 ready at Thu, 26 Feb 98 12:50:05
CST
expn root
```

```
250- John Doe <jdoe>
250 Jane User <juser>
vrfy root
250- John Doe <jdoe>
250 Jane User <juser>
vrfy jdoe
250 John Doe <john_doe@mailserver.internal.abc.com>
vrfy juser
250 John User <jane_user@mailserver.internal.abc.com>
^]
```

Another important check that needs to be run on these hosts in the DMZ is a validation that the system and important application files are valid and not hacked. This is done by running a checksum or a cyclic redundancy check (CRC) on the files. Because these values are not stored anywhere on the host, external applications need to be used for this function. Some suggested security products are freeware applications such as COPS and Tripwire, or third-party commercial products like AXENT Technologies Enterprise Security Manager™ (ESM), ISS RealSecure™ or Kane Security Analyst™.

Summary

The assumption must be made that one is not going to be able to stop everyone from getting in to a computers. An intruder only has to succeed once. Security practitioners, on the other hand, have to succeed every time. Once one comes to this conclusion, then the only strategy left is to secure the perimeter as best one can while allowing business to continue, and have some means to detect the intrusions as they happen. If one can do this, then one limits what the intruder can do.

35

Extranet Access Control Issues

Christopher King, CISSP

Many businesses are discovering the value of networked applications with business partners and customers. Extranets allow trading partners to exchange information electronically by extending their intranets. The security architecture necessary to allow this type of communication must provide adequate protection of corporate data and the proper separation of data among users (e.g., confidential partner information). The information security technologies must minimize the risk to the intranet while keeping the extranet configuration flexible. Corporations are acting as service providers, providing a common network and resources to be shared among the user base. The Web server is evolving into a universal conduit to corporate resources. Without adequate security controls, extranet security will become unmanageable.

Introduction

Most extranets are used for business-to-business (B2B) and electronic commerce applications between trading partners and external customers. Historically, these applications used value-added networks (VAN) with electronic data exchange (EDI) transactions. The VANs provided a private point-to-point connection between the enterprises, and EDI's security was inherent in the format of the data and the manual process after transmission. VANs, by design, were outsourced to VAN providers (e.g., Sterling, IBM, GEIS, and Harbinger). With the advent of virtual private network (VPN) technology, providing a private channel over a public network (i.e., the Internet), VAN-based EDI growth is currently at a standstill. A new data interchange format based on Extensible Markup Language (XML) is rivaling EDI for Internet-enabled applications.

Companies can use an extranet to:

- Supplement and possibly replace existing VANs using EDI
- Project management and control for companies that are part of a common work project
- Provide a value-added service to their customers that are difficult to replace
- Share product catalogs exclusively with wholesalers or those "in the trade"
- Collaborate with other companies on joint development efforts

There are two distinct types of extranets: a one-to-many and a many-to-many. A one-to-many is more common, linking many companies to a single resource (e.g., home banking). A many-to-many extranet is viewed as the intersection of a number of different company intranets (e.g., the Automotive Network Exchange). Extranets are soaring because they facilitate a seamless flow of information and commerce among employees, suppliers, and customers and because they sharply reduce communication costs. Extranet connectivity can be used for short- and long-term business relationships. This chapter concentrates on the access control mechanism and the administration aspects of extending one's intranet. The access control enforcement mechanisms generally fall into the following categories: **network** — VPN, firewall, intrusion detection; **authentication** — certificate, token, password; **platform** — intrusion detection, compliance management, Web-to-Web server, Web agent, monitoring, and auditing.

For an extranet to be successful it must be contained within a secure environment and add value to the existing line of business. Organizations that are currently implementing intranets should consider a security infrastructure that allows them to securely extend the intranet to form an extranet. This will allow them to leverage information sharing between trading partners.

Who is on the Wire?

Intranet, extranet, and the Internet are all networks of networks. The major difference between the three classes of networks is the aspect of network traffic control (i.e., who are the participants in the network). Intranets are owned by individual organizations (i.e., intra-enterprise systems). Some organizations operate their own network, and some outsource that function to network operations groups (e.g., EDS, AT&T Data Solutions, etc.). A key characteristic of intranet implementation is that protected applications are not visible to the Internet at large. Intranet access control relies heavily on the physical access point to the corporate LAN. Once physical access is gained into a corporate site, application access controls are the only constraint on access to corporate resources. Secure intranets are separated from the Internet by means of a firewall system. Inbound Internet traffic is NOT allowed into the corporate security perimeter except for e-mail. Outbound network traffic destined to the Internet from the intranet is not usually filtered. Some corporations constrain outbound traffic to allow only Web-based protocols (e.g., HTTP, FTP, and IIOP).

The rise in remote access usage is making the reliance on physical proximity to the corporate LAN a moot point. With a growing number of access points in today's corporate intranets, network and application security has to become more stringent to provide adequate protection for corporate resources. The lines between the intranet and other classes of networks are becoming blurred.

A one-to-many (e.g., provider-centric) extranet is a *secure* extension of an enterprise intranet. A many-to-many (e.g., user-centric) extranet is a secure extension of two or more enterprise intranets. This secure extension allows well-defined interactions between the participating organizations. This private network uses the Internet protocols and possibly the public network as a transport mechanism. "Private" means that this network is not publicly accessible. Only the extranet providers' suppliers, vendors, partners, and customers are allowed onto this network. Once access is gained to the network, fine-grained application and platform controls must exist (i.e., a combination of network and application security must be in place) to further restrict access to data and resources. The technology for building an extranet is essentially the same as that for intranets (e.g., Web-based). This does not mean that access to extranet resources will allow an extranet user to communicate with the provider's intranet directly. There must be a secure partition between the extranet and the provider's intranet. Extranet security must be tight so corporations can develop stronger business relationships and forge closer ties with individuals who need differing levels of access to information or resources on their network. The challenge is to develop a proper security architecture that allows semi-trusted users to *share* a network with other individual organizations. These organizations could be competitors, so access control is of the utmost importance.

Internet applications that employ application-level security do not constitute an extranet. There must be a *clear separation* between the extranet resources (e.g., database, application logic or platforms) and the Internet and intranet. An extranet requires a higher level of security and privacy than traditional intranets. Most corporations have strong perimeter security and lenient security controls once inside the intranet (i.e., hard and crunchy outside and soft and chewy middle). The extranet also has to be designed with industry-standard development techniques (e.g., IP, SQL, LDAP, S/MIME, RADIUS, and especially Web).

The Internet is a global network of networks providing ubiquitous access to an increasing user base. Enterprises use the Internet and its technologies to save money and to generate revenue. The Internet technology (e.g., Web) has influenced the other classes of networks. Web development tools are plentiful and come at a relatively low cost with a short development cycle. The problems with the current state of the Internet are security and reliability. Enterprises should not rely too heavily on the Internet for time-sensitive or critical applications.

Some of the differences between an intranet and the Internet are the quality of service (QoS) or lack of service level agreements (SLAs) which describe availability, bandwidth, latency, and response time. Most Internet service providers (ISPs) and networking device vendors are developing an Internet level of service capability. This will allow for classes of services with a price differential (see Exhibit 351).

EXHIBIT 35.1 Security Enforcement Categories for Each Network Classification

Enforcement	Intranet	Extranet	Internet
Security policy enforcement	The enterprisewide security policy is enforced by the intranet security architecture.	The majority is provided by the network facilitator and agreed upon by the extranet user base.	The Internet is under no auspices for security policy enforcement.
Physical/platform access enforcement	Highly controlled — only data center personnel have physical access to application server and network equipment.	Highly controlled — only the enterprise hosting the data center personnel has physical access to application server and network equipment. If a business partner owns a piece of equipment, it is shared between both organizations.	No physical access is provided to external users.
Network access enforcement	Private — only corporate personnel have access to this network via WAN and remote access methods. All network protocols are allowed.	Semi-private — only extranet users (e.g., business partners) have access to this network. Network protocols must be filtered to protect the intranet.	Public — all external users have ubiquitous access to an organization's public information. No network protocols other than e-mail and Web are allowed.
Application access enforcement	Semi-private — application provides some level of access control. In most cases it is a very lax security environment.	Private — users must be authenticated and authorized to perform operations depending on their rights (i.e., least privilege).	None — Web-based applications are used to disseminate static information. There are some instances of protected access pages using basic authentication.
Quality-of-service guarantee	High — with the proper networking equipment (e.g., smart switches and advanced routing protocols).	Depends on the extranet provider network and participating client network provider.	None — SLA between ISPs does not exist, yet. It is in the works.

Extranet Security Policy

The goal of an extranet security policy is to act as the foundation upon which all information-security related activities are based. In order for this security policy to be effective, it must receive approval and support from all the extranet participants (i.e., senior management). The security policy must keep up with the technological pace of the information systems technology. In other words, as access to corporate resources changes with the technology, the security must be updated. The security policy must balance the organization's operational requirements with the state-of-the-art in security solutions. Because both of these are under constant change, the policy must stay current. Some of the high-level statements in an extranet policy follow.

The extranet security architecture supports the following statements:

- The extranet must be securely partitioned from the corporate intranet.
- Secure network connectivity must be provided using a dedicated line or using a VPN.
- Extranet users must be uniquely identified using adequate authentication techniques.

- Authorization must adhere to the least-privilege principle.
- Extranet managers will receive monthly access reports to verify the proper use of the network.
- The extranet must NOT provide a routable path to the participant networks (i.e., the extranet provider's network should not allow packets to flow between partner networks).
- A real-time monitoring, auditing, and alerting facility must be employed to detect fraud and abuse.

Before the extranet can be connected to the outside world, the extranet provider must understand its network and the application vulnerabilities of extranet users and internal intranet users. This usually involves a detailed risk assessment by a certified third party. It also includes a formal review of the baseline security policy and security architecture that it meets. The assessments should be periodic, exhaustive, and include all of the member organizations of the extranet.

Secure extranet applications provide a well-defined set of data and resources to a well-defined set of authenticated individuals. To properly design authorization into an application, some basic security concepts must be employed, such as separation of duties, least privilege, and individual accountability. Separation of duties is the practice of dividing the steps in a critical function (e.g., direct DBMS access, Java applet updates) among different individuals. The least-privilege principle is the practice of restricting a user's access (DMBS updates or remote administration), or type of access (read, write, execute, delete) to the minimum necessary to perform the job. Individual accountability consists of holding someone responsible for his actions. Accountability is normally accomplished by identifying and authenticating users of the system and subsequently tracing actions on the system to the user who initiated them.

Network Partitioning

To enforce the proper separation of networks, a commercial suite of network access control devices must be used. Separating the networks from each other offers one level of security necessary for a secure extranet solution. The proper network topology must exist to further protect the networks. A combination of firewalls and real-time intrusion detection configured to be as stringent as possible should adequately control network traffic flow. Exhibit 35.2 depicts such a topology.

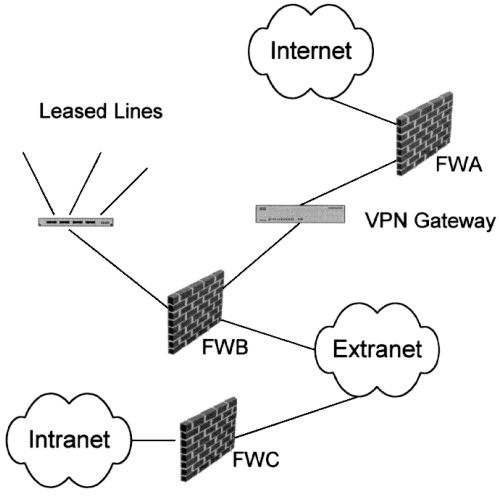


EXHIBIT 35.2 Extranet network Topology.

Each network is protected using a commercial firewall product (e.g., Checkpoint Firewall-1, Cisco PIX). There is no direct connection from the Internet to the intranet. The firewall closest to the Internet (FWA) only allows encrypted traffic into the VPN gateway. Most commercial firewalls have been around since 1994; VPN devices started appearing in early 1998. Because VPN devices are latecomers to the Internet, it is better to protect them with a firewall than to leave them unprotected from current and future Internet threats. Because the data is decrypted after the VPN gateway, it should be filtered before entering the extranet (FWB). The provider's intranet is protected from any extranet threats using an additional firewall (FWC).

Extranet users gain access to the extranet by traditional means (e.g., leased lines) or by using VPNs. In a one-to-many extranet, clients must not be able to communicate directly with each other via the extranet. The network routing rules must enforce a non-loopback policy (i.e., a network route between two clients).

Extranet Authentication

User accountability is the ability to bind critical data functions to a single user. It holds users responsible for their actions. The extranet security architecture must enforce user accountability. At the network level, user accountability is impossible because of proxy servers, application gateway firewalls, and address translation. All the users from an organization will have the same IP address. Authentication must be performed at the application layer.

Extranet authentication is not a trivial task due to its political nature, not due to its technology. Most users already have too many passwords to remember to access their own system. Because user administration is typically distributed to the partnering organization, once users have authenticated themselves to their own organization, they should not have to authenticate themselves again to the extranet. The extranet application should leverage the authentication information and status from the user's originating organization using a proxy authentication mechanism. This allows users to gain access to the extranet resources once they have authenticated themselves to their local domain.

Device authentication includes VPN gateways and public key infrastructure (PKI)-aware servers (e.g., Web and directory servers using Secure Socket Layer, SSL). VPN gateways optionally can use a shared secret instead of certificates, but this technique is unmanageable if the device count is too high.

Specific examples of proxy authentication techniques are NT domain authentication, cross certification with digital certificates, RADIUS, and a shared directory server.

Extranet Authorization

Once network access is granted, it is up to the application (most likely Web-based with a database back end) to provide further authentication and authorization. Most Web server access control is provided using basic authentication. The user's rights (i.e., Web files and directories they have access to) and authentication information combined is called a user's profile. This information is stored and enforced locally on the Web server. Local Web access controls are not a scalable solution, if the user base is large, then this type of solution is unmanageable. Access to Web files and directories is sufficient for static content security. New Web development tools ease the access into database, mainframe, and BackOffice systems. Web applications are starting to look more and more like traditional client/server applications of a few years ago. The Web server is becoming a universal conduit to corporate resources.

There are many access control enforcement points between the Web server and the data being accessed, such as the browser, the firewall, the application server, or the DBMS.

Exhibit 35.3 depicts how third-party Web access control (WAC) products such as Encommerce getAccess, Netegrity Siteminder, and Axent Webdefender provide Web login, authentication, authorization, personal navigation, and automated administration. Due to the Web's stateless nature, cookies are used to keep state between the browsers and the server. To prevent modification of the cookie by the end user, it is encrypted. The Web server must be modified to include a Web agent. The Web agent uses the Web server API (e.g., NSAPI for Netscape Enterprise Server and ISAPI for Microsoft's Internet Information Server). Access control information is controlled from a single point. Once a change is made in the security rulebase, it is replicated to all of the protected Web servers.

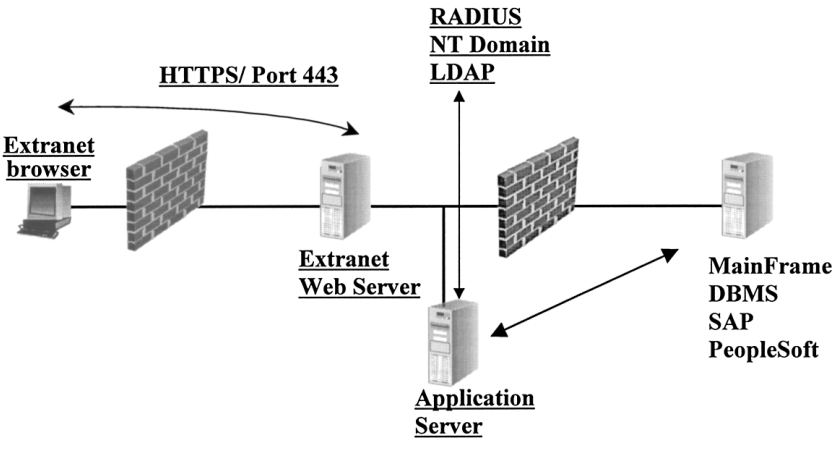


EXHIBIT 35.3 Web access control architecture.

Extranet Administration

Extranet system administration is performed by the organization providing the service. However, user administration remains a touchy subject. The user administration of the extranet is dictated by the relationships between the participating organizations. Extranet managers are the points-of-contact at each organization and are legally responsible for their users. For example, is user authentication centrally administered by the extranet provider, or is it distributed among the participants, leveraging it off their existing authentication database? It would be difficult to manage 1000 business partners with 1000 users each.

Corporate users are already inundated with username/password pairs. If extranet access were provided over the corporate network, another authentication scheme would only complicate the issue. Several questions that need to be addressed come to mind: (1) How can we integrate with an external business partner's security infrastructures? (2) How do we leverage the participants' existing security infrastructure?

Authentication is only a piece of the pie; what about authorization? Do we provide authorization at the user level, or use the concept of roles, grouping users into functions, for example, business managers, accountants, user administrators, clerks, etc.?

The way users get access to sensitive resources (i.e., items you wish to protect) is by a role-resource and user-role relationship. The extranet authorization model consists of the totality of all the user-role and role-resource relationships. This information is usually stored within a relational DBMS or a directory server. The extranet's system administrator, with input from the resource owners, is responsible for creating and maintaining this model.

The principle of least privilege will be used when an administrator assigns users to the system. Least privilege requires that an administrator grant only the most restrictive set of privileges necessary to perform authorized tasks. In other words, users will access their necessary resources to perform their job function with a minimum amount of system privileges.

Extranet Connection Agreements

Allowing access to private data from external business partners could pose some liability issues. One of the major problems is that the legal systems lag significantly behind the advances in technology. From an insurance coverage standpoint, the problem that underwriters have is the inability to calculate the security exposure for a given information system. The best defense is a proper security architecture derived from a detailed security policy. This solves the enterprise security problem, but in most cases the corporate security policy cannot be extended outside the enterprise. A separate extranet data connection agreement must be developed and adhered to by all participants. This agreement would specify the basic terms and conditions for doing business together in a secure fashion.

The following lists some considerations for data connection agreements:

- A description of the applications and information that will be accessible by the external partner
- A point of contact(s) for each participating organization, to be contacted in the event of a security incident
- The legal document (e.g., non-disclosure, and security procedures) signed by partners and the external customer's authorized representative
- The term or length (days), and start and end dates, of the service
- A protection of information statement that details the safeguard requirements (e.g., copying, transmitting to third parties, precautions, destruction) of the data transmitted
- The sharing of responsibilities by both parties; this includes the necessary access for a physical security audit and a logical security audit (e.g., network penetration tools) at each facility
- An indemnification statement that each party agrees to compensate the other party for any loss or damages incurred as a result of unauthorized access to the data facilities and misuse of information
- A termination statement that is executed if either party fails to adhere to the data connection agreement provisions
- Security awareness training for users at external or partner sites

Extranet Monitoring

Extranet monitoring is important for security and business reasons. Frequent analysis of audit data is useful in case questions arise about improper systems access and to generate marketing report data (i.e., how many times were my resources accessed and by whom).

Security monitoring usually occurs wherever access control decisions are being made, for example, the firewall, authentication server, and the application itself. The problem with monitoring is that there is no real-time analysis of the data, just log entries in some file or database. Data reduction from raw data logs is not a trivial task. No standards exist for data storage or formats, and users must compile diverse logs of information and produce their own reports from the application, firewall, or network operating system. The audit trail entries must contain a specific user ID, timestamp, function, and requested data. Using a scripting language such as PERL, a security manager will have to write a set of scripts to generate reports of log-in times, data accessed, and services used. In more security-intensive applications, the enterprise should install some real-time analysis tools (e.g., Internet Security Systems' RealSecure or Cisco's Net Ranger) to generate additional data and monitor for anomalous behavior.

Extranet Security Infrastructure

The extranet security infrastructure consists of all the supporting security services that are required to field a security architecture. Such an architecture would include a directory server, a certificate server, an authentication server, and Web security servers. These require firewall server management, the issuance and use of digital certificates or similar means of user authentication, encryption of messages; and the use of virtual private networks (VPNs) that tunnel through the public network.

VPN Technology

Virtual private network technology allows external partners to securely participate in the extranet using public networks as a transport (i.e., Internet). VPNs rely on tunneling and encapsulation techniques, which allow the Internet Protocol (IP) to carry a wide range of popular non-IP traffic (e.g., IPX, NetBEUI). VPN technology provides encryption and authentication features within an ancillary network device to firewalls and routers called a VPN gateway. Performance enhancements in the Internet backbone and access equipment now provide the throughput needed to compete with private networks. All of these enabling technologies are based on standards that yield end-to-end interoperability. Finally, preparing Points of Presence (POPs) for VPNs is relatively simple and inexpensive. Low costs with high margin VPNs are good business.

Because VPN technology uses encryption as the basis for its security, interoperability among vendors is a major issue. The Internet Engineering Task Force (IETF) IP Security (IPSec) specification was chosen to alleviate this problem. The IETF developed IPSec as a security protocol for the next generation IPv6. IPSec is an optional extension for the implementation of the current version, IPv4. IPv4 is widespread on the Internet and in corporate networks, but its design does not include any security provisions. IPSec provides confidentiality and integrity to information transferred over IP networks through network layer encryption and authentication. IPSec protects networks from IP network attacks, including denial of service, man-in-the-middle, and spoofing. Refer to Requests for Comment (RFC)2401 through 2412 for full details.

Before VPN devices can communicate, they must negotiate a mutually agreeable way of securing their data. As part of this negotiation, each node has to verify that the other node is actually the node it claims to be. VPN authentication schemes use digital certificate or a shared secret between communicating devices. A shared secret is a password agreed upon by the two device administrators in advance. When the administrators try to communicate, each must supply the agreed-upon password. Authentication based on certificates is more secure than password-based authentication because of distribution and formation. Passwords have to be difficult to guess and shared in a secure fashion. Because certificates are based on public key technology, they are immune to this problem.

With all of this said, using VPNs has the following drawbacks:

Drawback	Description
Not fault tolerant	VPN devices are not fault tolerant. The IPSec protocol does not currently support failover. This should be addressed and implemented before the end of 2000.
Performance	There are many implementation choices for VPNs (e.g., software, black box, and outboard cryptographic processors). Software solutions tend to be used for clients. Because VPN gateways are aggregating many simultaneous connections, a software-only gateway cannot keep up. Outboard cryptographic processors are used to assist in the intense cryptographic function by host-based devices (e.g., PCI slot). None of these solutions can compete with a dedicated hardware device (e.g., black box).
Reliable transport	The Internet service providers are not yet capable of providing adequate, peak or scalable bandwidth at a reasonable cost. Cisco and some of the large ISP are testing a technology called MultiProtocol Label Switching. MPLS allows the ISPs to offer different levels of service to their customer base.
Network placement	Most enterprises manage their own or outsource control over their Internet firewall. Where should the VPN gateway be placed? In front of, behind, parallel with, or on the firewall? These are questions with many trade-offs.
Addressing	Networks are not generally additive. Special care has to be taken in terms of addressing before joining two or more disparate networks. If two or more of the networks are using private address space (e.g., 10.x.x.x) with any overlap, routing can be tricky.
Key management (PKI)	VPN formation requires cryptographic information. Shared secrets between points are not scalable. The only solution is certificates. The problem that exists is that this technology is about six to nine months behind the VPN technology, which was finalized in November 1998.
Interoperability	IPSec compliance is a term that is overused by VPN vendors. The only real compliance is an interoperability report among heterogeneous vendors. As of this writing there are only six vendors who can fully interoperate.

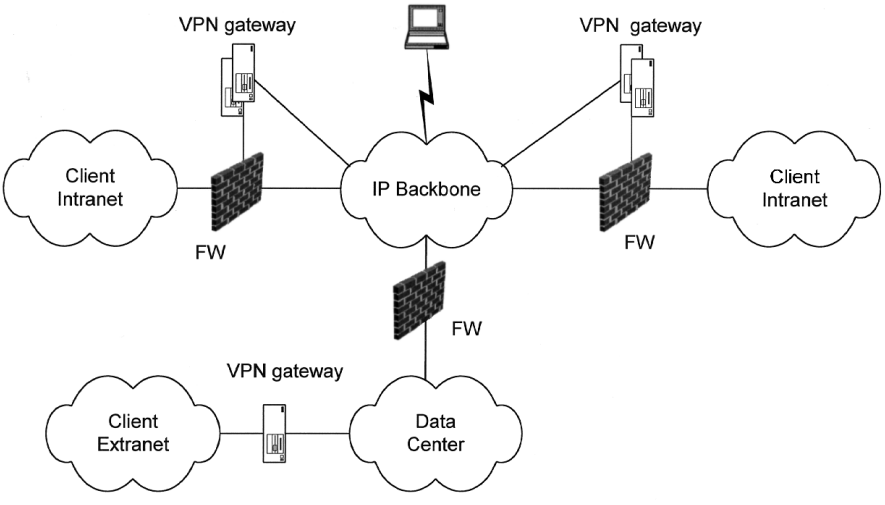


EXHIBIT 35.4 Outsourced extranet architecture.

Residual Risks/Liability

There is no such thing as complete security. There is an associated cost with providing an adequate level of security; the adequacy is measured against the best business practices in the industry. The addition of more security safeguards comes at a high cost and only offers a minor increase in the overall security level. Extranet security has the additional burden of providing even more security and privacy from participants who are competitors. Unauthorized access to repositories of information and applications could, in the wrong hands, prove detrimental to their participants. The resolution is to manage the risk and to weigh the benefits against the resultant risk. As a supplement to all of the security mechanisms, a lawyer should be involved in the extranet data agreement. The lawyer can draw up necessary warnings to deter casual intruders as well as agreements to protect your company in the event of misuse of the data. An alternative might be to outsource the extranet to a service provider.

Extranet Outsourcing

Many ISPs and telcos are offering extranet services that provide a managed network with controlled access. Extranet service providers have a strong technical knowledge of networking and security. They also have invested in the infrastructure required to manage an extranet, for example, a PKI with an X.500 directory service. Another advantage is that the service provider can offer better network reliability and bandwidth (e.g., service level agreements). If all the extranet participants utilize this existing service provider, an SLA can be negotiated. See Exhibit 35.4 for an example architecture of an outsourced extranet.

Automotive Network Exchange

The Automotive Network eXchange (ANX) is a many-to-many extranet between Chrysler Corp., General Motors Corp., and Ford Motor Company and their suppliers. This extranet utilizes VPN technology. ANX will be used to electronically route product shipment schedules, order information, engineering and drawing files for product designs, purchase orders, and other financial information. ANX replaces 50 to 100 direct-dial connections to the automakers, reducing telecommunication costs up to 70 percent, but the real payoffs are in the speed and ease of communications between suppliers and manufacturers. The real benefit is monetary

savings estimated in the billions from the traditional supply-chain costs and the speed of new automotive designs to less than a three-year design cycle. The improved exchange of information should result in new business practices between vendors and manufacturers.

Summary

Extranets have indeed arrived and may well mean changes to how business relationships are viewed. The key to maximizing participation is to make the extranet as accessible to as many partners as possible, regardless of their technical adeptness. The more participants there are, the greater the rates of return from the system. Major enterprise resource planning (ERP) systems (e.g., Baan and SAP) are providing hooks to allow external business partners to connect with automated back-office systems.

The network boundaries (extra, intra, and Inter) continue to erode so one will have to depend on application layer security. The problem is providing a common, or standard, protection scheme for applications. This is another emerging field of security, probably with a two- or more-year development and integration cycle.

The desire to provide an enhanced layer of security, reliability, and quality of service on top of the Internet will be the primary driver of VPNs as a subset of electronic commerce extranet deployment. These features are not offered by most ISPs. Next-generation Internet and Internet2 research and development projects are testing very high-speed (gigabit) networks. Large telephone companies are laying the foundation for the networks into which the Internet may eventually evolve, as well as the support equipment (routers, switches, hubs, and network interface cards) needed to drive networks at such high speeds. Network security and virtual private network technologies will be improved, which will facilitate future extranets.

Glossary

ANXAutomotive Network eXchange

B2BBusiness-to-Business

DBMSDatabase Management System

EDIElectronic Data Interchange

FTPFile Transfer Protocol

HTTPHyperText Transfer Protocol

IETFInternet Engineering Task Force

IIOPInternet Inter-ORB Protocol

IPInternet Protocol

IPSecIP Security

ISAPIInternet Server Application Program Interface

LDAPLightweight Directory Access Protocol

MPLSMultiProtocol Label Switching

NSAPINetscape Server Application Program Interface

PCIPeripheral Component Interconnect

PKIPublic Key Infrastructure

QoSQuality of Service

RADIUSRemote Authentication Dial-In User Service

RSARivest, Shamir, and Adleman

S/MIMESecure Multi-purpose Internet Mail Extension

SLAService Level Agreement

Network Layer Security

Steven F. Blanding

INTRODUCTION

Modern computer networks today are characterized by layered protocol architectures, allowing network designs to accommodate unlimited applications and interconnection techniques. This layered approach allows protocols to become modularized, that is, developed independently and put together with other protocols in such a way as to create one complete protocol. The recognized basis of protocol layering is the Open Systems Interconnection (OSI) architecture. The OSI standards establish the architectural model and define specific protocols to fit into this model, which defines seven layers. Protocols from each of the layers are grouped together into an OSI layer stack, which is designed to fulfill the communications requirements of an application process.

Standards are also needed to adequately support security in the OSI layered communications architecture. A broad, coordinated set of standards is required to ensure necessary security functionality and still provide a cost-effective implementation. Because of the complexity and flexibility of the OSI model, security must be carefully defined to avoid an increased potential for functions being duplicated throughout the architecture and incompatible security features being used in different parts of the architecture. There is also a possibility that different and potentially contradictory security techniques can be used in different applications or layers, where fewer techniques would provide the required results with less complexity and more economy.

Security standards were added to the OSI architecture to provide a broad, coherent, and coordinated approach to applying security functionality. The security standards can be grouped into categories as follows: (1) security architecture and framework standards, (2) security techniques standards, (3) layer security protocol standards, (4) application-specific

security standards, and (5) security management standards. This chapter will focus primarily on Network Layer Security, which is part of the family of layer security protocol standards. However, because the standards are closely interrelated, a brief overview of the security architecture and framework standards is required. These standards serve as a reference base for building standards in the other categories, including Network Layer Security.

NETWORK LAYER STRUCTURE, SERVICE, AND PROTOCOL

The Network Layer of the OSI model accommodates a variety of subnetwork technologies and interconnection strategies, making it one of the most complex of the seven layers in the model. The Network Layer must present a common service interface to the Transport Layer and coordinate between subnetworks of different technologies. There are also two styles of operation, connection-oriented and connectionless, that significantly contribute to this complexity.

There are three ISO standards that describe the Network Layer services, including ISO/IEC 8648, ISO/IEC 8880, and ISO/IEC 8348. The internal organization of the Network Layer is explained by the ISO/IEC 8648 standard. The general principles and the provision and support of the connection-mode and connectionless-mode network services are explained by the ISO/IEC 8880 standard. The network service definition, which includes the connection-mode, connectionless-mode addendum, and addressing addendum, is explained by the ISO/IEC 8348 standard. This standard also describes the concepts of *end system* and *intermediate system*. An end system models hardware across a complete seven-layer OSI communications model, while an intermediate system, which is located in the Network Layer, only functions across the lowest three OSI layers. Communications by an end system can occur directly with another end system or through several intermediate systems.

Intermediate systems can also include or refer to a real subnetwork, an internetworking unit connecting two or more real subnetworks, or a mix of both a real subnetwork and an internetworking unit. A collection of hardware and physical links that connect real systems is called a *real subnetwork*. Examples of real systems include local area networks or public packet-switching networks. With this foundation, many different Network Layer protocols can be established. Because the protocol can exist at the subnetwork level within the Network Layer, they do not need to be designed to specifically support the OSI standard. As a result, support for all the functions required by the Network Layer service does not need to be provided by the basic protocol of a subnetwork. To achieve OSI standard functionality, further sublayers of protocol can be provided above the subnetwork protocol.

Regardless of the type of interconnection designed, one of three roles is performed by a Network Layer protocol. These roles are subnetwork-independent convergence protocol (SNICP), subnetwork-dependent convergence protocol (SNDCP), and subnetwork-access protocol (SNAcP). The SNICP role provides functions to support the OSI network service over a well-defined set of underlying capabilities, which are not specifically based on any particular subnetwork. The role is to convey addressing and routing information over multiple interconnected networks and commonly applies to the interconnecting protocol used. The SNDCP role operates over a protocol to provide the SNAcP role in order to add capabilities required by an SNICP protocol or needed to provide the full OSI network service. The SNAcP role provides a subnetwork service at its end points, which may or may not be equivalent to the OSI network service. This protocol is inherently part of a particular type of subnetwork.

ISO/IEC 8473 identifies another protocol that is very important to the Network Layer — the Connectionless Network Protocol (CLNP). This protocol provides connectionless-mode network service within a SNICP role. The definition for how this protocol operates over X.25 packet-switched subnetworks or LAN subnetworks is contained within the ISO/IEC 8473 standard.

SECURITY SERVICE ARCHITECTURAL PLACEMENT

When designing security, significant decisions need to be made as to the layers(s) where data item or connection-based protection should be applied. Implementing security services in a layered communications architecture can be a complicated endeavor and can raise significant issues. The concept of protocol layering implies that data items can be embedded within data items and connections can be embedded within connections, with potentially multiple layers of nesting.

Guidance for where security services should be applied within the OSI model is identified in standard ISO/IEC 7498-2. As the first formal standard addressing layer assignment of security services, this standard, while providing guidance as to which OSI layers are appropriate for providing security services, does allow for many options. The security required is application dependent. Some services may need to be provided in different layers in different application scenarios, while some may even need to be provided in multiple layers in the same scenario. The complexity of these security services can be illustrated by a pair of end systems communicating with each other through a series of subnetworks.

An end system is typically defined as one piece of equipment, either a PC work station, minicomputer, or mainframe computer. An end system is described as having only one policy authority for security purposes. A

collection of communications facilities employing the same communications technology is a subnetwork. An example of a subnetwork is a local area network (LAN) or wide area network (WAN). A subnetwork is described as having only one policy authority for security purposes. Each subnetwork, however, typically has a different security environment and, as a result, will probably have a different policy authority. Also, an end system and the subnetwork to which it is connected may or may not have the same policy authority.

Another complication typically found in end systems is that they often simultaneously support multiple applications, such as e-mail, file access, and directory access for multiple users. These applications often need considerably different security requirements. Not only may security requirements differ among end systems and for subnetworks, but they may also vary within a subnetwork. Subnetworks generally comprise multiple links connecting multiple subnetwork components, and different links may pass through different security environments. As a result, individual links may need to be protected through a security mechanism.

To reduce the complexity, security services can be described more simply and effectively within a four-level model. The four levels at which specific and distinct requirements for security protocol elements arise include the application, end system, subnetwork, and the direct-link levels. In the application level, security protocol elements are application dependent. In the end-system level, security protocol elements provide protection on an end system-to-end system basis. In the subnetwork level, security protocol elements provide protection internal to a subnetwork, which is considered less trusted than other parts of the network environment. In the direct-link level, security protocol elements provide protection internal to a subnetwork, over a link that is considered less trusted than other parts of the subnetwork environment.

When determining where to locate security services within these four basic architectural layers, some general properties must first be examined that vary between higher and lower levels. These general properties include traffic mixing, route knowledge, number of protection points, protocol header protection, and source/sink binding.

Traffic mixing is a term used to describe the mix of data traffic between higher and lower levels of the OSI layer architecture. With the introduction of multiplexing, lower levels tend to have a greater tendency toward data items from different source and destination applications and users mixed in the data stream than at higher levels. The type of security policy can significantly alter this factor. In instances where the security policy tends to leave individual applications or users to specify the data protection required, placing security services at a higher level tends to be better.

Individual applications or users will have inadequate protection where security is specified at lower levels. In addition, some data would also be unnecessarily protected because of the security requirements of other data sharing the data stream.

Route knowledge is also an important factor in security placement. There tends to be more knowledge of the security characteristics of different routes and links at lower levels than at higher levels. Placing security at lower levels can have effectiveness and efficiency benefits in an environment where such characteristics vary significantly. Where protection is unnecessary on subnetworks or links, security costs can be eliminated, while targeted security services are specifically employed as appropriate.

The number of protection points can vary significantly depending on where security protection is placed. If security were placed at a very high level, such as the application layer, then security would also need to be placed in every sensitive application in every end system. If security were placed at a very low level, such as the direct-link level, then security would also need to be placed at the ends of every network link. If security were placed closer to the middle of the architecture, then security features would tend to need to be placed at significantly fewer points.

To have adequate protocol header protection, security services need to be placed at a low level. If security services were placed at higher levels, lower-level protocol headers would not receive protection, which in some environments may be sensitive.

Source/sink binding is the association of data with its source or sink. Implementation of data origin authentication and nonrepudiation security services depends on this binding. These security services are most effectively achieved at higher levels, especially at the application level. However, subject to special constraints, it can sometimes be achieved at lower levels.

END SYSTEM-LEVEL SECURITY

End system-level security relates to either the Transport Layer or subnetwork-independent Network Layer protocols. Standards have been developed supporting both options, ISO/IEC 10736 for the Transport Layer and ISO/IEC 11577 for the Network Layer. The types of security requirements that are suitable for an end system-level security solution fall into three broad categories. The first includes requirements relating to network connections that are not linked to any particular application. The second includes requirements dictated by the end-system authority that are to be enforced upon all communications regardless of the application. Finally, the third includes requirements based on the assumption that the end

systems are trusted, but that all underlying communications network(s) are untrusted.

In choosing between the Transport Layer or Network Layer for placement of end-level security protection, factors favoring the Network Layer approach include: (1) the ease of transparently inserting security devices at standardized physical interface points, (2) the ability to support any upper-layer architecture, including OSI, Internet, and proprietary architectures, and (3) the ability to use the same solution at the end-system and subnetwork levels.

SUBNETWORK-LEVEL SECURITY

Subnetwork-level security provides protection across one or more specific subnetworks. Subnetwork-level security needs to be distinguished from end system-level security for two important reasons. First, equipment and operational costs for subnetwork-level security solutions may be much lower than those for end system-level solutions because the number of end systems usually far exceeds the number of subnetwork gateways. Second, subnetworks close to end systems are trusted to the same extent as the end systems themselves since they are on the same premises and administered under the same conditions. As a result, subnetwork-level security should always be considered as a possible alternative to end system level security. In the OSI architecture, subnetwork-level security maps to the Network Layer.

NETWORK-LAYER SECURITY PROTOCOL

The network layer is among the complex of layers within the OSI model. As a result, several OSI standards are required to specify transmission, routing, and internetworking functions for this layer. The ISO/IEC 8880 standard describes an overview of the Network Layer. Two other standards, ISO/IEC 8348 and 8648, define the network service and describe the internal organization of the Network Layer, respectively. The most recent standard published is ISO/IEC 11577, which describes the Network-Layer Security Protocol (NLSP).

Different sublayers make up the Network Layer, each performing different roles, such as subnetwork access protocol (SNAcP) and subnetwork-dependent convergence protocol (SNDCP). The architectural placement of the NLSP can be in any of several different locations within the Network Layer, functioning as a sublayer. Above its highest layer is the Transport Layer, or possibly a router where a relay or routing function is in place.

Two service interfaces, the NLSP service interface and the underlying network (UN) service interface, are contained within the Network-Layer

Security Protocol. The NLSP service is the interface presented to an entity or sublayer above, and the UN service is the interface to a sublayer below. These service interfaces are specified in such a way as to appear like the network service, as defined in ISO/IEC 8348. The Network-Layer Security Protocol can also be defined in two different forms or variations, connection-oriented and connectionless. In the connection-oriented NLSP, the NLSP service and the UN service are connection oriented, whereas in the connectionless NLSP, these services are connectionless. The flexibility of the architecture results from the ability of the NLSP to support both end system-level or subnetwork-level security services.

For example, in a connection-oriented NLSP, suppose we defined X.25 as the underlying subnetwork technology. In this configuration, the NLSP is placed at the top of the Network Layer (just below the Transport Layer and just above the X.25 subnetwork), allowing the NLSP service to equate to a secure version of the OSI network service. In this example, the X.25 protocol is not aware that security is provided from above.

The NLSP can also provide subnetwork level security. In instances where the subnetwork is untrusted, the NLSP adds the necessary security, which can equate to either the OSI network service in the end system or to the network internal layer service (NILS) in a relay system. In connectionless cases, several configurations with practical applications are possible, such as the transfer of fully unencrypted connectionless network protocol (CLNP) headers, encrypted CLNP addresses with parts of the header not encrypted, or fully encrypted CLNP headers.

SECURE DATA TRANSFER

Encapsulation is a security function used to protect user data and sensitive parameters. In both connection-oriented and connectionless NLSP, the primary function is to provide this protection originating on request or response primitives issued at the NLSP service. The encapsulation function applies this security by generating data values for corresponding request or response primitives issued at the UN service, which is then reversed at the receiving end. This is very similar to the process used in the TLSP, where the generation and processing of the Security Encapsulation PDU occurs.

Different encapsulation functions are available for different environments within the NLSP. This provision includes the basic encapsulation function, which is very similar to the encapsulation function defined in the TLSP. The NLSP does have some additional features included in the basic function. Each octet string to be protected contains a string of fields including: (1) address parameters requiring protection, (2) quality-of-service

parameters requiring protection, (3) an indicator of the type of primitive (e.g., connect request, connect response, disconnect, etc.), (4) user data requiring protection, (5) test data for use in testing cryptographic system operation, and (6) security label.

When compared to the TLSP, the protection process is the same, with the exception of two additional fields included within the generated PDU. These are an integrity sequence number (ISN) and a traffic padding field. The integrity sequence number is used to support sequence integrity. Because transport protocol sequence numbers could serve this purpose in the TLSP, this feature was not required within that layer. The traffic padding field is used to support the traffic flow confidentiality service, which is a requirement of the NLSP but not the TLSP.

The encapsulation function can include either a clear header process or, as an alternative to the basic encapsulation function, a no-header process. In the clear header feature, a clear header is prefixed to the resulting protected octet string to give an NLSP secure data transfer PDU, which contains the security association identifier. The no-header encapsulation feature is also available for optional use only with connection-oriented NLSP. The no-header option can be used when the only security mechanism applied is encryption and when the encryption–decryption processes do not change the data lengths. In the no-header alternative, the secure data transfer PDU is replaced by an encrypted version of the data requiring protection. This allows the NLSP to be inserted transparently within the Network Layer. The data characteristics of the underlying services, such as data rates, packet sizes, and bandwidth, are not affected. As a result, security functions can easily be added to an existing service without changing the network architecture. However, the range of services that can be supported is greatly reduced because ICV, ISN, padding, and security labels cannot be used. Integrity services can still be maintained where the data has sufficient natural redundancy and if cryptographic chaining is used. Basic confidentiality is also not compromised and can still be supported.

The mapping of the same type of NLSP service primitives to UN service primitives, with the exception of connection establishment and release, is how the NLSP operates. If fields do not require protection, they are copied directly from one service primitive to the other. Those NLSP fields that do require protection are processed by the encapsulation function. The encapsulated result, or secure data transfer PDU, is mapped to a user data parameter of the UN service primitive. The application of the encapsulation function may result in data expansion, which could require the use of segmentation.

CONNECTION ESTABLISHMENT AND RELEASE

As mentioned previously, special procedures are required to handle connection establishment with connection-oriented NLSP. The NLSP is similar to the TLSP in that it not only supports internal security protocol, but also security associations managed by other means. The use of special procedures is dependent upon whether or not security association establishment needs to occur in conjunction with connection establishment.

Even where a suitable security association already exists (in other words, a situation not involving security association establishment), there is a requirement for a special NLSP protocol exchange at connection establishment time. This is needed to perform peer entity authentication, establish particular encryption and integrity keys for use on the connection, and to establish starting integrity sequence numbers. In this case, a connection security control PDU is defined in the NLSP to convey this information. At connection establishment, a two-way exchange of these PDUs occurs. The type of connection authentication mechanism specified for the particular security association determines the variation in the precise contents of the PDU. The PDU fields would include a security label, key reference or key derivation information, and encrypted versions of two integrity sequence numbers, one for each direction in traffic. Successful decryption of the integrity sequence number field can simultaneously provide protection against replay attacks on authentication, demonstrate key knowledge for authentication purposes, and confirm starting integrity sequence numbers.

The data exchanges may be much more complex where security association establishment is to occur in conjunction with connection establishment. This additional complexity is typically addressed through the definition of a separate security association PDU. This separate PDU is used to handle the need for more than a two-way exchange for authentication and key derivation purposes, as well as substantial attribute negotiation. Again, like the TLSP, the NLSP does not require a particular security association establishment technique. Instead, one suitable technique based on the Authenticated Diffie–Hellman exchange is described.

The last area of discussion in this section is a description of how the protocol exchanges for NLSP connection establishment map onto the UN service. Mapping directly onto the UN connection establishment primitives would be the ideal situation. However, in reality the required NLSP protocol exchanges add substantial overhead and prevent this possibility. There may not be space in the UN connection establishment PDUs for all the data that needs to be transferred since user data fields of network protocols are commonly limited in length. In addition to this, a multi-way protocol exchange may be needed to establish a security association.

These conditions require that two basic mapping alternatives be defined. An NLSP connection establishment can map directly to UN connection establishment where only a two-way exchange is necessary, and all required data can fit in the user data fields of the UN connect primitives. If these conditions do not exist, the required data transfers map to UN data exchanges following UN connection establishment. Additional complications may occur where data transfers map to UN data exchanges. There is a possibility that the throughput, window size, quality-of-service, and other service parameters eventually negotiated do not match the characteristics of the UN connection. When this occurs, a new UN connection is established with the required, now known, characteristics, and the original UN connection is released.

Mapping problems may also occur where, upon release of an NLSP connection, user data on the disconnect needs to be protected by the encapsulating function and the resultant PDU cannot fit in the user data parameter of UN disconnect. The NLSP PDU must map to a UN data exchange prior to UN disconnect in this scenario. The NLSP is a powerful and complex protocol because of the large number of possible mapping scenarios.

SUMMARY

In general, lower-layer security protocols support end system-level, direct-link-level, and subnetwork-level security services. Security services at the subnetwork and end system levels support confidentiality, integrity, access control, and authentication services. Security services at the direct-link level support confidentiality only. These services differ according to whether the environment is connection oriented or connectionless.

Throughout the lower layers, the concepts of protection quality-of-service and security associations are used. To signal protection requirements across layer boundaries and to negotiate requirements between two ends, protection quality-of-service is used. To provide a consistent type of protection to a sequence of data transfers between two systems, a security association is used to model the collection of related attribute information maintained between those systems. A security association can be established through Application Layer protocol exchanges, lower-layer protocol exchanges in the same layer that uses the security exchange, or through nonstandard methods.

The NLSP is very flexible, functioning at either the end-system or subnetwork level. The NLSP can be positioned at any of several places in the Network Layer, functioning as a sublayer. NLSP is able to conceal trusted subnetwork protocol information while this information travels through an untrusted subnetwork, depending on its positioning within the Network

Layer. Variations of NLSP include connection-oriented and connectionless. The connection-oriented variant works in conjunction with such protocols as X.25, and the connectionless variant works in conjunction with the Connectionless Network Protocol (CLNP). An encapsulation process very similar to that of TLSP is used by NLSP. To provide for the establishment of security associations, optional protocol support is used.

Transport Layer Security

Steven F. Blanding

INTRODUCTION

The Transport Layer of the OSI model ensures that a reliable end-to-end data transmission capability of the quality demanded by the session layer is offered to that layer, regardless of the nature of the underlying network over which the data will be transferred. This chapter will examine the services offered to transport service (TS) users and the security associated with the Transport Layer.

The basic Transport Layer standards are found in the ISO (International Organization for Standardization) /IEC (International Electrotechnical Commission) 8072 transport service definition, the ISO/IEC 8073 connection-oriented transport protocol specification, and the ISO/IEC 8602 connectionless transport protocol specification. These documents were first published in 1986 and 1987. Subsequent to these publications, security functionality was added to the Transport Layer with the completion of the Transport Layer Security Protocol (TLSP) standard, ISO/IEC 10736, in 1993. The U.S. government project initiated by the National Security Agency (NSA), Secure Data Network System (SDNS), produced specification Security Protocol 4 (SP4), which became the primary input to the development of the TLSP. SP4, which was a product of both industry and government, specifies security services, mechanisms, and protocols for protecting user data in networks based on the OSI model. The TLSP, even with additional contributions made toward it, is still based mostly on SP4.

Before the Transport Layer Security Protocol is presented, an overview of the Transport Layer is provided in order for the reader to have a basic understanding of the material, which is necessary to understand the security architecture.

TRANSPORT LAYER OVERVIEW

The transport service is defined in the ISO service definition document 8072. The transport service is in one of three phases at any time: (1) transport connection (TC) establishment, (2) data transfer, or (3) transport connection release. In the TC establishment phase, a connection is established between peer TS users (session entities). The session entity initiating the TC specifies the quality of service required of the connection, in terms of reliability and other aspects of the service. Once a TC is established, the session entities can exchange Transport Service Data Units (TSDUs) transparently over the connection. In the release phase, the TC is unconditionally released by either TS user.

The reliable end-to-end transmission of data is provided by the T-DATA service element, and the expedited data is provided by the T-EXPEDITED-DATA service element. The required level of service of the TC is dictated to the initiating transport entity in the quality-of-service (QOS) parameter of the T-CONNECT request. This is used as a basis for negotiation, during TC establishment, of an acceptable and attainable QOS between the end systems. The TS provider throughout the lifetime of the connection must then maintain this negotiated QOS.

The parameters associated with each TS primitive include *called address, calling address, expedited data option, quality of service, TS user data, responding address*, and *disconnect reason*. The called address and calling address are TSAP addresses and identify the TS user initiating the TC and the intended responder. The responding address conveys the TSAP as the called address, only differing from it when that address has been supplied by the initiating TS user in some generic form. Such a form results in a selection, by the responding end system, of a specific TSAP address, which is based upon the provided generic. It is this selection that is returned in the parameter. The expedited data option parameter is used to negotiate the availability of transport-expedited data service over the TC. If the calling TS user or TS provider does not offer this service, which is apparent in the T-CONNECT indication, then the called TS user may not insist upon it by including it on the response.

TS user data is a parameter that, in the case of T-DATA and T-EXPEDITED-DATA, is the mechanism for provision of transparent, reliable, TSDU exchange over a TC between peer TS users. In the case of the other services, this parameter enables a limited amount of transparent user data to be passed between TS users, which may qualify the services in question. TS user data is restricted in length according to service element type: a maximum of 32 octets for T-CONNECT, 64 octets for T-DISCONNECT, 16 octets for T-EXPEDITED-DATA, and no restriction for T-DATA.

The unconstrained size of normal data TSDUs will often not apply in practice. Constraints on implementation or on the operational environment of a transport entity, such as the size available buffering, lead to a limit being imposed on TSDUs. Such a limit will have repercussions on the higher layers, but these can be overcome by the use of segmentation by the peer session entities. Segmentation is the facility by which a Session Service Data User (SSDU), as an object of a data request, can be transmitted between peer session entities not in a single Session Protocol Data Unit (SPDU) but in segments, that is, in several consecutive SPDUs.

The quality of service parameter is itself a "list" of parameters. It is, on the T-CONNECT request, a statement by the initiating TS user concerning the level of service it requires of the, as yet unestablished, TC. It is concerned with such things as acceptable error rates and minimum acceptable data throughput. Both the calling and called transport entities may amend the QOS to a level they regarded as feasible, given knowledge of aspects of the network not necessarily visible to the initiating TS user. In the course of establishing the connection, the QOS is passed to the responding TS user in the indication. Acceptance of the connection results in a T-CONNECT confirmation, which carries a final QOS. If this is modified to an unacceptable level, the initiating TS user has the option of terminating the established connection by issuing a T-DISCONNECT request with an appropriate reason parameter value and also qualifying user data, such as "QOS negotiated to unacceptable level."

The reason parameter of the T-DISCONNECT indication gives the cause of the TC release. It shows whether the release was user or provider initiated, and could include the possible values "quality of service fallen below level agreed for this TC," "congestion or failure of local or remote TS provider," "unknown reason," "called TSAP address not valid," or "called TSAP address not available."

SUBNETWORK RELIABILITY

Errors originating in a subnetwork and consequently observed by the transport layer are of two types, *signaled* and *residual*. A signaled error is one detected by the network layer but where no steps are taken within that layer for recovery. The event is just signaled to the transport layer for recovery. Two examples are network disconnection (the network connection is lost) and network reset (the network connection is reset to a known state, possibly with loss of data in transit, but the connection remains available for use).

Residual errors are those apart from signaled errors. In effect, the network layer has not detected them. Examples are loss, corruption, duplication, and delivery out of sequence of TSDUs.

Subnetworks that are analyzed in terms of these two types of errors are categorized as either (1) a subnetwork where the rates of both types of errors are acceptable, (2) a subnetwork where the rate of residual errors is acceptable but not that of signaled errors, or (3) a subnetwork where the rate of residual errors is unacceptable. A network connection offered over a number of subnetworks of different error categories should expect a level of service that is the poorest level of service of the subnetworks over which it operates.

As part of transport connection establishment, the peer transport entities must establish the level of network service enhancement that must be undertaken in order to provide the agreed QOS for this connection. This involves the selection of the set of procedures that will be used during the connection. This selection is achieved as part of the connection establishment procedure in parallel with QOS negotiation.

TRANSPORT CLASSES

There is a set of five basic levels or classes of network service enhancement available from the Transport Layer. Each class is in some way related to the three categories of subnetwork identified above. Transport entities during TC establishment perform the procedure negotiation described above by agreeing on a transport to be used over the network for this particular TC. Inherent in a choice of class is a set of associated transport procedures.

Class 0, the simple class, provides the most minimal overhead, a basic transport connection designed to be used with network service where the rates of both types of errors are acceptable. Given that this type of network service provides reliable data transmission, only a basic level of transport activity is required. Class 1, the basic error recovery class, provides, with minimal overhead, a basic transport connection designed to be used with network services where the rate of residual errors is acceptable but not that of signaled errors. It handles signaled errors such as network disconnect without involving the TS user. Class 2, the multiplexing class, is as class 0 but with additional mechanisms to support the multiplexing of transport connections onto single network connections. Class 3, the error detection and recovery class, is as class 1 but with additional multiplexing mechanisms. Class 4, the error detection and recovery class, provides all the capability of class 3 together with mechanisms required to detect and recover from errors not signaled by the NS provider. This class also provides for increased throughput and for additional resilience against NS provider failure. It is designed to be used over a type of network where the rate of residual errors is unacceptable.

TRANSPORT PROCEDURES

The transport protocol is defined as a set of procedures, each of which relates to a particular activity. Implicit in the final negotiated transport class is the choice of a subset of those procedures that is necessary to provide the functionality of that class. An examination of the procedures will reveal that many are fundamental to basic transport service provision. These form a set of procedures common to all transport classes. These procedures include, but are not limited to, the following: assignment to a network connection; transport protocol data unit transfer; segmentation and reassembling; concatenation and separation; connection establishment; connection refusal; normal release; error release; association of TPDUs with transport connections; TPDU numbering; expedited data transfer; reassignment after failure; retention until acknowledgment of TPDUs; resynchronization; multiplexing and demultiplexing; explicit flow control; checksum; frozen references; retransmission on timeout; resequencing; inactivity control; treatment of protocol errors; and splitting and recombining.

Assignment to a network connection is a procedure that is common to all classes. Until an assignment is made, a transport class connection cannot be established. Assignment is the association of a TC with a network connection (NC). In the TC establishment stage, establishment cannot proceed until an assignment is made. However, once made and the TC established, the TC can be retained and assigned to a different NC. In either case, the transport entity may choose to establish a new NC or use a suitable existing NC.

The *transport protocol data unit transfer* procedure coordinates the conveyance of TPDUs between peer transport entities. It uses the network normal and expedited data service elements N-DATA and N-EXPEDITED-DATA. This procedure is common to all classes of transport. In the transport data PDUs, DaTa (DT) and Expedited Data (ED), the structure is such that the control section of the PDU, the protocol control information (PCI), comprises an identifier together with the length parameter giving the length of the PCI within the PDU. However, there is no length indication for the data field and the PDU. The whole is passed to the NS provider as NSDU, and it is from the overall length of this NSDU that the receiving transport entity can determine the size of the data field, which is calculated as the NSDU length minus the PCI length.

Segmentation and reassembling may also occur within the transport layer. A TSDU requested for transfer by a TS user may exceed the limit placed upon the amount of data that can be conveyed between peer transport entities in a single data (DT) TPDU. Such a limit reflects constraints within the network service on NSDUs associated with the N-DATA service

element. In this case, segmentation is invoked to break the TSDU into a series of appropriately sized DT TPDUs. On receipt by the peer transport entity, the sequence of DT TPDUs representing a segmented TSDU will be reassembled into the single TSDU. When this complete TSDU has been received, a data service indication is issued to the complete TS user. The End of Transport (EOT) parameter in each DT TPDU is only set when a complete TSDU has been transferred, and is used to recognize a segmented TSDU by a receiving transport entity. Where TSDUs are contained entirely within DT TPDUs, the EOT is set on every DT TPDU.

The transport layer also provides for *concatenation and separation* of TPDUs. A number of TPDUs can be concatenated into a single NSDU for transmission and separation by the receiving transport entity on receipt. If a data TPDU is one of the group of concatenated TPDUs, then it must be the last TPDU of the concatenation, and as a result, it can be the only TPDU.

The *connection establishment* procedure is available in all classes of transport to establish a TC after successful assignment to a network connection. A transport connection is established by negotiation between peers by the exchange of appropriate PDUs, which is conveyed by the use of network normal data, N-DATA. As a result of negotiation, the QOS to be maintained and the transport class to be used over the network are determined. There are optional procedures associated with particular classes that are in themselves optional within the class, and so negotiation of these optional features is also carried out at this time. For example, "Expedited Data Transfer" and "Retention Until Acknowledgment of TPDU" are both optional features in Class 1.

Connection refusal is a procedure that is initiated by the responding transport entity in response to either a T-DISCONNECT request from the responding TS user, or an inability to conform to the requirements of the initiating transport entity conveyed in the CR TPDU. Connection refusal is achieved by sending a Disconnect Request (DR) TPDU to the initiator using network normal data.

There are two types of release procedures — *normal release* and *error release*. A normal release can be described through two variants — implicit and explicit. In Class 0, the implicit variant of the normal release is achieved by disconnecting the NC using the N-DISCONNECT request, the receipt of which is considered to imply the release of the associated TC. The explicit variant of normal release is associated with all other classes. Under the explicit variant, the TC is released by a confirmed activity involving the exchange between peers of Disconnect Request (DR) and Disconnect Confirm (DC) TPDUs, using network normal data. An error release is used only in Classes 0 and 2. This is used to release the transport connection after a

signaled error has been received from the NS provider. The TS user is notified of the release by a T-DISCONNECT indication.

The *association of TPDUs* with transport connections is a procedure used in all classes while data is being received. Three actions are taken when a transport entity receives an NSDU from an NS provider. First, a check is made to determine that the NSDU can be decoded into one or more concatenation of TPDUs. Second, if concatenation is detected, then the separation procedure is invoked. Finally, where multiple TCs are associated with an NC over which the NSDU is received, ensure the TPDUs are associated with the appropriate TC.

TPDU numbering is a feature required to ensure that certain procedures are successfully performed. This is a sequence number, identified as a parameter in the PCI, which is carried in each DT TPDU. The procedures include those involving flow control, resequencing, and recovery.

The *expedited data transfer* procedure places the TS user data provided by a T-EXPEDITED-DATA request into the data field of an Expedited Data (ED) TPDU. Although the transport expedited data service is unconfirmed, transport protocol demands that the peer entity procedure be confirmed, and so each ED TPDU must be acknowledged by the receiving peer transport entity by use of an expedited data acknowledge (EA) TPDU. No more than one acknowledged ED TPDU can be outstanding for each data flow direction of the TC at any time.

The *reassignment after failure* procedure is invoked when a network signaled error is received, indicating the loss of the NC to which a TC is assigned. The result will be that the TC is assigned to a different NC, which either already existed and was owned by the transport entity or is newly created for the purpose. The procedure, resynchronization, is invoked when this reassignment is achieved; however, should a reassignment not be achieved, the TC will be considered released and the transport reference frozen. The *frozen reference* procedure (described below in a paragraph on *frozen references*) is then used to ensure that a reference is not reassigned to another TC after being frozen.

Retention until acknowledgment of TPDUs provides mechanisms whereby the transmitting transport entity can retain "copies" of TPDUs until an explicit acknowledgment is received from the peer. Should no acknowledgment be received after a certain period of time has elapsed, or should a signaled error occur, then the TPDUs can be retransmitted. The persistent loss of TPDUs will cause the QOS to fall below the negotiated acceptable level and the TC to be terminated.

Resynchronization is a procedure used to restore the TC to normal after reassignment of a TC after NC failure or after a signaled event from the NS provider, which indicates a problem in the NC. The purpose of the resynchronizing transport entity is to resume the activity on the TC that was outstanding at the time of the triggering event. Resynchronization is only attempted by the initiating transport entity of the TC. The peer takes only a passive role in the resynchronization process. Since both entities are aware of the need for resynchronization, one of the peer transport entities must take a passive role or the resynchronization by both peers would result in unnecessary event collision resolutions. The passive entity responds by setting a timer for resynchronization-related TPDUs to be received from the TC initiator. If resynchronization does not occur, the timer expires and the entity considers the TC released and the reference frozen.

Multiplexing and demultiplexing procedures are available to Classes 2, 3, and 4. This process allows more than one TC to share a single NC. Multiplexing takes place where a transport entity transmits or receives TPDUs belonging to different TCs over the same NC. The transport entity receiving the TPDUs must perform demultiplexing. Demultiplexing is accomplished by invoking the association of TPDUs procedure where the TC to which individual TPDUs belong is determined. Network efficiencies are obtained where both multiplexing and concatenation procedures are used together, and a single NSDU is transferred containing concatenated TPDUs for different TCs.

Explicit flow control is a procedure available to Classes 2, 3, and 4. In Class 2 explicit flow control is optional and in Classes 3 and 4 it is mandatory. This procedure regulates the flow of DT TPDUs between peer transport entities over a TC within the transport layer and acts independently of flow control available in the network.

Checksum is an optional procedure used only in Class 4. The checksum is a value calculated according to an algorithm defined in the protocol specification, which has the octets comprising the TPDU with which it is associated as its arguments. The checksum is identified in the TPDU as the checksum parameter. After transmission over the network, the checksum is recalculated and compared to the value in the TPDU parameter. Corruption is assumed if the values are different. In this situation, the TPDU is discarded, no acknowledgment is sent, and the transmitting transport entity retransmits the TPDU.

Frozen references are used by Classes 1, 3, and 4. They are used to ensure that a reference is not reassigned to another TC after being frozen. References are information relating to the identity of a TC. *Retransmission on timeout* is a procedure used to provide retransmission by the sender of

TPDUs that appear to have become lost. In this situation, the transmitting transport entity detects lost TPDUs when it does not receive an acknowledgment during a fixed time period and when acknowledgments are known to be outstanding. When this happens, the first TPDU in the sequence of unacknowledged TPDUs is retransmitted, and the timer is reset and left to expire. After several retransmissions without acknowledgment, the sending transport entity will invoke the release procedure and inform the TS user of the failure. Only Class 4 uses this procedure.

The *resequencing* procedure is used to sort misordered DT TPDUs by the NS provider. This provides for correctly ordered octets delivered to the TS user by each TPDU regardless of the inconsistencies of the network, which may cause out-of-order TPDUs. Misordering can occur when a TPDU is segmented by the transport entity into many TPDUs and where splitting results in these TPDUs traveling between end systems spread over a number of network connections.

The procedure that addresses unsignaled termination of a network connection is *inactivity control*. This procedure, which is used only in Class 4, is invoked on the expiry of an inactivity timer maintained by the transport entity. It times the period over which no TPDU is received. Inactivity control expires after a fairly lengthy interval and then invokes the *normal release* procedure. To protect against termination because of inactivity due to traffic congestion, the interval must be long enough to avoid timing out a good connection.

The *treatment of protocol errors* procedure is used when a TPDU is received that cannot be interpreted under the rules of the standard, when no error has been received and there are no checksum errors. Several different appropriate actions are possible depending on the operational details of the errors. This procedure is used in all classes.

The TC is enabled to make use of multiple NCs through the procedure *splitting and recombining*. The result of this can be increased throughput or greater resilience against failure in particularly unreliable networks. Once an association exists between one TC and many NCs, TPDUs of that TC can be transmitted over any of the NCs. As a result, TPDUs may arrive at the peer transport entity out of sequence. This procedure is only available to Class 4.

EXPEDITED DATA

Expedited data is a special form of data transfer where data is guaranteed to arrive at the receiving user before any data subsequently transmitted by a call on any data service. The intention is that data transferred by the use of expedited data will arrive before normal data already submitted for transmission by the user that has not yet been delivered; however, it

will not arrive before any previously submitted, undelivered expedited data. While expedited data is known only generally at higher levels of the OSI model, it is at the Transport Layer where the mechanics of expedited data become visible.

Expedited data is class dependent. That is, expedited data is provided entirely within Classes 2, 3, and 4 but is not provided in Class 0. In these classes, expedited TSDUs are sent as ED TPDUs over network normal data service. In Class 1, the expedited effect is provided by the expedited mechanism within the Transport Layer, together with the use of the network expedited data service to convey ED TPDUs. If this network service is not available, then the network normal data service is used.

QUALITY OF SERVICE

An application signals its lower-layers communications requirements using the concept of quality of service (QOS). This signaling occurs via a QOS parameter, which accompanies a connection establishment request or connectionless data item passed from the upper layers to the lower layers across the Transport Layer service boundary. The Transport Layer uses a similar QOS parameter in a connection-establishment request or connectionless data item it passes to the Network Layer. If the Network Layer cannot provide an adequate QOS, the Transport Layer should upgrade the provided QOS to the requisite level by adding value in its own protocol. This is done by selecting the appropriate transport protocol class and options.

The QOS parameter can convey a great deal of information covering such requirements as throughput, residual error rate, and connection failure probability. QOS can be expressed as a set of performance criteria. They generally fall into two groups: speed and accuracy/reliability. The connection establishment phase criteria include QOS parameters for *establishment delay* and *establishment failure probability*. The connection release phase criteria consists of the QOS parameters *release delay* and *release failure probability*. The data transfer phase criteria include the QOS parameters *throughput*, *transit delay, residual error rate, connection resilience,* and *transfer failure probability*.

The component of QOS relevant to security is called protection QOS. It is used to indicate the security services that need to be invoked and the strength of the mechanism that needs to be used to support a security service. TLSP and NLSP use a definition of protection QOS which includes a component for each relevant security service. For each component, it is possible to specify an integer value which indicates a required level for that service. The range of integers available and the meanings of the particular values are not specified in a standard. They are implied by the

particular agreed-upon set of security rules for the security association in use. The use of integers implies an ordering relationship between levels, with a higher level implying a stronger mechanism.

The level-based approach to protection QOS can be supplemented by the passing of a security label between the layers, such as between the transport and network service layers. This label serves as an indicator of required QOS. The security labels used for this purpose may be the same labels used to support access control, but they would have a different meaning. For example, the label "unclassified but sensitive" might imply use of a commercial-grade confidentiality mechanism based on DES encryption, where the label "secret" implies use of a confidentiality mechanism with a higher-grade classified encryption algorithm.

At either the Transport Layer or Network Layer, the establishment of QOS for a connection involves negotiation between the two peer entities, with the aim of best matching the QOS requirements of the two service users with the capabilities of the two service providers. With protection QOS, another element is introduced. Either peer entity may inject, at the service-provider level, administration protection QOS constraints. These are minimum-security requirements imposed by system administration in order to satisfy the local system security policy. For example, a user application may request a connection with no security protection at all but, depending upon circumstances, the local system administration at one or both peer entities may upgrade the required QOS to make confidentiality protection of a certain level mandatory. The negotiation of protection QOS can take place partly in security association establishment and partly in the regular exchange of QOS parameters in the connection establishment protocol.

SECURITY ARCHITECTURE

The transport layer security protocol (TLSP) is located completely within the Transport Layer. Except for the passing of protection QOS parameters, the existence and operation of TLSP are completely transparent to both the upper layers and the underlying Network Layer. TLSP is designed to supplement the regular Transport Layer protocols rather than change them. The TLSP is designed to work in conjunction with the transport protocol data unit (TPDU) and associated processing procedures of the TPDU without any modification to procedures or formats by effectively adding another protocol sublayer. Regular TPDUs are protected by being encapsulated within TLSP PDUs at the sending end prior to being passed to the Network Layer. The encapsulation is removed at the receiving end to produce the regular TPDU, which then continues under normal protocol processing.

The processing procedures are explained in ISO/IEC 8073 for connection-oriented processing and in ISO/IEC 8602 for connectionless-oriented processing. The protection of all regular PDUs associated with one transport connection is governed by one security association in the connection-oriented case. In other words, the same form of protection is applied to all PDUs. The protection scheme, however, can become more complex where Transport Layer multiplexing is located below the TLSP. In this instance, different transport connections or different connectionless TPDUs can be provided with different types of protection, even though the PDUs may be multiplexed onto one network connection between the two end systems. Where Transport Layer concatenation procedures are used, the same security association must protect all the concatenated PDUs. Concatenation procedures are located above the TLSP. The concatenated sequence of TPDUs is processed by the TLSP similarly to a single TPDU without concatenation.

SECURITY MECHANISMS

The encapsulation function of the TLSP supports the provision of several security services and can involve any required combination of security mechanisms. These mechanisms are *security label, direction indicator, integrity check-value (ICV), encryption padding*, and *encryption*.

A *security label* is prefixed to the TPDU to support the provision of an access control service. Fields are provided to define a unique defining authority identifier plus a label value in a format controlled by the defining authority. No particular label format is defined in the OIS. A *direction indicator* is a flag field prefix containing a bit indicating the direction of the TPDU transfer. This prefix contains a reference to a recognized initiator/responder relationship determined at security association establishment and is used to repulse reflection attacks. The ICV is a value that is computed and appended involving a process where padded octets are added to the data before the ICV computation. The ICV is the primary mechanism for providing both connection integrity and connectionless integrity services. *Encryption padding* is the padding of octets into the data where it is required by the encryption algorithm or for purposes of hiding lengths of protected PDUs. *Encryption* is the mechanism for providing connection or connectionless confidentiality and for providing necessary protection to information generated by other security mechanisms.

For the connection-oriented case, some security services are provided through the combined behavior of the TLSP encapsulation function and the normal procedures of the Transport Layer. Sequence integrity is achieved using the sequence numbers provided by Class 2, 3, or 4 transport protocol, together with connection integrity. Separate sequence numbering systems are maintained for normal data and expedited data flows.

Integrity recovery is accomplished using the Class 4 transport protocol recovery procedures, in conjunction with connection integrity. Sequence integrity cannot be used with Class 0 or Class 1 transport protocol.

Entity authentication is effectively a two-stage process. The first stage is security association establishment, which results in each transport entity knowing a key that it can use to verify the other entity of its identity. With security association establishment complete, the second stage is entity authentication on connection establishment. This is accomplished through each entity demonstrating knowledge of the applicable key by using that key for ICV generation or encryption in the encapsulation of the connect request TPDU. As protection against replay, the connect request and connect confirm TPDUs use connection reference values which must be unique within the lifetime of the key. This is most easily achieved by having a sequential component in the connection references. The system would then increment this component for each new connection establishment attempted.

For the connectionless case, the same basic two-stage process is used for data origin authentication. The key used in the encapsulation process is used to obtain the required authentication for a connectionless TPDU by providing a demonstrated knowledge of that key. With the key used for authentication purposes, peer addresses in connection establishment or connectionless TPDUs are also required to be checked for consistency as further protection against masquerade attacks.

SECURITY ASSOCIATION ATTRIBUTES

The TLSP also incorporates features including *security association attributes* and *agreed set of security rules (ASSR)*. The term *security association* is used to model the collections of related information maintained in two or more systems for purposes of providing the same type of protection to a sequence of distinct data transfers. The information items maintained in a security association are known as attributes of that security association. *Security association identifiers* include a local identifier and a remote identifier, which are octet strings of a length determined by the ASSR.

The term *ASSR* is used to describe an agreement between two or more systems as to which security mechanisms are to be used and which values are to be applied to parameters of those mechanisms. This avoids having to negotiate mechanism details with every security association establishment by using an agreed-upon set of security rules in a predefined package of security mechanism information. These security rules are registered and assigned a unique identifier, which is then made known to all potential users.

Other security association attributes held by a TLSP entity include *integrity sequence numbers, ICV mechanisms, encryption mechanisms, initiator/responder indicator, protection QOS, label mechanism, security mechanism,* and *peer TLSP entity address.* The last sequence numbers sent or received for normal and expedited data streams are *integrity sequence number attributes. ICV mechanism attributes* and *encryption mechanism attributes* include an algorithm, key granularity, key reference, and block size for determining necessary padding. In setting the direction, the *initiator/responder indicator* indicates which TLSP entity takes the role of initiator and which takes the role of responder. As mentioned previously, the *protection QOS indicator* is defined as a QOS label plus an integer level value for each entity service. The ASSR defines the range of integer values and the QOS label format. The set of allowable security labels for the security association is referred to as *label mechanism attributes. Security mechanism attributes* indicate which security mechanisms are used (e.g., entity authentication, security labels, integrity check values, integrity sequence numbers, and encryption). Finally, the *peer TLSP entity address* is the connection reference that is stored if the security association is tied to a particular transport connection.

SECURITY ASSOCIATION PROTOCOL

A security association may be established through Application Layer protocol exchanges (even though the security exchange is used by a lower-layer protocol), or through protocol exchanges at the same architectural layer that uses the security exchange, or through unspecified means (which may or may not involve online data communications). In order to accommodate protocol exchanges at the same architectural layer that uses the security exchange, an optional *security association protocol* in the TLSP is used.

PDU formats capable of supporting security association establishment, security association release, and the establishment of a new data key (rekeying) within a live security association is defined by the security association protocol. Establishing initial data keys and values for all security association attributes is the function of security association establishment.

LIST OF FREQUENTLY USED ACRONYMS

ASSR	Agreed Set of Security Rules
DC	Disconnect Confirm
DR	Disconnect Request
DT	Data Transport

ED	Expedited Data
ICV	Integrity Check-Value
IEC	International Electronic Commission
ISO	International Organization for Standardization
NC	Network Connection
NS	Network Service
NSDU	Network Service Data Unit
PDU	Protocol Data Unit
QOS	Quality of Service
SPDU	Session Protocol Data Unit
SSDU	Session Service Data User
TC	Transport Connection
TLSP	Transport Layer Security Protocol
TPDU	Transport Protocol Data Unit
TS	Transport Service
TSAP	Transport Service Address Protocol
TSDU	Transport Service Data Unit

SQLStructured Query Language

SSLSecure Socket Layer

VANValue Added Network

VPNVirtual Private Network

WACWeb Access Control

XMLeXtensible Markup Language

36

Application-Layer Security Protocols for Networks

Bill Stackpole, CISSP

We're Not In Kansas Anymore

The incredible growth of Internet usage has shifted routine business transactions from fax machine and telephones to e-mail and E-commerce. This shift can be attributed in part to the economical worldwide connectivity of the Internet but also to the Internet capacity for more sophisticated types of transactions. Security professionals must understand the issues and risks associated with these transactions if they want to provide viable and scalable security solutions for Internet commerce.

Presence on the Internet makes it possible to conduct international, multiple-party and multiple-site transactions regardless of time or language differences. This level of connectivity has, however, created a serious security dilemma for commercial enterprises. How can a company maintain transactional compatibility with thousands of different systems and still ensure the confidentiality of those transactions? Security measures once deemed suitable for text-based messaging and file transfers seem wholly inadequate for sophisticated multimedia and E-commerce transfers. Given the complexity of these transactions, even standardized security protocols like IPSec are proving inadequate.

This chapter covers three areas that are of particular concern: electronic messaging, World Wide Web (WWW) transactions, and monetary exchanges. All are subject to potential risk of significant financial losses as well as major legal and public relations liabilities. These transactions require security well beyond the capabilities of most lower-layer security protocols. They require application-layer security.

A Layer-by-Layer Look at Security Measures

Before going into the particulars of application-based security it may be helpful to look at how security is implemented at the different ISO layers. Exhibit 36.1 depicts the ISO model divided into upper-layer protocols (those associated with the application of data) and lower-layer protocols (those associated with the transmission of data). Examples of some of the security protocols used at each layer are listed on the right. Let's begin with Layer 1.

These are common methods for providing security at the physical layer:

- Securing the cabling conduits — encase them in concrete
- Shielding against spurious emissions — TEMPEST
- Using media that are difficult to tap — fiber optics

While effective, these methods are limited to things within your physical control.

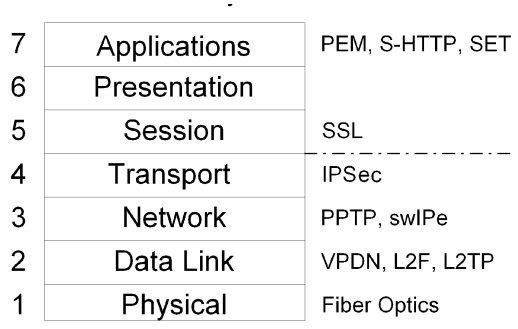


EXHIBIT 36.1 ISO seven layer model.

Common Layer-2 measures include physical address filtering and tunneling (i.e., L2F, L2TP). These measures can be used to control access and provide confidentiality across certain types of connections but are limited to segments where the end points are well known to the security implementer. Layer-3 measures provide for more sophisticated filtering and tunneling (i.e., PPTP) techniques. Standardized implementations like IPSec can provide a high degree of security across multiple platforms. However, Layer-3 protocols are ill-suited for multiple-site implementations because they are limited to a single network. Layer-4 transport-based protocols overcome the single network limitation but still lack the sophistication required for multiple-party transactions. Like all lower-layer protocols, transport-based protocols do not interact with the data contained in the payload, so they are unable to protect against payload corruption or content-based attacks.

Application-Layer Security — ALS 101

This is precisely the advantage of upper-layer protocols. Application-based security has the capability of interpreting and interacting with the information contained in the payload portion of a datagram. Take, for example, the application proxies used in most firewalls for FTP transfers. These proxies have the ability to restrict the use of certain commands even though the commands are contained within the payload portion of the packet. When an FTP transfer is initiated, it sets up a connection for passing commands to the server. The commands you type (e.g., LIST, GET, PASV) are sent to the server in the payload portion of the command packet as illustrated in Exhibit 36.2. The firewall proxy — because it is application-based — has the ability to "look" at these commands and can therefore restrict their use.

Lower-layer security protocols like IPSec do not have this capability. They can encrypt the commands for confidentiality and authentication, but they cannot restrict their use.

But what exactly is application-layer security? As the name implies, it is security provided by the application program itself. For example, a data warehouse using internally maintained access control lists to limit user access to files, records, or fields is implementing application-based security. Applying security at the application level makes it possible to deal with any number of sophisticated security requirements and accommodate additional requirements as they come along. This scenario works particularly well when all your applications are contained on a single host or secure intranet, but it becomes problematic when you attempt to extend its functionality across the Internet to thousands of different systems and applications. Traditionally, security in these environments has been addressed in a proprietary fashion within the applications themselves, but this is rapidly changing. The distributed nature of applications on the Internet has given rise to several standardized solutions designed to replace these *ad hoc*, vendor-specific security mechanisms.

EXHIBIT 36.2 File Transfer Protocol – Command – Packet

Ethernet Header	IP Header	TCP Header	Payload
0040A0…40020A	10.1.2.1…10.2.1.2	FTP (Command)	List…

Interoperability — The Key to Success for ALS

Interoperability is crucial to the success of any protocol used on the Internet. Adherence to standards is crucial to interoperability. Although the ALS protocols discussed in this chapter cover three distinctly different areas, they are all based on a common set of standards and provide similar security services. This section introduces some of these common elements. Not all common elements are included, nor are all those covered found in every ALS implementation, but there is sufficient commonality to warrant their inclusion.

Cryptography is the key component of all modern security protocols. However, the management of cryptographic keys has in the past been a major deterrent to its use in open environments like the Internet. With the advent of digital certificates and public key management standards, this deterrent has been largely overcome. Standards like the Internet Public Key Infrastructure X.509 (pkix) and the Simple Public Key Infrastructure (spki) provide the mechanisms necessary to issue, manage, and validate cryptographic keys across multiple domains and platforms. All of the protocols discussed in this chapter support the use of this Public Key Infrastructure.

Standard Security Services — Maximum Message Protection

All the ALS protocols covered in this chapter provided these four standard security services:

1. Confidentiality (a.k.a. privacy) — the assurance that only the intended recipient can read the contents of the information sent to them.
2. Integrity — the guarantee that the information received is exactly the same as the information that was sent.
3. Authentication — the guarantee that the sender of a message or transmission is really who he claims to be.
4. Non-repudiation — the proof that a message was sent by its originator even if the originator claims it was not.

Each of these services relies on a form of cryptography for its functionality. Although the service implementations may vary, they all use a fairly standard set of algorithms.

Algorithms Tried and True

The strength of a cryptographic algorithm can be measured by its longevity. Good algorithms continue to demonstrate high cryptographic strength after years of analysis and attack. The ALS protocols discussed here support three types of cryptography — symmetric, asymmetric, and hashing — using time-tested algorithms.

Symmetric (also called secret key) *cryptography* is primarily used for confidentiality functions because it has high cryptographic strength and can process large volumes of data quickly. In ALS implementations, DES is the most commonly supported symmetric algorithm. *Asymmetric or public key cryptography* is most commonly used in ALS applications to provide confidentiality during the initialization or set-up portion of a transaction. Public keys and digital certificates are used to authenticate the participating parties to one another and exchange the symmetric keys used for the remainder of the transaction. The most commonly supported asymmetric algorithm in ALS implementations is RSA.

Cryptographic hashing is used to provide integrity and authentication in ALS implementations. When used separately, authentication validates the sender and the integrity of the message, but using them in combination provides proof that the message was not forged and therefore cannot be refuted (non-repudiation). The three most commonly used hashes in ALS applications are MD2, MD5, and SHA. In addition to a common set of algorithms, systems wishing to interoperate in an open environment must be able to negotiate and validate a common set of security parameters. The next section introduces some of the standards used to define and validate these parameters.

Standardized Gibberish Is Still Gibberish!

For applications to effectively exchange information they must agree upon a common format for that information. Security services, if they are to be trustworthy, require all parties to function in unison. Communication parameters must be established, security services, modes, and algorithms agreed upon, and cryptographic keys

exchanged and validated. To facilitate these processes the ALS protocols covered in this chapter support the following formatting standards:

- X.509. The X.509 standard defines the format of digital certificates used by certification authorities to validate public encryption keys.
- PKCS. The Public Key Cryptography Standard defines the underlying parameters (object identifiers) used to perform the cryptographic transforms and to validate keying data.
- CMS. The Cryptographic Message Syntax defines the transmission formats and cryptographic content types used by the security services. CMS defines six cryptographic content types ranging from no security to signed and encrypted content. They are data, signedData, envelopedData, signedAndEnvelopedData, digestData, and encryptedData.
- MOSS. The MIME Object Security Services defines two additional cryptographic content types for multipart MIME (Multimedia Internet Mail Extensions) objects that can be used singly or in combination. They are multipart-signed and multipart-encrypted.

Encryption is necessary to ensure transaction confidentiality and integrity on open networks, and the Public Key/Certification Authority architecture provides the infrastructure necessary to manage the distribution and validation of cryptographic keys. Security mechanisms at all levels now have a standard method for initiating secure transactions, thus eliminating the need for proprietary solutions to handle secure multiple-party, multiple-site, or international transactions. A case in point is the new SET credit card transaction protocol.

Setting the Example — Visa's Secure Electronic Transaction Protocol

SET (Secure Electronic Transaction) is an application-based security protocol jointly developed by Visa and MasterCard. It was created to provide secure payment card transactions over open networks. SET is the electronic equivalent of a face-to-face or mail-order credit card transaction. It provides confidentially and integrity for payment transmissions and authenticates all parties involved in the transaction. Let's walk through a SET transaction to see how this application-layer protocol handles a sophisticated multi-party financial transaction.

A SET transaction involves five different participants: the *cardholder*, the *issuer* of the payment card, the *merchant*, the *acquirer* that holds the merchant's account, and a *payment gateway* that processes SET transactions on behalf of the acquirer. The policies governing how transactions are conducted are established by a sixth party, the *brand* (i.e., Visa), but they do not participate in payment transactions.

A SET transaction requires two pairs of asymmetric encryption keys and two digital certificates: one for exchanging information and the other for digital signatures. The keys and certificates can be stored on a "smart" credit card or embedded into any SET-enabled application (i.e., Web browser). The keys and certificates are issued to the cardholder by a certification authority (CA) on behalf of the issuer. The merchant's keys and digital certificates are issued to them by a certification authority on behalf of the acquirer. They provide assurance that the merchant has a valid account with the acquirer. The cardholder and merchant certificates are digitally signed by the issuing financial institution to ensure their authenticity and to prevent them from being fraudulently altered. One interesting feature of this arrangement is that the cardholder's certificate does not contain his account number or expiration date. That information is encoded using a secret key that is only supplied to the payment gateway during the payment authorization. Now that we know all the players, let's get started.

Step 1

The cardholder goes shopping, selects his merchandise, and sends a purchase order to the merchant requesting a SET payment type. (The SET specification does not define how shopping is accomplished so it has no involvement in this portion of the transaction.) The cardholder and merchant, if they haven't already, authenticate themselves to each other by exchanging certificates and digital signatures. During this exchange the merchant also supplies the payment gateway's certificate and digital signature information to the cardholder. You will see how this is used later. Also established in this exchange is a pair of randomly generated symmetric keys that will be used to encrypt the remaining cardholder–merchant transmissions.

Step 2

Once the above exchanges have been completed, the merchant contacts the payment gateway. Part of this exchange includes language selection information to ensure international interoperability. Once again, certificate and digital signature information is used to authenticate the merchant to the gateway and establish random symmetric keys. Payment information (PI) is then forwarded to the gateway for payment authorization. Notice that only the *payment* information is forwarded. This is done to satisfy regulatory requirements regarding the use of strong encryption. Generally, the use of strong cryptography by financial institutions is not restricted if the transactions *only contain monetary values*.

Step 3

Upon receipt of the PI, the payment gateway authenticates the cardholder. Notice that the cardholder is authenticated without contacting the purchase gateway directly. This is done through a process called dual-digital signature. The information required by the purchase gateway to authenticate the cardholder is sent to the merchant with a different digital signature than the one used for merchant–cardholder exchanges. This is possible because the merchant sent the purchase gateway certificates to the cardholder in an earlier exchange! The merchant simply forwards this information to the payment gateway as part of the payment authorization request. Another piece of information passed in this exchange is the secret key the gateway needs to decrypt the cardholder's account number and expiration date.

Step 4

The gateway reformats the payment information and forwards it via a private circuit to the issuer for authorization. When the issuer authorizes the transaction, the payment gateway notifies the merchant, who notifies the cardholder, and the transaction is complete.

Step 5

The merchant finalizes the transaction by issuing a Payment Capture request to the payment gateway causing the cardholder's account to be debited, and the merchant's account to be credited for the transaction amount.

A single SET transaction like the one outlined above is incredibly complex, requiring more than 59 different actions to take place successfully. Such complexity requires application-layer technology to be managed effectively. The beauty of SET, however, is its ability to do just that in a secure and ubiquitous manner. Other protocols are achieving similar success in different application areas.

From Postcards to Letters — Securing Electronic Messages

Electronic messaging is a world of postcards. As messages move from source to destination, they are openly available (like writing on a postcard) to be read by those handling them. If postcards are not suitable for business communications, it stands to reason that electronic mail on an open network is not either. Standard business communications require confidentiality, and other more sensitive communications require additional safeguards like proof of delivery or sender verification, features that are not available in the commonly used Internet mail protocols. This has led to the development of several security-enhanced messaging protocols. PEM is one such protocol.

Privacy Enhanced Mail (PEM) is an application-layer security protocol developed by the IETF (Internet Engineering Task Force) to add confidentiality and authentication services to electronic messages on the Internet. The goal was to create a standard that could be implemented on any host, be compatible with existing mail systems, support standard key management schemes, protect both individually addressed and list-addressed mail, and not interfere with nonsecure mail delivery. When the standard was finalized in 1993 it had succeeded on all counts. PEM supports all four standard security services, although all services are not necessarily part of every message. PEM messages can be MIC-CLEAR messages that provide integrity and authentication only; MIC-ONLY messages that provide integrity and authentication with support for certain gateway implementations; or ENCRYPTED messages that provide integrity, authentication, and confidentiality.

These are some of PEM's key features:

- *End-to-end confidentiality*. Messages are protected against disclosure from the time they leave the sender's system until they are *read* by the recipient.
- *Sender and forwarder authentication*. PEM digital signatures authenticate both senders and forwarders and ensure message integrity. PEM utilizes an integrity check that allows messages to be received in any order and still be verified — an important feature in environments like the Internet where messages can be fragmented during transit.
- *Originator non-repudiation*. This feature authenticates the *originator* of a PEM message. It is particularly useful for forwarded messages because a PEM digital signature only authenticates the last sender. Non-repudiation verifies the originator no matter how many times the message is forwarded.
- *Algorithm independence*. PEM was designed to easily accommodate new cryptographic and key management schemes. Currently PEM supports common algorithms in four areas: DES for data encryption, DES and RSA for key management, RSA for message integrity, and RSA for digital signatures.
- *PKIX support*. PEM fully supports interoperability on open networks using the Internet Public Key Infrastructure X.509.
- *Delivery system independence*. PEM achieves delivery-system independence because its functions are contained in the body of a standard message and use a standard character set as illustrated in Exhibit 36.3.
- *X.500 distinguished name support*. PEM uses the distinguished name (DN) feature of the X.500 directory standard to identify senders and recipients. This feature separates mail from specific individuals allowing organizations, lists, and systems to send and receive PEM messages.

RIPEM (Riordan's Internet Privacy Enhanced Mail) is a public domain implementation of the PEM protocol although not in its entirety. Because the author, Mark Riordan, placed the code in the public domain, it has been ported to a large number of operating systems. Source and binaries are available via FTP to U.S. and Canadian citizens from **ripem.msu.edu**. Read the **GETTING_ACCESS** file in the **/pub/crypt/** directory before attempting any downloads.

Secure/Multipurpose Internet Mail Extensions (S/MIME) is another application-layer protocol that provides all four standard security services for electronic messages. Originally designed by RSA Data Security, the S/MIME specification is currently managed by the IETF S/MIME Working Group. Although S/MIME is not an IETF standard, it has already garnered considerable vendor support, largely because it is based on well-proven standards that provide a high degree of interoperability. Most notable is, of course, the popular and widely used MIME standard, but S/MIME also utilizes the CMS, PKCS, and X.509 standards. Like PEM, S/MIME is compatible with most existing Internet mail systems and does not interfere with the delivery of nonsecure messages. However, S/MIME has the added benefit of working seamlessly with other MIME transports (i.e., HTTP) and can even function in mixed-transport environments. This makes it particularly attractive for use with automated transfers like EDI and Internet FAX.

There are two S/MIME message types: *signed*, and *signed and enveloped*. Signed messages provide integrity and sender authentication, while signed and enveloped messages provide integrity, authentication, and confidentiality. The remaining features of S/MIME are very similar to PEM and do not warrant repeating here.

A list of commercial S/MIME products that have successfully completed S/MIME interoperability testing is available on the RSA Data Security Web site at **www.rsa.com/smime/html/interop_center.html.** A public domain version of S/MIME written in PERL by Ralph Levien is available at **www.c2.org/~raph/premail.html**.

Open Pretty Good Privacy (OpenPGP), sometimes called PGP/MIME, is another emerging ALS protocol on track to becoming an IETF standard. It is based on PGP, the most widely deployed message security program on the Internet. OpenPGP is very similar in features and functionality to S/MIME, but the two are not interoperable because they use slightly different encryption algorithms and MIME encapsulations. A list of PGP implementations and other OpenPGP information is available at **http://www-ns.rutgers.edu/~mione/openpgp/**. Freeware implementations of OpenPGP are available at the North American Cryptography Archives (**www.cryptography.org**).

Taming HTTP — Web Application Security

Web-based applications are quickly becoming the standard for all types of electronic transactions because they are easy to use and highly interoperable. These features are also their major security failing. Web transactions

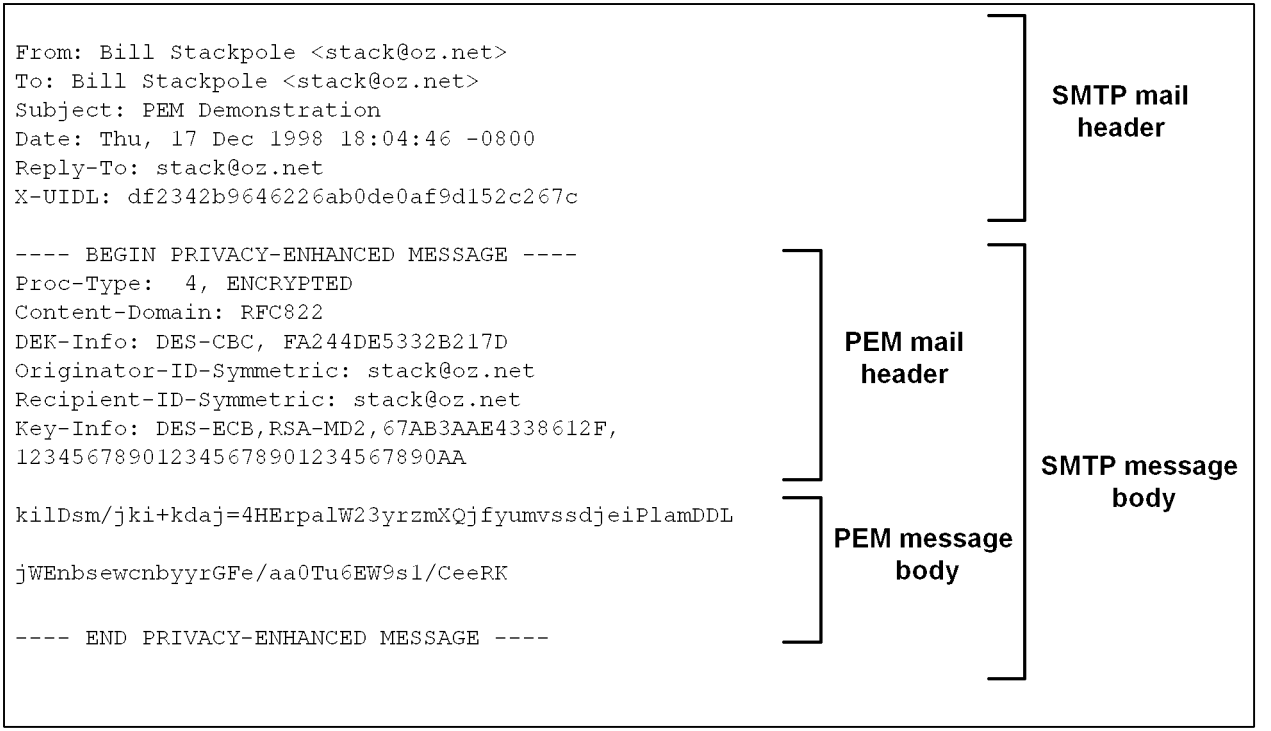


EXHIBIT 36.3 Delivery system independence.

traverse the network in well-known and easily intercepted formats, making them quite unsuitable for most business transactions. This section will cover some of the mechanisms used to overcome these Web security issues.

Secure HyperText Transfer Protocol (S/HTTP) is a message-oriented security protocol designed to provide end-to-end confidentiality, integrity, authentication, and non-repudiation services for HTTP clients and servers. It was originally developed by Enterprise Integration Technologies (now Verifone, Inc.) in 1995. At this writing, S/HTTP is still an IETF draft standard, but it is already widely used in Web applications. Its success can be attributed to a flexible design that is rooted in established standards. The prominent standard is, of course, HTTP, but the protocol also utilizes the NIST Digital Signature Standard (DSS), CMS, MOSS, and X.509 standards. S/HTTP's strict adherence to the HTTP messaging model provides delivery-system independence and makes it easy to integrate S/HTTP functions into standard HTTP applications. Algorithm independence and the ability to negotiate security options between participating parties assures S/HTTP's interoperability for years to come. Secure HTTP modes of operation include message protection, key management, and a transaction freshness mechanism.

Secure HTTP protection features include the following:

- *Support for MOSS and CMS*. Protections are provided in both content domains using the CMS "application/s-http" content-type or the MOSS "multipart-signed" or "multipart-encrypted" header.
- *Syntax compatibility*. Protection parameters are specified by extending the range of HTTP message headers, making S/HTTP messages syntactically the same as standard HTTP messages, except the range of the headers is different and the body is usually encrypted.
- *Recursive protections*. Protections can be used singly or applied one layer after another to achieve higher levels of protection. Layering the protections makes it easier for the receiving system to parse them. The message is simply parsed one protection at a time until it yields a standard HTTP content type.
- *Algorithm independence*. The S/HTTP message structure can easily incorporate new cryptographic implementations. The current specification requires supporting MD5 for message digests, MD5-HMAC for authentication, DES-CBC for symmetric encryption, and NIST-DSS for signature generation and verification.
- *Freshness feature*. S/HTTP uses a simple challenge–response to ensure that the data being returned to the server is "fresh." In environments like HTTP, where long periods of time can pass between messages, it is difficult to track the state of a transaction. To overcome this problem, the originator of an HTTP message sends a freshness value (nonce) to the recipient along with the transaction data. The recipient returns the nonce with a response. If the nonces match, the data is fresh, and the transaction can continue. Stale data indicates an error condition.

Secure HTTP Key management modes include:

- *Manual exchange*. Shared secrets are exchanged through a simple password and mechanism like PAP. The server simply sends the client a dialog box requesting a userID and password then authenticates the response against an existing list of authorized users.
- *Public key exchange*. Keys are exchanged using the Internet Public Key Infrastructure with full X.509 certificate support. S/HTTP implementations are required to support Diffie–Hellman for in-band key exchanges.
- *Out-of-band key exchange*. Symmetric keys can be prearranged through some other media (i.e., snail mail). This feature, unique to the S/HTTP, permits parties that do not have established public keys to participate in secure transactions.
- *In-band symmetric key exchange*. S/HTTP can use public key encryption to exchange random symmetric keys in instances where the transaction would benefit from the higher performance of symmetric encryption.

Many commercial Web browsers and servers implement the S/HTTP protocol, but the author was unable to find any public domain implementations. A full implementation of S/HTTP including the C source code is available in the SecureWeb Toolkit™ from Terisa (www.spyrus.com). The kit also contains the source code for SSL.

Secure Socket Layer (SSL) is a client/server protocol designed by Netscape to provide secure communications for its Web browser and server products. It was quickly adopted by other vendors and has become the

de facto standard for secure Web transactions. However, SSL is not limited to Web services; it can provide confidentiality, integrity, authentication, and non-repudiation services between any two communicating applications. The current version of SSL (SSL V3.0) is on track to becoming an IETF standard. While included here as an application-layer protocol, SSL is actually designed to function at the session and application-layers. The SSL Record Protocol provides security services at the session layer — the point where the application interfaces to the TCP/IP transport sockets. It is used to encapsulate higher-layer Protocols and data for compression and transmission. The SSL Handshake protocol is an application-based service used to authenticate the client and server to each other and negotiate the security parameters for each communication session.

The SSL Handshake Protocol utilizes public key encryption with X.509 certificate validation to negotiate the symmetric encryption parameters used for each client/server session. SSL is a stateful protocol. It transitions through several different states during connection and session operations. The Handshake Protocol is used to coordinate and maintain these states. One SSL session may include multiple connections, and participating parties may have multiple simultaneous sessions. The session state maintains the peer certificate information, compression parameters, cipher parameters, and the symmetric encryption key. The connection state maintains the MAC and asymmetric keys for the client and server as well as the vectors (if required) for symmetric encryption initialization. SSL was designed to be fully extensible and can support multiple encryption schemes. The current version requires support for these schemes:

- DES, RC2, RC4, and IDEA for confidentiality
- RSA and DSS for peer authentication
- SHA and MD5 for message integrity
- X.509 and FORTEZZA certificates for key validation
- RSA, Diffie–Hellman, and FORTEZZA for key exchange

SSL also supports NULL parameters for unsigned and unencrypted transmissions. This allows the implementer to apply an appropriate amount of security for their application. The support for the FORTEZZA hardware encryption system is unique to the SSL as is the data compression requirement. SSL uses a session caching mechanism to facilitate setting up multiple sessions between clients and servers and resuming disrupted sessions.

There is an exceptional public domain implementation of SSL created by Eric Young and Tim Hudson of Australia called SSLeay. It includes a full implementation of Netscape's SSL version 2 with patches for Telnet, FTP, Mosaic, and several Web servers. The current version is available from the SSLeay Web site at www.ssleay.org. The site includes several SSL white papers and an excellent *Programmers' Reference*.

Don't Show Me the Money — Monetary Transaction Security

The success of commerce on the Internet depends upon its ability to conduct monetary transactions securely. Although purchasing seems to dominate this arena, bill payment, fund and instrument transfers, and EDI are important considerations. The lack of standards for electronic payment has fostered a multitude of proprietary solutions, including popular offerings from Cybercash (Cybercoin), Digital (Millicent), and Digicash. However, proprietary solutions are not likely to receive widespread success in a heterogeneous environment like the Internet. This section will concentrate on standardized solutions. Since the SET protocol has been covered in some detail already, only SET implementations will be mentioned here.

Secure Payment (S/PAY) is a developer's toolkit based on the SET protocol. It was developed by RSA Data Security, although the marketing rights currently belong to the Trintech Group (www.trintech.com). The S/PAY library fully implements the SET v1.0 cardholder, merchant, and acquirer functions and the underlying encryption and certificate management functions for Windows 95/NT and major UNIX platforms. Included in the code is support for hardware-based encryption engines, smart card devices, and long-term private key storage. Trintech also offers full implementations of SET merchant, cardholder, and acquirer software. This includes their PayWare Net-POS product, which supports several combinations of SSL and SET technologies aimed at easing the transition from Web SSL transactions to fully implemented SET transactions.

Open Financial Exchange (OFX) is an application-layer protocol created by Checkfree, Intuit, and Microsoft to support a wide range of consumer and small business banking services over the Internet. OFX is an open specification available to any financial institution or vendor desiring to implement OFX services. OFX uses SSL with digital certificate support to provide confidentiality, integrity, and authentication services to its

transactions. The protocol has gained considerable support in the banking and investment industry because it supports just about every conceivable financial transaction. Currently, the OFX committee is seeking to expand OFX's presence through interoperability deals with IBM and other vendors. Copies of the OFX specification are available from the Open Financial Exchange Web site (www.ofx.net).

Micro Payment Transfer Protocol (MPTP) is part of The World Wide Web Consortium (W3C) Joint Electronic Payment Initiative. Currently, MPTP is a W3C working draft. The specification is based on variations of Rivest and Shamir's Pay-Word, Digital's Millicent, and Bellare's iKP proposals. MPTP is a very flexible protocol that can be layered upon existing transports like HTTP or MIME to provide greater transaction scope. It is highly tolerant of transmission delays allowing much of the transaction processing to take place off-line. MPTP is designed to provide payments through the services of a third-party broker. In the current version, the broker must be common to both the customer and the vendor, although inter-broker transfers are planned for future implementations. This will be necessary if MPTP is going to scale effectively to meet Internet demands.

Customers establish an account with a broker. Once established, they are free to purchase from any vendor common to their broker. The MPTP design takes into consideration the majority of risks associated with electronic payment and provides mechanisms to mitigate those risks, but it does not implement a specific security policy. Brokers are free to define policies that best suit their business requirements.

MPTP relies on S/Key technology using MD5 or SHA algorithms to authorize payments. MPTP permits the signing of messages for authentication, integrity, and non-repudiation using public or secret key cryptography and fully supports X.509 certificates. Although MPTP is still in the draft stages, its exceptional design, flexibility, and high performance destine it to be a prime contender in the electronic payment arena.

Java Electronic Commerce Framework (JECF) is our final item of discussion. JECF is not an application protocol. It is a framework for implementing electronic payment processing using active-content technology. Active-content technology uses an engine (i.e., a JAVA virtual machine) installed on the client to execute program components (e.g., applets) sent to it from the server. Current JECF active-content components include the Java Commerce Messages, Gateway Security Model, Commerce JavaBeans, and Java Commerce Client (JCC).

JECF is based around the concept of an electronic wallet. The wallet is an extensible client-side mechanism capable of supporting any number of E-commerce transactions. Vendors create Java applications consisting of service modules (applets) called Commerce JavaBeans that plug in to the wallet. These applets implement the operations and protocols (i.e., SET) necessary to conduct transactions with the vendor. There are several significant advantages of this architecture:

- Vendors are not tied to specific policies for their transactions. They are free to create modules containing policies and procedures best suited to their business.
- Clients are not required to have specialized applications. Because JavaBean applets are active content, they can be delivered and dynamically loaded on the customer's system as the transaction is taking place.
- Applications can be updated dynamically. Transaction applets can be updated or changed to correct problems or meet growing business needs without having to send updates to all the clients. The new modules will be loaded over the old during their next transaction.
- Modules can be loaded or unloaded on-the-fly to accommodate different payment, encryption, or language requirements. OFX modules can be loaded for banking transactions and later unloaded when the customer requires SET modules to make a credit card purchase.
- JavaBean modules run on any operating system, browser, or application supporting Java. This gives vendors immediate access to the largest possible customer base.

The flexibility, portability, and large Java user base make the Java Electronic Commerce Framework (JECF) a very attractive E-commerce solution. It is sure to become a major player in the electronic commerce arena.

If It's Not Encrypted Now. . .

The Internet has dramatically changed the way we do business, but that has not come without a price. Security for Internet transactions and messaging is woefully lacking, making much of what we are doing on the Internet an open book for all to read. This can not continue. Despite the complexity of the problems we are facing, there are solutions. The technologies outlined in this chapter provide real solutions for mitigating Internet

business risks. We can secure our messages, Web applications, and monetary exchanges. Admittedly, some of these applications are not as polished as we would like, and some are difficult to implement and manage, but they are nonetheless effective and most certainly a step in the right direction.

Someday all of our business transactions on the Internet will be encrypted, signed, sealed, and delivered, but I am not sure we can wait for that day. Business transactions on the Internet are increasing, and new business uses for the Internet are going to be found. Waiting for things to get better is only going to put us further behind the curve. Someone has let the Internet bull out of the cage and we are either going to take him by the horns or get run over! ALS now!

Bibliography

Crocker, S., Freed, N., Galvan, J., and Murphy, S., RFC 1848 — MIME object security services, *IETF,* October 1995.

Dusse, Steve and Matthews, Tim, S/MIME: anatomy of a secure e-mail standard, *Messaging Magazine,* 1998.

Freier, Alan O., Karlton, Philip, and Kocher, Paul C., "INTERNET-DRAFT — The SSL Protocol Version 3.0," November 18, 1996.

Hallam-Baker, Phillip, "Micro Payment Transfer Protocol (MPTP) Version 1.0," Joint Electronic Payment Initiative — W3C, November 1995.

Hirsch, Frederick, Introducing SSL and certificates using SSLeay, the Open Group Research Institute, *World Wide Web Journal*, Summer 1997.

Hudson, T.J. and Young, E.A., *SSL Programmers Reference*, July 1, 1995.

Lundblade, Laurence, *A Review of E-mail Security Standards,* Qualcomm Inc., 1998.

Pearah, David, *Micropayments,* Massachusetts Institute of Technology, April 23, 1997.

PKCS #7: Cryptographic Message Syntax Standard, RSA Laboratories Technical Note Version 1.5, RSA Laboratories, November 1, 1993.

Ramsdell, Blake, "INTERNET-DRAFT — S/MIME Version 3 Message Specification," Worldtalk Inc., August 6, 1998.

Resorla, E. and Schiffman, A., "INTERNET-DRAFT — The Secure HyperText Transfer Protocol," Terisa Systems, Inc., June 1998.

Schneier, Bruce, *E-Mail Security: How to Keep Your Electronic Messages Private*, John Wiley & Sons, 1995.

SET Secure Electronic Transaction Specification, Book 1: Business Description, Setco, Inc., May 31, 1997.

Resources

E-Payments Resource Center, Trintech Inc., www.trintech.com
The Electronic Messaging Association, www.ema.org
Information Society Project Office (ISPO), www.ispo.cec.be
The Internet Mail Consortium (IMC), www.inc.org
Java Commerce Products, http://java.sun.com
SET Reference Implementation (SETREF), Terisa Inc., www.terisa.com
SET — Secure Electronic Transaction LLC, www.setco.org
S/MIME Central, http://www.rsa.com/smime/
Transaction Net and the Open Financial Exchange, www.ofx.net

37

Application Layer: Next Level of Security

Keith Pasley, CISSP

Business applications and business data are the core backbone of most enterprises today. A current trend in business is to increase providing direct access via the Internet to certain business data to entities external to an enterprise. The two most relied upon business applications accessible from the Internet are e-mail and Web-enabled applications.

This rapidly growing trend supports various business goals that include increased competitive advantage, reduced costs, strengthened customer loyalty, establishing additional revenue streams, increased productivity, and many others. However, exposing critical business application access via the Internet does increase the risk profile for businesses. The following are possible elements of such a risk profile:

- Business operations become more dependent on the application
- Increased opportunity for exploiting application vulnerabilities
- Cost of disruption increases
- Increased targeting of the application by malicious entities
- Increased number of application-based vulnerabilities
- Speed-to-market pressures alter the performance/security dynamic of application

Such a risk profile does not necessarily imply that it is a bad or negative idea to deploy Internet-facing applications. In fact, businesses take calculated risks every day and can reap significant financial and competitive advantages by doing so. A similar disciplined approach to analyzing the relative benefits and liabilities of deploying Internet applications involves the application of risk management techniques. Essentially, risk management involves enumerating what could go wrong, how much it could cost, the likelihood of the event happening, and then deciding what responses to the event would be acceptable to the business.

Within the framework of application security, risk management involves an examination of the above on an application-by-application basis. One approach is to review the actual software code of the application as part of the software development life cycle. Goals of such a review could include subjecting the code to examination by qualified people other than the developers who originated the code. A so-called "second set of eyes" could, for example, identify vulnerabilities, check for unintended functionality, and identify bad coding practices (assuming there is an established standard against which to measure).

In some environments, code review is impractical due to the sheer volume of lines of code in a program, the time it would take for such a review, or the organizational structure may prohibit the capability of a central code review group's ability to enforce the results of the review. Additionally, in some environments where software code is changed very frequently with very little, if any, change in management

discipline, code review may simply not be appropriate. For such environments, another approach might be appropriate.

Another approach to this is to enforce a consistent application security policy via technology. One such technology is an emerging class of security components generally known as an *application firewall.* An application firewall is a security component that analyzes data at the application layer, which is often the easiest path for attackers to gain unauthorized access to enterprise resources. Most network firewalls and traditional intrusion detection systems (IDSs), in practical terms, can only control Internet Protocol (IP) packet-based network access and detect port- and protocol-type security events based on static rules or signatures. Although essential as a primary element in a comprehensive enterprise security architecture, network firewalls and IDSs are recognized as security components that can be easily vaulted over by their very nature. For example, most enterprise firewall policies allow in- and outbound access to internal or DMZ-based Web servers without meaningful inspection of the application data contained in data packets traversing the firewall. Potentially, an attacker could either send malicious data into the Web application or, conversely, extract sensitive data from the application. Application firewalls aim to consistently enforce application security policy as a security layer around an enterprise's application infrastructure.

Application firewalls are increasingly being offered by security vendors in the form of rack-mountable appliances that integrate operating system and security software preloaded on purposed-built hardware, and are engineered to balance security functionality with performance.

This chapter focuses on effective strategies for enhancing the security of Web-enabled and e-mail application infrastructures. Each is described in this chapter, yet the focus of this chapter is on the business impact of application security. As such, no detailed discussions of specific application vulnerabilities are included.

The Problem: Applications Are the Highest-Risk Attack Vector

As the Internet has created more business opportunity — for example, extending the boundaries of the enterprise outside the physical facilities of a business — so has business exposure to risk increased. If one were to identify and prioritize resources by value to the business, one would find in most cases that specific data and applications would be counted among the highest in value to an organization. Most businesses would not be able to operate competitively if data and applications were somehow taken away, either by malicious acts or by accident. Another, more granular way to look at this situation would be to imagine if the existing traditional network security controls of a data-centric business failed, would the business' critical data and applications still be protected? Not surprisingly, the answer is no. This is a realization that is being brought to the attention of data owners and security professionals by either circumstance or critical infrastructure analysis. From a technical perspective, this means that the traditional perimeter security approach of deploying firewalls and intrusion detection systems as the sole defense mechanisms is flawed with respect to current and emerging threats. Why?

One of the most important issues facing e-mail and Web-enabled businesses today is the open port problem; that is, most business firewalls allow Web application server access via port 80 and e-mail server access via port 25. Unfortunately, most traditional network firewalls are not capable of actually analyzing the data payload for malicious attacks. The majority of firewalls can only see data at the packet, or network, level — information such as source/destination IP address, TCP port number, and other packet routing information headers. This means that if an attacker can hide an attack within the data payload itself, then the attack will go through the firewall and into the target application infrastructure. The traditional network-centric approach, which only addresses perimeter security, is no longer thought of as being effective in protecting the heart and soul of a business — its business data.

Web Services Security

Another emerging Web-enabled application is Web services. Web services comprise the sum total of application components whose functionality and interfaces are exposed to potential users through the

use of Web technology standards such as SOAP, XML, UDDI, WSDL, and HTTP. Web services are application-to-application, computer-to-computer transaction-based communications using predefined data formats in a platform- and language-neutral context. Traditional Web-enabled applications are interactive and Web-browser based. Application-level security strategies are complicated by the automated intent of Web services. Security standards are emerging and are being integrated into available security products. Application scanning and application firewall technologies are now emerging that allow for security checks against Web service data and protocols. The use of Web services to extend core business applications to external entities is expected to grow significantly in a relatively short time as businesses recognize the value of this capability. Therefore, the security issues of Web-enabled applications based on Web services will need to be checked from a perspective of automated processing between two or more security domains. Aside from the method of access, an approach similar to the Web-enabled application security strategy discussed in this chapter can be used.

The foundation of Web services is Extensible Markup Language (XML). A protocol for communicating XML-based messages, Simple Object Access Protocol (SOAP), is itself based on XML: SXML is used to create specific message formats with which two or more parties agree to comply when sending messages between applications. Defining protocols for assuring the confidentiality, integrity, and availability of Web services is a technological and business challenge that is currently being addressed by industry standards bodies. For example, IBM and Microsoft are working together to define a core set of facilities for protecting the confidentiality and integrity of an XML-based message. Their work also includes defining authentication and authorization mechanisms for creating and validating security assertions of Web service participants.

Hackers know that most business firewalls allow Web and e-mail traffic, that Web and e-mail applications are notoriously vulnerable to attack, and that many businesses focus on network perimeter security, not Web application security.

Any business connected to the Internet has a need for some level of protection beyond traditional perimeter security. Surprisingly, given the high risk of exposing e-mail and Web-enabled internal applications to wide access, many companies do not even monitor application-level events. As a result, a company may not even know that an application has been attacked.

Wide access to e-mail and Web-enabled applications is both a goal and a security risk. As a business goal, Web applications fulfill a business need to provide information and expose business logic to increase business efficiency. However, the ability to access such business architecture means that attackers have more of an opportunity to exploit known and unknown weaknesses in the architecture. Just as the decision to deploy Internet-accessible applications is a business decision, so it is that implementing application-level security must be addressed from a business management decision perspective. There are compelling and significant business management issues that can justify application-level security.

A Management Issue

Application security is both a business issue (see Exhibit 37.1) and a technical issue (see Exhibit 37.2). It is a technical issue in that more effective technology is needed to address the higher risk of exposed businesses. It is a business issue in that an ineffective security strategy means increased risk.

EXHIBIT 37.1 The SANS Institute List of Seven Management Errors that Lead to Computer Security Vulnerabilities

7. Pretend the problem will go away if they ignore it.
6. Authorize reactive, short-term fixes so problems re-emerge rapidly.
5. Fail to realize how much money their information and organizational reputations are worth.
4. Rely primarily on a firewall.
3. Fail to deal with the operational aspects of security: make a few fixes and then not allow the follow-through necessary to ensure the problems stay fixed.
2. Fail to understand the relationship of information security to the business problem: they understand physical security but do not see the consequences of poor information security.
1. Assign untrained people to maintain security and provide neither the training nor the time to make it possible to do the job.

Source: SANS Institute, http://www.sans.org/resources/errors.php.

EXHIBIT 37.2 The Open Web Application Security Project (OWASP) List of Ten Common Web Application Vulnerabilities

1. *Unvalidated Parameters:* Information from Web requests is not validated before being used by a Web application. Attackers can use these flaws to attack backside components through a Web application.
2. *Broken Access Control:* Restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these flaws to access other users' accounts, view sensitive files, or use unauthorized functions.
3. *Broken Account and Session Management:* Account credentials and session tokens are not properly protected. Attackers that can compromise passwords, keys, session cookies, or other tokens can defeat authentication restrictions and assume other users' identities.
4. *Cross-Site Scripting (XSS) Flaws:* The Web application can be used as a mechanism to transport an attack to an end user's browser. A successful attack can disclose the end user's session token, attack the local machine, or spoof content to fool the user.
5. *Buffer Overflows:* Web application components in some languages that do not properly validate input can be crashed, and, in some cases, used to take control of a process. These components can include CGI, libraries, drivers, and Web application server components.
6. *Command Injection Flaws:* Web applications pass parameters when they access external systems or the local operating system. If an attacker can embed malicious commands in these parameters, the external system might execute those commands on behalf of the Web application.
7. *Error Handling Problems:* Error conditions that occur during normal operation are not handled properly. If an attacker can cause errors to occur that the Web application does not handle, they can gain detailed system information, deny service, cause security mechanisms to fail, or crash the server.
8. *Insecure Use of Cryptography:* Web applications frequently used cryptographic functions to protect information and credentials. These functions and the code to integrate them have proven difficult to code properly, frequently resulting in weak protection.
9. *Remote Administration Flaws:* Many Web applications allow administrators to access the site using a Web interface. If these administrative functions are not very carefully protected, an attacker can gain full access to all aspects of a site.
10. *Web and Application Server Misconfiguration:* Having a strong server configuration standard is critical to a secure Web application. These servers have many configuration options that affect security and are not secure out of the box.

Source: OWASP, http://www.owasp.org/.

Part of the problem of ineffective application security is denial of the problem. In many instances, program developers and software vendors assume that because no vulnerability has been reported on an application, that it must be secure. This way of thinking is also found in business management circles with respect to already-deployed Web applications. The thinking goes: why invest in application infrastructure security when the company has had no attacks on its key business applications?

The answer to this question must be framed in terms that the audience can relate to. Business audiences think in terms of quantifiable returns on the investment. Technical audiences usually respond to things that make their jobs easier, enhance their status, or increase their value to employers. This chapter focuses on the business justification for application security.

One could surmise from the SANS list of seven management errors (Exhibit 37.1) that executive management's attitude toward recognizing and understanding the business impact of security breaches can actually influence the likelihood of a security breach. Providing business impact awareness of relevant application security vulnerabilities to business managers is a valuable role of security professionals. However, security risk management, being a continual process that must be managed, must be embraced — from the executive management level on down — throughout an organization to be effective.

The Business Risk

Competitor company B accesses company A's Web-accessible database, which contains company A's future marketing campaign strategy, by exploiting a well-known Web vulnerability. Company B, now having advanced knowledge of the upcoming marketing changes, is able to preempt company A's market opportunity for competitive advantage. A costly mistake could have been minimized or even avoided. Indeed, cost avoidance and cost reduction are two reasons to apply an application security strategy within a business.

As noted earlier, the two business applications most relied upon for Internet-connected business operations are Web-enabled applications and e-mail applications. Each of these applications relies on several related network infrastructure components. The sum total of the application itself and the network services that support the functioning of the application can be referred to as the "application infrastructure." The application infrastructure can be visualized using a three-layer model.

To isolate the various points of attack, the layered components of an application infrastructure can be reduced down to a simple model that includes a proxy layer, an internal application server layer, and a business database layer. These layers comprise the essence of an application infrastructure, although they are dependent on *network infrastructure* components, as described later.

For example, using the above model, it is possible to map the components of an e-mail infrastructure.

- *Proxy layer:* mail relay/mail exchanger/Webmail Web server
- *Internal application server layer:* internal mail server
- *Business database layer:* internal mail store/user database

An example Web application infrastructure would include:

- *Proxy layer:* web listener/web server
- *Internal application server layer:* business application server
- *Business database layer:* database server

Additionally, various network infrastructure services that are critical and common to the operation of both application architectures include Domain Name Service (DNS) servers, network routing/switch fabric (including load balancers), time servers, malicious code (including anti-virus) scanners, and protocol accelerators (e.g., SSL accelerators).

The risk to businesses that Internet-accessible applications bring is greater opportunity for attack and more points of attacks. This risk translates into lost revenue, increased costs, and lost productivity due to recovering from a security breach.

Managing Risk: Application Layer Security Primer

As discussed, Internet-accessible applications are comprised of multiple components that can be represented using a four-tier model. Isolating the functionality of Internet applications helps in understanding the various access points and potential weaknesses of a particular application architecture. However, one vulnerable component of the overall architecture can allow an attacker to undermine the entire system.

For example, if a DNS server that is relied upon by an Internet application is subverted by someone maliciously modifying the record that tells where mail for a certain domain should be routed, then it makes no difference how strong the e-mail anti-virus protection is; the attacker has undermined the entire system. This example highlights the fact that application security must be addressed using a holistic, comprehensive, and systems approach.

Many vulnerabilities in applications are caused by poor programming technique, invalid design, and lack of security awareness by software developers. However, as noted, application security is both a management and a technical issue. Therefore, the solution begins with an awareness of the issues by business managers. Business managers ultimately determine the priorities of software development teams. An example of business managers effecting a change of priority from functionality to default security is found in Microsoft. Although many are skeptical of the commitment of Microsoft to design software products with security as a priority, there have been tremendous steps made in the right direction by Microsoft management. Microsoft's secure product development program included sending all of its application developers to classes on secure coding practices, tying code security goals to performance compensation, and establishing security oversight teams to check for compliance, among other steps. Indeed, management has a clear role to play in reducing the risk of business applications.

Due to the pervasiveness of simple tools for hacking Web access, together with the proliferation of Web applications, the incidence of attack will only increase in the future. Web attacks are becoming more common than pure network-based attacks, with a resulting increase in the severity and damage done. The cost of recovering from a Web attack is growing as the sophistication of attacks increases. As the cost of an attack reaches the value of the target application, while budget dollars are decreasing, it becomes increasingly important to balance spending on security components according to highest return on asset value. Significantly, many companies were found to overspend on security tools, deploying expensive security components in areas that did not justify the expense.

To a determined attacker, the application itself yields the highest rewards. However, the Web application currently poses the greatest risk to businesses. Each application is different, with its own set of specific risks. One way to determine if enough has been done to secure a Web application is to have a Web application assessment performed on the entire Web application infrastructure, including the Web application itself. There are security consulting firms that are starting to appear in the marketplace that specialize in Web application security. These companies typically use a combination of automated security tools and hands-on experience to assess the security posture of Web applications. In some cases, security or IT groups within a company may perform their own assessment. However, expertise in this area is scarce and relatively expensive.

If employing a consulting firm to perform an application assessment, ask about the credentials and experience of its consultants who will be doing the assessment. Ask for the names and types of testing tools that will be used. Find out if the consultants provide remediation services or just a simple findings report. It may make sense to have the same consulting company that performed the assessment recommend which remediation products to use, because it may already have in-depth knowledge of the application architecture, it may have already done the technology research to save time, or it may have the necessary expertise to install and manage the security tools. With such a high impact to the business if not done properly, risk can be reduced if professionals who are specifically skilled in Web application security execute this application security strategy.

Another factor to consider is cost of fix after deployment versus cost of fix early in the development cycle. Early IT software development practice evolved with an emphasis on testing application functionality. The idea then was to reduce application total cost of ownership (TCO) by identifying bugs in the software early in the development cycle. This approach is still used today. Similarly, performing application *security* scanning early in the development cycle has been proven to significantly reduce the total cost of ownership of an application due to decreased vulnerability/exploit/fix cycles.

Organizational Standards for Software Development

One approach to integrated Web application security is to embed the application security scan function into the application quality assurance (QA) cycle. As a step in the QA process, this strategy provides an opportunity to apply a consistent security baseline against all of a company's Web applications. Once established, the same application scan could be run after any significant changes are made to the Web application throughout its life cycle to ensure that the security posture has not been altered. Existing applications can be scanned as part of a regular security assessment.

In some environments it is either not possible or very difficult to perform Web application scans due to lack of direct control of the application-hosting environment. This is the case if the application is hosted by a third-party facility, the application belongs to a business partner, or other similar situations. A strategy of implementing a security check of application data streams at the network perimeter, as a first hop-in/last hop-out application scan, can be effective. An application firewall inserted just inside the network firewall configured to intercept the Web data to and from the third party would allow a similar enforcement capability as proactive scanning of the application itself. Application firewall technology is discussed in the following section.

Technology

It is important to remember that any security solution includes the combination of people, processes, and technology. This is an important consideration, particularly in the case of application security. If a person makes a configuration mistake that leads to a security breach, the technology cannot be blamed. Similarly, if a flawed process is implemented, the technology cannot be blamed. This highlights the fact that a good practice for developing an application security policy includes mapping out the interaction of process, people, and technology. This section discusses strategies for implementing this interaction.

There are currently two classes of security technology that address application-level security: the application scanner and the application firewall/security gateway.

Web Application Scanner: More Than Just Securing a Host

The application scanner is a tool used to test applications for known and unknown vulnerabilities, unintended functionality, and poor coding practice, among other tests. The Web application scanner is usually implemented as software running on a laptop or designated desktop computer. Scanning can be done from outside the network perimeter or inside the network, just in from the Web server layer component. Web application scanning tools provide a report that lists vulnerabilities found, along with some remediation suggestions. Some of the tools provide specialized tests for particular Web application environments (e.g., IBM Websphere, BEA, Oracle). Popular products that provide Web-enabled application scanning include Sanctum's AppScan, Kavado's Scando, and SpiDynamics's WebInspect.

Application Firewall: Not Just Looking at Network Packets

The other class of application security tool is the application firewall/security gateway. A Web application firewall is inserted in the path between the user and the Web server layer of the Web application infrastructure. In most cases, this means at the network perimeter or in a DMZ "quarantine" area of a network. A Web application firewall intercepts all HTTP and HTML traffic going to and from a Web application and looks for anything that indicates improper behavior. For example, an application could detect and block users from browsing outside a site's allowed URL list, attempts to masquerade via cookie modification, buffer overflow attempts, incorrect form data entry via form field validation, and attempts to add data to a site or attempts to access restricted areas of a site via improper GET and POST methods. Most Web application firewalls include a "learning mode" that allows the device to record the proper behavior of a Web application. After a few days of "learning" proper site behavior, the Web application firewall would then dynamically create a policy and enforce "proper" site behavior based on "learned" knowledge. Additionally, Web application firewalls can be configured manually. No modification to the protected Web application itself is necessary. Multiple Web applications can be protected simultaneously by one device, or, if needed, the devices can be scaled out via load balancing and managed by the Web application firewall's central management console.

E-Mail Application-Level Firewalls: More than Just Anti-Virus

Similar in function to a Web application-level firewall, an e-mail application-level firewall can protect e-mail application infrastructures. E-mail application-level firewalls can be installed at the network perimeter, in a DMZ, or in some cases directly on the Internet. The architectural idea is that this device is the first-hop-in/last-hop-out checkpoint for e-mail application attacks. Thus positioned, the e-mail application-level firewall bears the brunt of an attack leaving the e-mail infrastructure intact and operational during the attack. The technical value of an e-mail application-level firewall lies in that it buys time to allow for patching or updating the target e-mail infrastructure component. Additionally, a consistent e-mail security posture can be maintained using the e-mail application-level firewall as an additional security layer. The e-mail application-level firewall inspects e-mail protocols and e-mail messages for attack attempts and enforces policy via mechanisms such blocking, logging, or alerting on detection. Although there are a few vendors of the firewalls themselves, a significant market-leading, single-purpose-

built e-mail application-level security scanning tool is yet to emerge. Current testing tools for e-mail application infrastructures include a hodgepodge of open source and commercial network vulnerability assessment tools.

An emerging and highly segmented market, e-mail application-level firewalls provide some level of hardening for self-protection and multiple controls against a wide variety of threats to the e-mail infrastructure. The threat profile of e-mail application infrastructures includes redirecting mail via DNS poisoning, malicious code attacks to disrupt or corrupt e-mail message integrity, large volumes of unsolicited e-mails or server connections aimed at reducing the availability of e-mail service — mail bombs and spam attacks. E-mail application-level firewalls proxy e-mail connections between external e-mail servers and internal e-mail servers, never allowing direct connection from the outside. The e-mail application-level firewall can also enforce a message retention policy by archiving messages to archiving hosts for later retrieval as needed.

E-mail application-level firewalls are a different class of device than the popular software-based mail server add-on products. Mail server add-on products typically provide specific mail security functionality, such as content filtering, anti-virus, and anti-spam — similar to e-mail application-level firewalls. However, they are implemented on the actual mail server itself. Software-based mail server security products generally do not have the capability to examine a message before it enters the e-mail infrastructure and they commonly introduce e-mail processing performance degradation. Scalability becomes an issue in the larger, more complex e-mail architectures. A special case involves Web mail: Web browser accessible e-mail systems. There are two classes of Web mail from a protection strategy perspective that should be considered: Web mail service provided by a company to its community of users, and external consumer-oriented Web mail services such as Yahoo, MSN, and AOL accessed from inside a company network. When planning a strategy in this regard, remember that all Web mail should be considered hostile until proven otherwise. This means that the e-mail application-level firewall should be capable of inspecting Web mail traffic for protocol and syntax attacks, similar in result to Internet mail protocols (SMTP, POP3, IMAP4) and syntax checking.

As mentioned, such devices are usually best implemented as a security appliance. A security appliance approach provides specific functionality implemented on optimized hardware, with the results including higher performance at a lower cost, decreased ongoing maintenance costs, and scalability efficiency. The software-based approach means that the customer must assume increased costs of integrating hardware, operating system, and application. Additionally, the host operating system should be hardened. Such hardening of the operating system requires expertise and ongoing diligence in managing the host operating system, applying patches, and hardware upgrades.

Vendors offering products in the Web application firewall market segment include Sanctum (AppShield), Kavado (InterDo), and Teros\Stratum8 (APS 100). E-mail application firewall vendors include CipherTrust (IronMail) and Borderware (MxTreme).

The Bottom Line: Balancing Security Protection against Assets Being Protected

The traditional network firewall has a respected and necessary place in most enterprise security environments. It provides a first line of defense, access control, and a security control point. However, many businesses open bi-directional access to critical business applications, such as e-mail and Web-enabled applications that have historically been vulnerable to numerous attacks. Access to these applications is provided by opening up the network firewall, port 80 for Web and port 25 for e-mail. Hackers are now predominantly sneaking in through these open ports to run application-level exploits against the application infrastructure, those components that form the essence of a Web-enabled or e-mail application. E-mail and many Web-enabled business applications are core, mission-critical assets that, if disabled, could cause significant damage and prove very costly to recover from — if even possible. If there is more

risk of attacks at the higher-value assets, e-mail and Web-enabled applications, then it makes sense to balance security spending appropriately to protect these critical applications.

Conclusion

This chapter discussed the technology available at the time of writing. Although current application security technology provides some level of protection, there is much room for improvement. Network security technology components — such as network-based firewalls, VPNs, and intrusion detection and response — continue to make up the essential first line of defense; however, the threat horizon has changed. This change in attack vectors requires a reorientation toward the emerging sources and targets of attack — attackers coming through application ports to target application vulnerabilities of core business information systems.

References

SANS Institute, http://www.sans.org/resources/errors.php
OWASP, http://www.owasp.org/

38

Security of Communication Protocols and Services

William Hugh Murray, CISSP

The information security manager is confronted with a wide variety of communications protocols and services. At one level, the manager would like to be able to ignore how the information gets from one place to another; he would like to be able to *assume* security. At another, he understands that he has only limited control over how the information moves; because the user may be able to influence the choice of path, the manager prefers not to rely upon it. However, that being said, the manager also knows that there are differences in the security properties of the various protocols and services that he may otherwise find useful.

This chapter describes the popular protocols and services, discusses their intended uses and applications, and describes their security properties and characteristics. It compares and contrasts similar protocols and services, makes recommendations for their use, and also recommends compensating controls or alternatives for increasing security.

Introduction

For the past century, we have trusted the dial-switched voice-analog network. It was operated by one of the most trusted enterprises in the history of the world. It was connection-switched and point-to-point. While there was some eavesdropping, most of it was initiated by law enforcement and was, for the most part, legitimate. While a few of us carefully considered what we would say, most of us used the telephone automatically and without worrying about being overheard. Similarly, we were able to recognize most of the people who called us; we trusted the millions of copies of the printed directories; and we trusted the network to connect us only to the number we dialed. While it is not completely justified, we have transferred much of that automatic trust to the modern digital network and even to the Internet.

All other things being equal, the information security manager would like to be able to ignore how information moves from one place to another. He would like to be able to assume that he can put it into a pipe at point A and have it come out reliably only at point B. Of course, in the real world of the modern integrated network, this is not the case. In this world the traffic is vulnerable to eavesdropping, misdirection, interference, contamination, alteration, and even total loss.

On the other hand, relatively little of this happens; the vast majority of information is delivered when and how it is intended and without any compromise. This happens in part despite the way the information is moved and in part because of how it is moved. The various protocols and services have different security properties and qualities. Some provide error detection, corrective action such as retransmission, error correction, guaranteed delivery, and even information hiding.

The different levels of service exist because they have different costs and performance. They exist because different traffic, applications, and environments have different requirements. For example, the transfer of a program file has a requirement for bit-for-bit integrity; in some cases, if you lose a bit, it is as bad as losing the whole file. On the other hand, a few seconds, or even tens of seconds, of delay in the transfer of the file may have little impact. However, if one is moving voice traffic, the loss of tens of bits may be perfectly acceptable, while delay in seconds is intolerable. These costs must be balanced against the requirements of the application and the environment.

While the balance between performance and cost is often struck without regard to security, the reality is that there are security differences. The balance between performance, cost, and security is the province of the information security manager. Therefore, he needs to understand the properties and characteristics of the protocols so he can make the necessary trade-offs or evaluate those that have already been made.

Finally, all protocols have limitations and many have fundamental vulnerabilities. Implementations of protocols can compensate for such vulnerabilities only in part. Implementers may be faced with hard design choices, and they may make errors resulting in implementation-induced vulnerabilities. The manager must understand these so he will know when and how to compensate.

Protocols

A protocol is an agreed-upon set of rules or conventions for communicating between two or more parties. "Hello" and "goodbye" for beginning and ending voice phone calls are examples of a simple protocol. A slightly more sophisticated protocol might include lines that begin with tags, like "This is (name) calling."

Protocols are to codes as sentences and paragraphs are to words. In a protocol, the parties may agree to addressing, codes, format, packet size, speed, message order, error detection and correction, acknowledgments, key exchange, and other things.

This section deals with a number of common protocols. It describes their intended use or application, characteristics, design choices, and limitations.

Internet Protocol

The Internet Protocol, IP, is a primitive and application-independent protocol for addressing and routing packets of data within a network. It is the "IP" in TCP/IP, the protocol suite that is used in and defines the Internet. It is intended for use in a relatively flat, mesh, broadcast, connectionless, packet-switched net like the Internet.

IP is analogous to a postcard in the 18th century. The sender wrote the message on one side of the card and the address and return address on the other. He then gave it to someone who was going in the general direction of the intended recipient. The message was not confidential; everyone who handled it could read it and might even make an undetected change to it.

IP is a "best efforts" protocol; it does not guarantee message delivery nor provide any evidence as to whether or not the message was delivered. It is unchecked; the receiver does not know whether or not he received the entire intended message or whether or not it is correct. The addresses are unreliable; the sender cannot be sure that the message will go only where he intends or even when he intends. The receiver cannot be sure that the message came from the address specified as the return address in the packet.

The protocol does not provide any checking or hiding. If the application requires these, they must be implied or specified someplace else, usually in a higher (i.e., closer to the application) protocol layer.

IP specifies the addresses of the sending or receiving hardware device;* but if that device supports multiple applications, IP does not specify which of those it is intended for.

*There is a convention of referring to all network addressable devices as "hosts." Such usage in other documents equates to the use of device or addressable device here. IPv6 defines "host."

EXHIBIT 38.1 IP Network Address Formats

Network Class	Description	Address Class	Network Address	Device Address
A	National	0 in bit 0	1–7	8–31
B	Enterprise	10 in bits 0–1	2–15	16–31
C	LAN	110 in 0–2	3–23	24–31
D	Multicast	1110 in 0–3	4–31	
E	Reserved	1111 in 0–3		

IP uses 32-bit addresses. However, the use or meaning of the bits within the address depends upon the size and use of the network. Addresses are divided into five classes. Each class represents a different design choice between the number of networks and the number of addressable devices within the class. Class A addresses are used for very large networks where the number of such networks is expected to be low but the number of addressable devices is expected to be very high. Class A addresses are used for nation states and other very large domains such as .mil, .gov, and .com. As shown in Exhibit 38.1, a zero in bit position 0 of an address specifies it as a class A address. Positions 1 through 7 are used to specify the network, and positions 8 through 31 are used to specify devices within the network. Class C is used for networks where the possible number of networks is expected to be high but the number of addressable devices in each net is less than 128. Thus, in general, class B is used for enterprises, states, provinces, or municipalities, and class C is used for LANs. Class D is used for multicasting, and Class E is reserved for future uses.

You will often see IP addresses written as nnn.nnn.nnn.nnn.

While security is certainly not IP's long suit, it is responsible for much of the success of the Internet. It is fast and simple. In practice, the security limitations of IP simply do not matter much. Applications rely upon higher-level protocols for security.

Internet Protocol v6.0 (IPng)

IPv6 or "next generation" is a backwardly compatible new version of IP. It is intended to permit the Internet to grow both in terms of the number of addressable devices, particularly class A addresses, and in quantity of traffic. It expands the address to 128 bits, simplifies the format header, improves the support for extensions and options, adds a "quality-of-service" capability, and adds address authentication and message confidentiality and integrity. IPv6 also formalizes the concepts of packet, node, router, host, link, and neighbors that were only loosely defined in v4.

In other words, IPng addresses most of the limitations of IP, specifically including the security limitations. It provides for the use of encryption to ensure that information goes only where it is intended to go. This is called secure-IP. Secure-IP may be used for point-to-point security across an arbitrary network. More often, it is used to carve virtual private networks (VPNs) or secure virtual networks (SVNs)* out of such arbitrary networks.

Many of the implementations of secure-IP are still proprietary and do not guarantee interoperability with all other such implementations.

User Datagram Protocol (UDP)

UDP is similar to IP in that it is connectionless and offers "best effort" delivery service, and it is similar to TCP in that it is both checked and application specific.

*VPN is used here to refer to the use of encryption to connect private networks across the public network, gateway-to-gateway. SVN is used to refer to the use of encryption to talk securely, end-to-end, across arbitrary networks. While the term VPN is sometimes used to describe both applications, different implementations of secure-IP may be required for the two applications.

EXHIBIT 38.2 UDP Datagram

Bit Positions	Usage
0–15	Source Port Address
16–31	Destination Port Address
32–47	Message Length (n)
48–63	Checksum
64–n	Data

Exhibit 38.2 shows the format of the UDP datagram. Unless the UDP source port is on the same device as the destination port, the UDP packet will be encapsulated in an IP packet. The IP address will specify the physical device, while the UDP address will specify the logical port or application on the device.

UDP implements the abstraction of "port," a named logical connection or interface to a specific application or service within a device. Ports are identified by a positive integer. Port identity is local to a device, that is, the use or meaning of port number is not global. A given port number can refer to any application that the sender and receiver agree upon. However, by convention and repeated use, certain port numbers have become identified with certain applications. Exhibit 38.3 lists examples of some of these conventional port assignments.

Transmission Control Protocol (TCP)

TCP is a sophisticated composition of IP that compensates for many of its limitations. It is a connection-oriented protocol that enables two applications to exchange streams of data synchronously and simultaneously in both directions. It guarantees both the delivery and order of the packets. Because packets are given a sequence number, missing packets will be detected, and packets can be delivered in the same order in which they were sent; lost packets can be automatically resent. TCP also adapts to the latency of the network. It uses control flags to enable the receiver to automatically slow the sender so as not to overflow the buffers of the receiver.

TCP does not make the origin address reliable. The sequence number feature of TCP resists address spoofing. However, it does not make it impossible. Instances of attackers pretending to be trusted nodes have been reported to have toolkits that encapsulate the necessary work and special knowledge to implement such attacks.

Like many packet-switched protocols, TCP uses path diversity. This means some of the meaning of the traffic may not be available to an eavesdropper. However, eavesdropping is still possible. For example, user identifiers and passphrases usually move in the same packet. "Password grabber" programs have been detected in the network. These programs simply store the first 256 or 512 bits of packets on the assumption that many will contain passwords.

Finally, like most stateful protocols, some TCP implementations are vulnerable to denial-of-service attacks. One such attack is called *SYN flooding*. Requests for sessions, SYN flags, are sent to the target, but the acknowledgments are ignored. The target allocates memory to these requests and is overwhelmed.

EXHIBIT 38.3 Sample UDP Ports

Port Number	Application	Description
23	Telnet	
53	DNS	Domain name service
43		Whois
69	TFTP	Trivial file transfer service
80	HTTP	Web service
119	Net News	
137		NetBIOS name service
138		NetBIOS datagrams
139		NetBIOS session data

Telnet

The Telnet protocol describes how commands and data are passed from one machine on the network to another over a TCP/IP connection. It is described in RFC 855. It is used to make a terminal or printer on one machine and an operating system or application on another appear to be local to each other. The user invokes the Telnet client by entering its name or clicking its icon on his local system and giving the name or address and port number of the system or application that he wishes to use. The Telnet client must listen to the keyboard and send the characters entered by the user across the TCP connection to the server. It listens to the TCP connection and displays the traffic on the user's terminal screen. The client and server use an escape sequence to distinguish between user data and their communication with each other.

The Telnet service is a frequent target of attack. By default, the Telnet service listens for login requests on port 23. Connecting this port to the public network can make the system and the network vulnerable to attack. When connected to the public net, this port should expect strong authentication or accept only encrypted traffic.

File Transfer Protocol (FTP)

FTP is the protocol used on the Internet for transferring files between two systems. It divides a file into IP packets for sending it across the Internet. The object of the transfer is a file. The protocol provides automatic checking and retransmission to provide for bit-for-bit integrity. (See section titled Services below.)

Serial Line Internet Protocol (SLIP)

SLIP is a protocol for sending IP packets over a serial line connection. It is described in RFC 1055. SLIP is often used to extend the path from an IP-addressable device, like a router at an ISP, across a serial connection, a dial connection (e.g., a dial connection) to a non-IP device (e.g., a serial port on a PC). It is a mechanism for attaching non-IP devices to an IP network.

SLIP encapsulates the IP packet and bits in the code used on the serial line. In the process, the packet may gain some redundancy and error correction. However, the protocol itself does not provide any error detection or correction. This means that errors may not be detected until the traffic gets to a higher layer. Because SLIP is usually used over relatively slow (56 Kb) lines, this may make error correction at that layer expensive. On the other hand, the signaling over modern modems is fairly robust. Similarly, SLIP traffic may gain some compression from devices (e.g., modems) in the path but does not provide any compression of its own.

Because the serial line has only two endpoints, the protocol does not contain any address information; that is, the addresses are implicit. However, this limits the connection to one application; any distinctions in the intended use of the line must be handled at a higher layer.

Because SLIP is used on point-to-point connections, it may be slightly less vulnerable to eavesdropping than a shared-media connection like Ethernet. However, because it is closer to the endpoint, the data may be more meaningful. This observation also applies to PPP below.

Point-to-Point Protocol (PPP)

PPP is used for applications and environments similar to those for SLIP but is more sophisticated. It is described in RFC 1661, July 1994. It is *the* Internet standard for transmission of IP packets over serial lines. It is more robust than SLIP and provides error-detection features. It supports both asynchronous and synchronous lines and is intended for simple links that deliver packets between two peers. It enables the transmission of multiple network-layer protocols (e.g., IP, IPX, SPX) simultaneously over a single link. For example, a PC might run a browser, a Notes client, and an e-mail client over a single link to the network.

To facilitate all this, PPP has a Link Control Protocol (LCP) to negotiate encapsulation formats, format options, and limits on packet format.

Optionally, a PPP node can require that its partner authenticate itself using CHAP or PAP. This authentication takes place after the link is set up and before any traffic can flow. (See CHAP and PAP below.)

HyperText Transfer Protocol (HTTP)

HTTP is used to move data objects, called pages, between client applications, called browsers, running on one machine, and server applications, usually on another. HTTP is the protocol that is used on and that defines the World Wide Web. The pages moved by HTTP are compound data objects composed of other data and objects. Pages are specified in a language called HyperText Markup Language, or HTML. HTML specifies the appearance of the page and provides for pages to be associated with one another by cross-references called hyperlinks.

The fundamental assumption of HTTP is that the pages are public and that no data-hiding or address reliability is necessary. However, because many electronic commerce applications are done on the World Wide Web, other protocols, described below, have been defined and implemented.

Security Protocols

Most of the traffic that moves in the primitive TCP/IP protocols is public; that is, none of the value of the data derives from its confidentiality. Therefore, the fact that the protocols do not provide any data-hiding does not hurt anything. The protocols do not add any security, but the data does not need it. However, there is some traffic that is sensitive to disclosure and which does require more security than the primitive protocols provide. The absolute amount of this traffic is clearly growing, and its proportion may be growing also. In most cases, the necessary hiding of this data is done in alternate or higher-level protocols.

A number of these secure protocols have been defined and are rapidly being implemented and deployed. This section describes some of those protocols.

Secure Socket Layer (SSL)

Arguably, the most widely used secure protocol is SSL. It is intended for use in client–server applications in general. More specifically, it is widely used between browsers and Web servers on the WWW. It uses a hybrid of symmetric and asymmetric key cryptography, in which a symmetric algorithm is used to hide the traffic and an asymmetric one, RSA, is used to negotiate the symmetric keys.

SSL is a session-oriented protocol; that is, it is used to establish a secure connection between the client and the server that lasts for the life of the session or until terminated by the application.

SSL comes in two flavors and a number of variations. At the moment, the most widely used of the two flavors is *one-way SSL*. In this implementation, the server side has a private key, a corresponding public key, and a certificate for that key-pair. The server offers its public key to the client. After reconciling the certificate to satisfy itself as to the identity of the server, the client uses the public key to securely negotiate a session key with the server. Once the session key is in use, both the client and the server can be confident that only the other can see the traffic.

The client side has a public key for the key-pair that was used to sign the certificate and can use this key to verify the bind between the key-pair and the identity of the server. Thus, the one-way protocol provides for the authentication of the server to the client but not the other way around. If the server cares about the identity of the client, it must use the secure session to collect evidence about the identity of the client. This evidence is normally in the form of a user identifier and a passphrase or similar, previously shared, secret.

The other flavor of SSL is *two-way SSL*. In this implementation both the client and the server know the public key of the other and have a certificate for this key. In most instances the client's certificate is issued by the server, while the server's certificate was issued by a mutually trusted third party.

Secure-HTTP (S-HTTP)

S-HTTP is a secure version of HTTP designed to move individual pages securely on the World Wide Web. It is page oriented as contrasted to SSL, which is connection or session oriented. Most browsers (thin clients) that implement SSL also implement S-HTTP, may share key-management code, and may be used in ways that are not readily distinguishable to the end user. In other applications, S-HTTP gets the nod where very high performance is required and where there is limited need to save state between the client and the server.

Secure File Transfer Protocol (S-FTP)

Most of the applications of the primitive File Transfer Protocol are used to transfer public files in private networks. Much of it is characterized as "anonymous;" that is, one end of the connection may not even recognize the other. However, as the net spreads, FTP is increasingly used to move private data in public networks.

S-FTP adds encryption to FTP to add data-hiding to the integrity checking provided in the base protocol.

Secure Electronic Transaction (SET)

SET is a special protocol developed by the credit card companies and vendors and intended for use in multi-party financial transactions like credit card transactions across the Internet. It provides not only for hiding credit card numbers as they cross the network, but also for hiding them from some of the parties to the transaction and for protecting against replay.

One of the limitations of SSL when used for credit card numbers is that the merchant must become party to the entire credit card number and must make a record of it to use in the case of later disputes. This creates a vulnerability to the disclosure and reuse of the credit card number. SET uses public key cryptography to guarantee the merchant that he will be paid without his having to know or protect the credit card number.

Point-to-Point Tunneling* Protocol (PPTP)

PPTP is a protocol (from the PPTP Forum) for hiding the information in IP packets, including the addresses. It is used to connect (portable computer) clients across the dial-switched point-to-point network to the Internet and then to a (MS) gateway server to a private (enterprise) network or to (MS) servers on such a network. As its name implies, it is a point-to-point protocol. It is useful for implementing end-to-end secure virtual networks (SVNs) but less so for implementing any-gateway-to-any-gateway virtual private networks (VPNs).

It includes the ability to:

- Query the status of Comm Servers
- Provide in-band management
- Allocate channels and place outgoing calls
- Notify server on incoming calls
- Transmit and receive user data with flow control in both directions
- Notify server on disconnected calls

One major advantage of PPTP is that it is included in MS 32-bit operating systems. (At this writing, the client-side software is included on 32-bit MS Windows operating systems Dial Up Networking [rel.

*Tunneling is a form of encapsulation in which the encrypted package, the passenger, is encapsulated inside a datagram of the carrier protocol.

1.2 and 1.3]. The server-side software is included in the NT Server operating system. See L2TP below.) A limitation of PPTP, when compared to secure-IP or SSL, is that it does not provide authentication of the endpoints. That is, the nodes know that other nodes cannot see the data passing between but must use other mechanisms to authenticate addresses or user identities.

Layer 2 Forwarding (L2F)

L2F is another mechanism for hiding information on the Internet. The encryption is provided from the point where the dial-switched point-to-point network connects the Internet service provider (ISP) to the gateway on the private network. The advantage is that no additional software is required on the client computer; the disadvantage is that the data is protected only on the Internet and not on the dial-switched network.

L2F is a router-to-router protocol used to protect data from acquisition by an ISP, across the public digital packet-switched network (Internet) to receipt by a private network. It is used by the ISP to provide data-hiding servers to its clients. Because the protocol is implemented in the routers (Cisco), its details and management are hidden from the end users.

Layer 2 Tunneling Protocol (L2TP)

L2TP is a proposal by MS and Cisco to provide a client-to-gateway data-hiding facility that can be operated by the ISP. It responds to the limitations of PPTP (must be operated by the owner of the gateway) and L2F (does not protect data on the dial-switched point-to-point net). Such a solution could protect the data on both parts of the public network but as a service provided by the ISP rather than by the operator of the private network.

Secure Internet Protocol (Secure-IP or IPSec)

IPSec is a set of protocols to provide for end-to-end encryption of the IP packets. It is being developed by the Internet Engineering Task Force (IETF). It is to be used to bind endpoints to one another and to implement VPNs and SVNs.

Internet Security Association Key Management Protocol (ISAKMP)

ISAKMP is a proposal for a public-key certificate-based key-management protocol for use with IPSec. Because in order to establish a secure session the user will have to have both a certificate and the corresponding key and because the session will not be vulnerable to replay or eavesdropping, ISAKMP provides "strong authentication." What is more, because the same mechanism can be used for encryption as for authentication, it provides economy of administration.

Password Authentication Protocol (PAP)

As noted above, PPP provides for the parties to identify and authenticate each other. One of the protocols for doing this is PAP. (See also CHAP below). PAP works very much like traditional login using a shared secret. A sends a prompt or a request for authentication to B, and B responds with an identifier and a shared secret. If the pair of values meets A's expectation, then A acknowledges B.

This protocol is vulnerable to a replay attack. It is also vulnerable to abuse of B's identity by a privileged user of A.

Challenge Handshake Authentication Protocol (CHAP)

CHAP is a standard challenge–response peer-to-peer authentication mechanism. System A chooses a random number and passes it to B. B encrypts this challenge under a secret shared with A and returns it to A. A also computes the value of the challenge encrypted under the shared secret and compares this value to the value returned by B. If this response meets A's expectation, then A acknowledges B.

Many implementations of PPP/CHAP provide that the remote party be periodically reauthenticated by sending a new challenge. This resists any attempt at "session stealing."

Services

Telnet

File Transfer

FTP is the name of a protocol, but it is also the name of a service that uses the protocol to deliver files. The service is symmetric in that either the server or the client can initiate a transfer in either direction, either can get a file or send a file, either can do a get or a put. The client may itself be a server. The server may or may not recognize its user, and may or may not restrict access to the available files.

Where the server does restrict access to the available files, it usually does that through the use of the control facilities of the underlying file system. If the file server is built upon the UNIX operating system and file system or the Windows operating systems, then it will use the rules-based file access controls of the file system. If the server is built upon the NT operating system, then it will use the object-oriented controls of the NT file system. If the file service is built on MVS, and yes that does happen, then it is the optional access control facility of MVS that will be used.

Secure Shell (SSH 2)

Secure Shell is a UNIX-to-UNIX client-server program that uses strong cryptography for protecting all transmitted data, including passwords, binary files, and administrative commands between systems on a network. One can think of it as a client-server command processor or shell. While it is used primarily for system management, it should not be limited to this application.

SSH2 implements Secure-IP and ISAKMP at the application layer, as contrasted to the network layer, to provide a secure network computing environment. It provides node identification and authentication, node-to-node encryption, and secure command and file transfer. It compensates for most of the protocol limitations noted above. It is now preferred to and used in place of more limited or application-specific protocols or implementations such as Secure-FTP.

Conclusions

Courtney's first law says that nothing useful can be said about the security of a mechanism except in the context of an application and an environment. Of course, the converse of that law says that, in such a context, one can say quite a great deal.

The Internet is an open, not to say hostile, environment in which most everything is permitted. It is defined almost exclusively by its addresses and addressing schema and by the protocols that are honored in it. Little else is reliable.

Nonetheless, most sensitive applications can be done there as long as one understands the properties and limitations of those protocols and carefully chooses among them. We have seen that there are a large number of protocols defined and implemented on the Internet. No small number of them are fully adequate for all applications. On the other hand, the loss in performance, flexibility, generality, and function in order to use those that are secure for the intended application and environment is small. What is more, as the cost of performance falls, the differences become even less significant.

The information security manager must understand the needs of his applications, and know the tools, protocols, and what is possible in terms of security. Then he must choose and apply those protocols and implementations carefully.

Security Management for the World Wide Web

Lynda L. McGhie
Phillip Q. Maier

Companies continue to flock to the Internet in ever-increasing numbers, despite the fact that the overall and underlying environment is not secure. To further complicate the matter, vendors, standards bodies, security organizations, and practitioners cannot agree on a standard, compliant, and technically available approach. As a group of investors concerned with the success of the Internet for business purposes, it is critical that we pull our collective resources and work together to quickly establish and support interoperable security standards; open security interfaces to existing security products and security control mechanisms within other program products; and hardware and software solutions within heterogeneous operating systems which will facilitate smooth transitions.

Interfaces and teaming relationships to further this goal include computer and network security and information security professional associations (CSI, ISSA, NCSA), professional technical and engineering organizations (I/EEE, IETF), vendor and product user groups, government and standards bodies, seminars and conferences, training companies/institutes (MIS), and informal networking among practitioners.

Having the tools and solutions available within the marketplace is a beginning, but we also need strategies and migration paths to accommodate and integrate Internet, intranet, and World Wide Web (WWW) technologies into our existing IT infrastructure. While there are always emerging challenges, introduction of newer technologies, and customers with challenging and perplexing problems to solve, this approach should enable us to maximize the effectiveness of our existing security investments, while bridging the gap to the long awaited and always sought after perfect solution!

Security solutions are slowly emerging, but interoperability, universally accepted security standards, application programming interfaces (APIs) for security, vendor support and cooperation, and multiplatform security products are still problematic. Where there are products and solutions, they tend to have niche applicability, be vendor-centric or only address one of a larger set of security problems and requirements. For the most part, no single vendor or even software/vendor consortium has addressed the overall security problem within "open" systems and public networks. This indicates that the problem is very large, and that we are years away from solving today's problem, not to mention tomorrow's.

This chapter establishes and supports the need for an underlying baseline security framework that will enable companies to successfully evolve to doing business over the Internet and using internal intranet- and World Wide Web-based technologies most effectively within their own corporate computing and networking infrastructures. It presents a solution set that exploits existing skills, resources, and security implementations.

By acknowledging today's challenges, bench-marking today's requirements, and understanding our "as is condition" accordingly, we as security practitioners can best plan for security in the twenty-first century. Added benefits adjacent to this strategy will hopefully include a more cost-effective and seamless integration of security policies, security architectures, security control mechanisms, and security management processes to support this environment.

For most companies, the transition to "open" systems technologies is still in progress and most of us are somewhere in the process of converting mainframe applications and systems to distributed network-centric client-server infrastructures. Nevertheless, we are continually challenged to provide a secure environment today, tomorrow, and in the future, including smooth transitions from one generation to another. This chapter considers a phased integration methodology that initially focuses on the update of corporate policies and procedures, including most security policies and procedures; secondly, enhances existing distributed security architectures to accommodate the use of the Internet, intranet, and WWW technologies; thirdly, devises a security implementation plan that incorporates the use of new and emerging security products and techniques; and finally, addresses security management and infrastructure support requirements to tie it all together.

It is important to keep in mind, as with any new and emerging technology, Internet, intranet, and WWW technologies do not necessarily bring new and unique security concerns, risks, and vulnerabilities, but rather introduce new problems, challenges and approaches within our existing security infrastructure.

Security requirements, goals, and objectives remain the same, while the application of security, control mechanisms, and solution sets are different and require the involvement and cooperation of multidisciplined technical and functional area teams. As in any distributed environment, there are more players, and it is more difficult to find or interpret the overall requirements or even talk to anyone who sees or understands the big picture. More people are involved than ever before, emphasizing the need to communicate both strategic and tactical security plans broadly and effectively throughout the entire enterprise. The security challenges and the resultant problems become larger and more complex in this environment. Management must be kept up-to-date and thoroughly understand overall risk to the corporation's information assets with the implementation or decisions to implement new technologies. They must also understand, fund, and support the influx of resources required to manage the security environment.

As with any new and emerging technology, security should be addressed early in terms of understanding the requirements, participating in the evaluation of products and related technologies, and finally in the engineering, design, and implementation of new applications and systems. Security should also be considered during all phases of the systems development life cycle. This is nothing new, and many of us have learned this lesson painfully over the years as we have tried to retrofit security solutions as an adjunct to the implementation of some large and complex system. Another important point to consider throughout the integration of new technologies, is "technology does not drive or dictate security policies, but the existing and established security policies drive the application of new technologies." This point must be made to management, customers, and supporting IT personnel.

For most of us, the WWW will be one of the most universal and influential trends impacting our internal enterprise and its computing and networking support structure. It will widely influence our decisions to extend our internal business processes out to the Internet and beyond. It will enable us to use the same user interface, the same critical systems and applications, work towards one single original source of data, and continue to address the age-old problem: how can I reach the largest number of users at the lowest cost possible?"

THE PATH TO INTERNET/BROWSER TECHNOLOGIES

Everyone is aware of the staggering statistics relative to the burgeoning growth of the Internet over the last decade. The use of the WWW can even top that growth, causing the traffic on the Internet to double every six months. With five internal Web servers being deployed for every one external Web server, the rise of the intranet is also more than just hype. Companies are predominately using the Web technologies on the intranet to share

information and documents. Future application possibilities are basically any enterprise-wide application such as education and training; corporate policies and procedures; human resources applications such as a resume, job posting, etc.; and company information. External Web applications include marketing and sales.

For the purpose of this discussion, we can generally think of the Internet in three evolutionary phases. While each succeeding phase has brought with it more utility and the availability of a wealth of electronic and automated resources, each phase has also exponentially increased the risk to our internal networks and computing environments.

Phase I, the early days, is characterized by a limited use of the Internet, due in the most part to its complexity and universal accessibility. The user interface was anything but user friendly, typically limited to the use of complex UNIX-based commands via line mode. Security by obscurity was definitely a popular and acceptable way of addressing security in those early days, as security organizations and MIS management convinced themselves that the potential risks were confined to small user populations centered around homogeneous computing and networking environments. Most companies were not externally connected in those days, and certainly not to the Internet.

Phase II is characterized by the introduction of the first versions of data base search engines, including Gopher and Wide Area Information System (WAIS). These tools were mostly used in the government and university environments and were not well known nor generally proliferated in the commercial sector.

Phase III brings us up to today's environment, where Internet browsers are relatively inexpensive, readily available, easy to install, easy to use through GUI frontends and interfaces, interoperable across heterogeneous platforms, and ubiquitous in terms of information access.

The growing popularity of the Internet and the introduction of the "Internet" should not come as a surprise to corporate executives who are generally well read on such issues and tied into major information technology (IT) vendors and consultants. However, quite frequently companies continue to select one of two choices when considering the implementation of WWW and Internet technologies. Some companies, who are more technically astute and competitive, have jumped in totally and are exploiting Internet technologies, electronic commerce, and the use of the Web. Others, of a more conservative nature and more technically inexperienced, continue to maintain a hard-line policy on external connectivity, which basically continues to say "NO."

Internet technologies offer great potential for cost savings over existing technologies, representing huge investments over the years in terms of

revenue and resources now supporting corporate information infrastructures and contributing to the business imperatives of those enterprises. Internet-based applications provide a standard communications interface and protocol suite ensuring interoperability and access to the organization's heterogeneous data and information resources. Most WWW browsers run on all systems and provide a common user interface and ease of use to a wide range of corporate employees.

Benefits derived from the development of WWW-based applications for internal and external use can be categorized by the cost savings related to deployment, generally requiring very little support or end-user training. The browser software is typically free, bundled in vendor product suites, or very affordable. Access to information, as previously stated, is ubiquitous and fairly straightforward.

Use of internal WWW applications can change the very way organizations interact and share information. When established and maintained properly, an internal WWW application can enable everyone on the internal network to share information resources, update common use applications, receive education and training, and keep in touch with colleagues at their home base, from remote locations, or on the road.

INTERNET/WWW SECURITY OBJECTIVES

As mentioned earlier, security requirements do not change with the introduction and use of these technologies, but the emphasis on where security is placed and how it is implemented does change. The company's Internet, intranet, and WWW security strategies should address the following objectives, in combination or in prioritized sequence, depending on security and access requirements, company philosophy, the relative sensitivity of the company's information resources, and the business imperative for using these technologies.

- Ensure that Internet- and WWW-based application and the resultant access to information resources are protected, and that there is a cost-effective and user-friendly way to maintain and manage the underlying security components over time as new technology evolves and security solutions mature in response.
- Information assets should be protected against unauthorized usage and destruction. Communication paths should be encrypted as well as transmitted information that is broadcast over public networks.
- Receipt of information from external sources should be decrypted and authenticated. Internet- and WWW-based applications, WWW pages, directories, discussion groups, and data bases should all be secured using access control mechanisms.
- Security administration and overall support should accommodate a combination of centralized and decentralized management.

- User privileges should be linked to resources, with privileges to those resources managed and distributed through directory services.
- Mail and real-time communications should also be consistently protected. Encryption key management systems should be easy to administer, compliant with existing security architectures, compatible with existing security strategies and tactical plans, and secure to manage and administer.
- New security policies, security architectures, and control mechanisms should evolve to accommodate this new technology; not change in principle or design.

Continue to use risk management methodologies as a baseline for deciding how many of the new Internet, intranet, and WWW technologies to use and how to integrate them into the existing Information Security Distributed Architecture. As always, ensure that the optimum balance between access to information and protection of information is achieved during all phases of the development, integration, implementation, and operational support life cycle.

INTERNET AND WWW SECURITY POLICIES AND PROCEDURES

Having said all of this, it is clear that we need new and different policies, or minimally, an enhancement or refreshing of current policies supporting more traditional means of sharing, accessing, storing, and transmitting information. In general, high-level security philosophies, policies, and procedures should not change. In other words, who is responsible for what (the fundamental purpose of most high-level security policies) does not change. These policies are fundamentally directed at corporate management, process, application and system owners, functional area management, and those tasked with the implementation and support of the overall IT environment. There should be minimal changes to these policies, perhaps only adding the Internet and WWW terminology.

Other high-level corporate policies must also be modified, such as the use of corporate assets, responsibility for sharing and protecting corporate information, etc. The second-level corporate policies, usually more procedure oriented typically addressing more of the "how," should be more closely scrutinized and may change the most when addressing the use of the Internet, intranet, and Web technologies for corporate business purposes. New classifications and categories of information may need to be established and new labeling mechanisms denoting a category of information that cannot be displayed on the Internet or new meanings to "all allow" or "public" data. The term "public," for instance, when used internally, usually means anyone authorized to use internal systems. In most companies, access to internal networks, computing systems, and information is

severely restricted and "public" would not mean unauthorized users, and certainly not any user on the Internet.

Candidate lower-level policies and procedures for update to accommodate the Internet and WWW include external connectivity, network security, transmission of data, use of electronic commerce, sourcing and procurement, E-mail, nonemployee use of corporate information and electronic systems, access to information, appropriate use of electronic systems, use of corporate assets, etc.

New policies and procedures (most likely enhancements to existing policies) highlight the new environment and present an opportunity to dust off and update old policies. Involve a broad group of customers and functional support areas in the update to these policies. The benefits are many. It exposes everyone to the issues surrounding the new technologies, the new security issues and challenges, and gains buy-in through the development and approval process from those who will have to comply when the policies are approved. It is also an excellent way to raise the awareness level and get attention to security up front.

The most successful corporate security policies and procedures address security at three levels, at the management level through high-level policies, at the functional level through security procedures and technical guidelines, and at the end-user level through user awareness and training guidelines. Consider the opportunity to create or update all three when implementing Internet, intranet, and WWW technologies.

Since these new technologies increase the level of risk and vulnerability to your corporate computing and network environment, security policies should probably be beefed up in the areas of audit and monitoring. This is particularly important because security and technical control mechanisms are not mature for the Internet and WWW and therefore more manual processes need to be put in place and mandated to ensure the protection of information.

The distributed nature of Internet, intranet, and WWW and their inherent security risks can be addressed at a more detailed level through an integrated set of policies, procedures, and technical guidelines. Because these policies and processes will be implemented by various functional support areas, there is a great need to obtain by-in from these groups and ensure coordination and integration through all phases of the systems' life cycle. Individual and collective roles and responsibilities should be clearly delineated to include monitoring and enforcement.

Other areas to consider in the policy update include legal liabilities, risk to competition-sensitive information, employees' use of company time while "surfing" the Internet, use of company logos and trade names by

	Auth.	Trans. Controls	Encryption	Audit	Ownership
External Public Data				(X)	X
Internal Public Data				(X)	X
Internal Cntl. Data	X	X	(X)	X	X
External Cntl. Data	X	X	X	X	X
Update Applications	X	X		X	X

Exhibit 1. Sample Data Protection Classification Hierarchy

employees using the Internet, defamation of character involving company employees, loss of trade secrets, loss of the competitive edge, ethical use of the Internet, etc.

DATA CLASSIFICATION SCHEME

A data classification scheme is important to both reflect existing categories of data and introduce any new categories of data needed to support the business use of the Internet, electronic commerce, and information sharing through new intranet and WWW technologies. The whole area of nonemployee access to information changes the approach to categorizing and protecting company information.

The sample chart below (Exhibit 1) is an example of how general to specific categories of company information can be listed, with their corresponding security and protection requirements to be used as a checklist by application, process, and data owners to ensure the appropriated level of protection, and also as a communication tool to functional area support personnel tasked with resource and information protection. A supplemental chart could include application and system names familiar to corporate employees, or types of general applications and information such as payroll, HR, marketing, manufacturing, etc.

Note that encryption may not be required for the same level of data classification in the mainframe and proprietary networking environment, but in "open" systems and distributed and global networks transmitted data are much more easily compromised. Security should be applied based on a thorough risk assessment considering the value of the information, the risk introduced by the computing and network environment, the technical control mechanisms feasible or available for implementation, and the ease of administration and management support. Be careful to apply the right "balance" of security. Too much is just as costly and ineffective as too little in most cases.

APPROPRIATE USE POLICY

It is important to communicate management's expectation for employee's use of these new technologies. An effective way to do that is to supplement the corporate policies and procedures with a more user-friendly bulletined

Examples of *Unacceptable* Use include but not limited to the following:

1. Using company equipment, functions or services for nonbusiness-related activities while on company time; which in effect is mischarging
2. Using the equipment or services for financial or commercial gain
3. Using the equipment or services for any illegal activity
4. Dial-in usage from home for Internet services for personal gain
5. Accessing nonbusiness-related news groups or BBS
6. Willful intent to degrade or disrupt equipment, software or system performance
7. Vandalizing the data or information of another user
8. Gaining unauthorized access to resources or information
9. Invading the privacy of individuals
10. Masquerading as or using an account owned by another user
11. Posting anonymous messages or mail for malicious intent
12. Posting another employee's personal communication or mail without the original author's consent; this excludes normal business E-mail forwarding
13. Downloading, storing, printing, or displaying files or messages that are profane, obscene, or that use language or graphics which offends or tends to degrade others
14. Transmitting company data over the network to noncompany employees without following proper release precedures
15. Loading software obtained from outside the standard company's procurement channels onto a company system without proper testing and approval
16. Initiating or forwarding electronic chain mail.

Examples of *Acceptable Use* include but not limited to the following:

1. Accessing the Internet, computer resources, fax machines, and phones for information directly related to your work assignment
2. Off-hour usage of computer systems for degree-related school work where allowed by local site practices
3. Job related On Job Training (OJT)

Exhibit 2. Appropriate Use Policy

---

list of requirements. The list should be specific, highlight employee expectations and outline what employees can and cannot do on the Internet, intranet, and WWW. The goal is to communicate with each and every employee, leaving little room for doubt or confusion. An Appropriate Use Policy (Exhibit 2) could achieve these goals and reinforce the higher level. Areas to address include the proper use of employee time, corporate computing and networking resources, and acceptable material to be viewed or downloaded to company resources.

Most companies are concerned with the Telecommunications Act and their liabilities in terms of allowing employees to use the Internet on company time and with company resources. Most find that the trade-off is highly skewed to the benefit of the corporation in support of the utility of the Internet. Guidelines must be carefully spelled out and coordinated with the legal department to ensure that company liabilities are addressed

through clear specification of roles and responsibilities. Most companies do not monitor their employee's use of the Internet or the intranet, but find that audit trail information is critical to prosecution and defense for computer crime.

Overall computer security policies and procedures are the baseline for any security architecture and the first thing to do when implementing any new technology. However, you are never really finished as the development and support of security policies is an iterative process and should be revisited on an ongoing basis to ensure that they are up-to-date, accommodate new technologies, address current risk levels, and reflect the company's use of information and network and computing resources.

There are four basic threats to consider when you begin to use Internet, intranet, and Web technologies:

- Unauthorized alteration of data
- Unauthorized access to the underlying operating system
- Eavesdropping on messages passed between a server and a browser
- Impersonation

Your security strategies should address all four. These threats are common to any technology in terms of protecting information. In the remainder of this chapter, we will build upon the general "good security practices and traditional security management" discussed in the first section and apply these lessons to the technical implementation of security and control mechanisms in the Internet, intranet, and Web environments.

The profile of a computer hacker is changing with the exploitation of Internet and Web technologies. Computerized bulletin board services and network chat groups link computer hackers (formerly characterized as loners and misfits) together. Hacker techniques, programs and utilities, and easy-to-follow instructions are readily available on the net. This enables hackers to more quickly assemble the tools to steal information and break into computers and networks, and it also provides the "would-be" hacker a readily available arsenal of tools.

INTERNAL/EXTERNAL APPLICATIONS

Most companies segment their networks and use firewalls to separate the internal and external networks. Most have also chosen to push their marketing, publications, and services to the public side of the firewall using file servers and Web servers. There are benefits and challenges to each of these approaches. It is difficult to keep data synchronized when duplicating applications outside the network. It is also difficult to ensure the security of those applications and the integrity of the information. Outside the firewall is simply *outside*, and therefore also outside the protections of the internal security environment. It is possible to protect that

information and the underlying system through the use of new security technologies for authentication and authorization. These techniques are not without trade-offs in terms of cost and ongoing administration, management, and support.

Security goals for external applications that bridge the gap between internal and external, and for internal applications using the Internet, intranet, and WWW technologies should all address these traditional security controls:

- Authentication
- Authorization
- Access control
- Audit
- Security administration

Some of what you already used can be ported to the new environment, and some of the techniques and supporting infrastructure already in place supporting mainframe-based applications can be applied to securing the new technologies.

Using the Internet and other public networks is an attractive option, not only for conducting business-related transactions and electronic commerce, but also for providing remote access for employees, sharing information with business partners and customers, and supplying products and services. However, public networks create added security challenges for IS management and security practitioners, who must devise security systems and solutions to protect company computing, networking, and information resources. Security is a CRITICAL component.

Two watchdog groups are trying to protect online businesses and consumers from hackers and fraud. The council of Better Business Bureaus has launched BBBOnline, a service that provides a way to evaluate the legitimacy of online businesses. In addition, the national computer security association, NCSA, launched a certification program for secure WWW sites. Among the qualities that NCSA looks for in its certification process are extensive logging, the use of encryption including those addressed in this chapter, and authentication services.

There are a variety of protection measures that can be implemented to reduce the threats in the Web/server environment, making it more acceptable for business use. Direct server protection measures include secure Web server products which use differing designs to enhance the security over user access and data transmittal. In addition to enhanced secure Web server products, the Web server network architecture can also be addressed to protect the server and the corporate enterprise which could be placed in a vulnerable position due to server enabled connectivity. Both

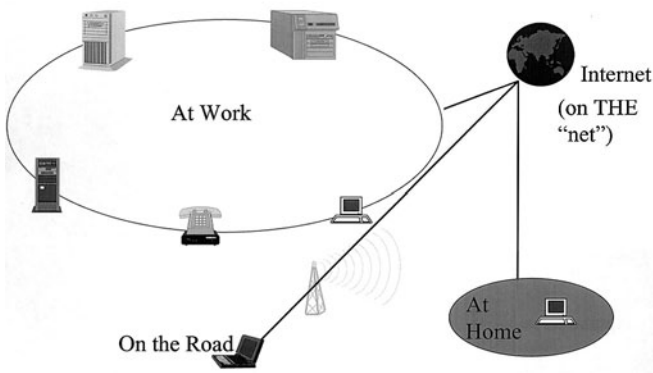


Exhibit 3. Where are your Users?

secure server and secure Web server designs will be addressed, including the application and benefits to using each.

WHERE ARE YOUR USERS?

Discuss how the access point where your users reside contributes to the risk and the security solutions set. Discuss the challenge when users are all over the place and you have to rely on remote security services that are only as good as the users' correct usage. Issues of evolving technologies can also be addressed. Concerns for multiple layering of controls and dissatisfied users with layers of security controls, passwords, hoops, etc. can also be addressed.

WEB BROWSER SECURITY STRATEGIES

Ideally, Web browser security strategies should use a network-based security architecture that integrates your company's external Internet and the internal intranet security policies. Ensure that users on any platform, with any browser, can access any system from any location if they are authorized and have a "need-to-know." Be careful not to adopt the latest evolving security product from a new vendor or an old vendor capitalizing on a hot marketplace.

Recognizing that the security environment is changing rapidly, and knowing that we don't want to change our security strategy, architecture, and control mechanisms every time a new product or solution emerges, we need to take time and use precautions when devising browser security solutions. It is sometimes a better strategy to stick with the vendors that you have already invested in and negotiate with them to enhance their existing products, or even contract with them to make product changes

specific or tailored to accommodate your individual company requirements. Be careful in these negotiations as it is extremely likely that other companies have the very same requirements. User groups can also form a common position and interface to vendors for added clout and pressure.

You can basically secure your Web server as much as or as little as you wish with the current available security products and technologies. The trade offs are obvious: cost, management, administrative requirements, and time. Solutions can be hardware, software and personnel intensive.

Enhancing the security of the Web server itself has been a paramount concern since the first Web server initially emerged, but progress has been slow in deployment and implementation. As the market has mushroomed for server use, and the diversity of data types that are being placed on the server has grown, the demand has increased for enhanced Web server security. Various approaches have emerged, with no single *de facto* standard yet emerging (though there are some early leaders — among them Secure Sockets Layer [SSL] and Secure Hypertext Transfer Protocol [S-HTTP]). These are two significantly different approaches, but both widely seen in the marketplace.

Secure Socket Layer (SSL) Trust Model

One of the early entrants into the secure Web server and client arena is Netscape's Commerce Server, which utilizes the Secure Sockets Layer (SSL) trust model. This model is built around the RSA Public Key/Private Key architecture. Under this model, the SSL-enabled server is authenticated to SSL-aware clients, proving its identity at each SSL connection. This proof of identity is conducted through the use of a public/private key pair issued to the server validated with x.509 digital certificates. Under the SSL architecture, Web server validation can be the only validation performed, which may be all that is needed in some circumstances. This would be applicable for those applications where it is important to the user to be assured of the identity of the target server, such as when placing company orders, or other information submittal where the client is expecting some important action to take place. Exhibit 4 diagrams this process.

Optionally, SSL sessions can be established that also authenticate the client and encrypt the data transmission between the client and the server for multiple I/P services (HTTP, Telnet, FTP). The multiservice encryption capability is available because SSL operates below the application layer and above the TCP/IP connection layer in the protocol stack, and thus other TCP/IP services can operate on top of a SSL-secured session.

Optionally, authentication of a SSL client is available when the client is registered with the SSL server, and occurs after the SSL-aware client connects and authenticates the SSL server. The SSL client then submits its digital

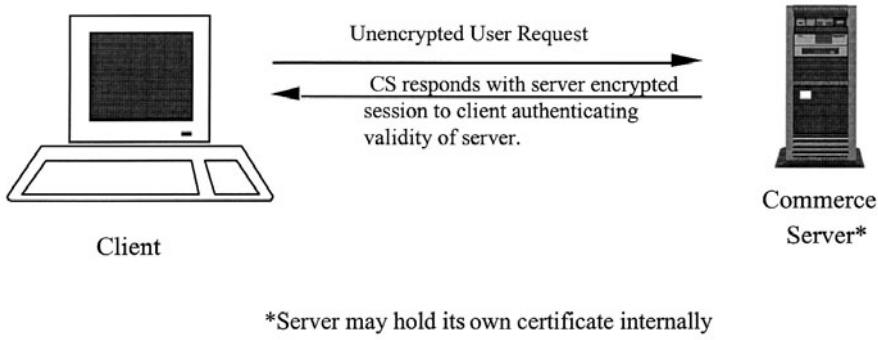


Exhibit 4. Server Authentication

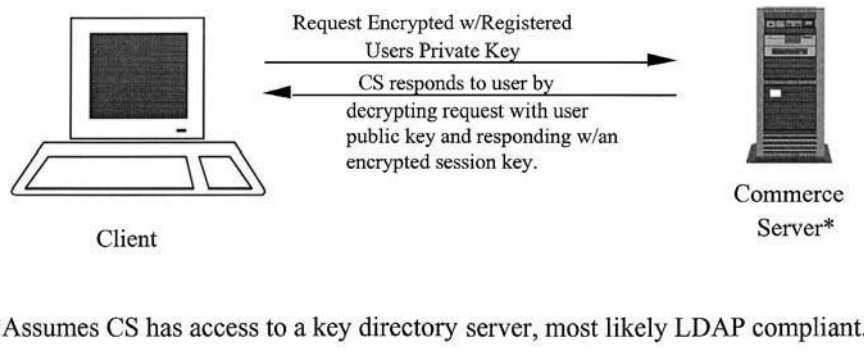


Exhibit 5. Client and Server Authentication

certificate to the SSL server, where the SSL server validates the client's certificate and proceeds to exchange a session key to provide encrypted transmissions between the client and the server. Exhibit 5 provides a graphical representation of this process for mutual client and server authentication under the SSL architecture. This type of mutual client/server authentication process should be considered when the data being submitted by the client are sensitive enough to warrant encryption prior to being submitted over a network transmission path.

Though there are some "costs" with implementing this architecture, these cost variables must be considered when proposing a SSL server implementation to enhance your Web server security. First of all, the design needs to consider whether to only provide server authentication, or both server and client authentication. The issue when expanding the

authentication to include client authentication includes the administrative overhead of managing the user keys, including a key revocation function. This consideration, of course, has to assess the size of the user base, potential for growth of your user base, and stability of your proposed user community. All of these factors will impact the administrative burden of key management, especially if there is the potential for a highly unstable or transient user community.

The positive considerations for implementing a SSL-secured server is the added ability to secure other I/P services for remote or external SSL clients. SSL-registered clients now have the added ability to communicate securely by utilizing Telnet and FTP (or other I/P services) after passing SSL client authentication and receiving their session encryption key. In general the SSL approach has very broad benefits, but these benefits come with the potential added burden of higher administration costs, though if the value of potential data loss is great, then it is easily offset by the administration cost identified above.

Secure Hypertext Transfer Protocol (S-HTTP)

Secure Hypertext Transfer Protocol, (S-HTTP) is emerging as another security tool and incorporates a flexible trust model for providing secure Web server and client HTTP communications. It is specifically designed for direct integration into HTTP transactions, with its focus on flexibility for establishing secure communications in a HTTP environment while providing transaction confidentiality, authenticity/integrity, and nonrepudiation. S-HTTP incorporates a great deal of flexibility in its trust model by leaving defined variable fields in the header definition which identifies the trust model or security algorithm to be used to enable a secure transaction. S-HTTP can support symmetric or asymmetric keys, and even a Kerberos-based trust model. The intention of the authors was to build a flexible protocol that supports multiple trusted modes, key management mechanisms, and cryptographic algorithms through clearly defined negotiation between parties for specific transactions.

At a high level the transactions can begin in a untrusted mode (standard HTTP communication), and "setup" of a trust model can be initiated so that the client and the server can negotiate a trust model, such as a symmetric key-based model on a previously agreed-upon symmetric key, to begin encrypted authentication and communication. The advantage of a S-HTTP-enabled server is the high degree of flexibility in securely communicating with Web clients. A single server, if appropriately configured and network enabled, can support multiple trust models under the S-HTTP architecture and serve multiple client types. In addition to being able to serve a flexible user base, it can also be used to address multiple data classifications on a single server where some data types require higher-level encryption or

protection then other data types on the same server and therefore varying trust models could be utilized.

The S-HTTP model provides flexibility in its secure transaction architecture, but focuses on HTTP transaction vs. SSL which mandates the trust model of a public/private key security model, which can be used to address multiple I/P services. But the S-HTTP mode is limited to only HTTP communications.

INTERNET, INTRANET, AND WORLD WIDE WEB SECURITY ARCHITECTURES

Implementing a secure server architecture, where appropriate, should also take into consideration the existing enterprise network security architecture and incorporate the secure server as part of this overall architecture. In order to discuss this level of integration, we will make an assumption that the secure Web server is to provide secure data dissemination for external (outside the enterprise) distribution and/or access. A discussion of such a network security architecture would not be complete without addressing the placement of the Web server in relation to the enterprise firewall (the firewall being the dividing line between the protected internal enterprise environment and the external "public" environment).

Setting the stage for this discussion calls for some identification of the requirements, so the following list outlines some sample requirements for this architectural discussion on integrating a secure HTTP server with an enterprise firewall.

- Remote client is on public network accessing sensitive company data
- Remote client is required to authenticate prior to receiving data
- Remote client only accesses data via HTTP
- Data is only updated periodically
- Host site maintains firewall
- Sensitive company data must be encrypted on public networks
- Company support personnel can load HTTP server from inside the enterprise

Based on these high-level requirements, an architecture could be set up that would place a S-HTTP server external to the firewall, with one-way communications from inside the enterprise "to" the external server to perform routine administration, and periodic data updates. Remote users would access the S-HTTP server utilizing specified S-HTTP secure transaction modes, and be required to identify themselves to the server prior to being granted access to secure data residing on the server. Exhibit 6 depicts this architecture at a high level. This architecture would support a secure HTTP distribution of sensitive company data, but doesn't provide absolute protection due to the placement of the S-HTTP server entirely external to

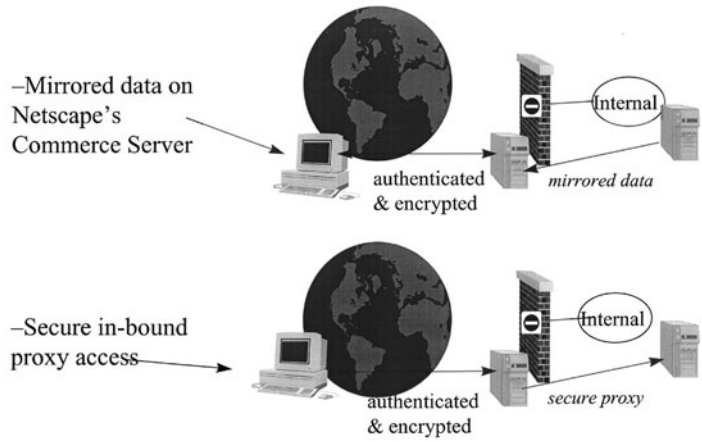


Exhibit 6. Externally Placed Server

the protected enterprise. There are some schools of thought that since this server is unprotected by the company-controlled firewall, the S-HTTP server itself is vulnerable, thus risking the very control mechanism itself and the data residing on it. The opposing view on this is that the risk to the overall enterprise is minimized, as only this server is placed at risk and its own protection is the S-HTTP process itself. This process has been a leading method to secure the data, without placing the rest of the enterprise at risk, by placing the S-HTTP server logically and physically outside the enterprise security firewall.

A slightly different architecture has been advertised that would position the S-HTTP server inside the protected domain, as Exhibit 7 indicates. The philosophy behind this architecture is that the controls of the firewall (and inherent audits) are strong enough to control the authorized access to the S-HTTP server, and also thwart any attacks against the server itself. Additionally, the firewall can control external users so that they only have S-HTTP access via a logically dedicated path, and only to the designated S-HTTP server itself, without placing the rest of the internal enterprise at risk. This architecture relies on the absolute ability of the firewall and S-HTTP of always performing their designated security function as defined; otherwise, the enterprise has been opened for attack through the allowed path from external users to the internal S-HTTP server. Because these conditions are always required to be true and intact, the model with the server external to the firewall has been more readily accepted and implemented.

Both of these architectures can offer a degree of data protection in a S-HTTP architecture when integrated with the existing enterprise firewall

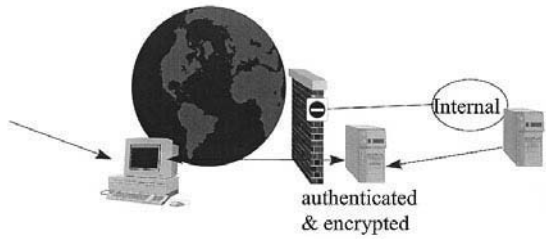


Exhibit 7. Internally Placed Server

architecture. As an aid in determining which architectural approach is right for a given enterprise, a risk assessment can provide great input to the decision. This risk assessment may include decision points such as:

- Available resources to maintain a high degree of firewall audit and S-HTTP server audit
- Experience in firewall and server administration
- Strength of their existing firewall architecture

SECURE WWW CLIENT CONFIGURATION

There is much more reliance on the knowledge and cooperation of the end user and the use of a combination of desktop and workstation software, security control parameters within client software, and security products all working together to mimic the security of the mainframe and distributed application's environments. Consider the areas below during the risk assessment process and the design of WWW security solution sets.

- Ensure that all internal and external company-used workstations have resident and active antivirus software products installed. Preferably use a minimum number of vendor products to reduce security support and vulnerabilities as there are varying vendor schedules for providing virus signature updates.
- Ensure that all workstation and browser client software is preconfigured to return all WWW and other external file transfers to temporary files on the desktop. Under no circumstances should client server applications or process-to-process automated routines download files to system files, preference files, bat files, start-up files, etc.
- Ensure that JAVA script is turned off in the browser client software desktop configuration.
- Configure browser client software to automatically flush the cache, either upon closing the browser or disconnecting from each Web site.
- When possible or available, implement one of the new security products that scans WWW downloads for viruses.

- Provide user awareness and education to all desktop WWW and Internet users to alert them to the inherent dangers involved in using the Internet and WWW. Include information on detecting problems, their roles and responsibilities, your expectations, security products available, how to set and configure their workstations and program products, etc.
- Suggest or mandate the use of screen savers, security software programs, etc., in conjunction with your security policies and distributed security architectures.

This is a list of current areas of concern from a security perspective. There are options that when combined can tailor the browser to the specifications of individual workgroups or individuals. These options will evolve with the browser technology. The list should continue to be modified as security problems are corrected or as new problems occur.

AUDIT TOOLS AND CAPABILITIES

As we move further and further from the "good old days" when we were readily able to secure the "glass house", we rely more on good and sound auditing practices. As acknowledged throughout this chapter, security control mechanisms are mediocre at best in today's distributed networking and computing environments. Today's auditing strategies must be robust, available across multiple heterogeneous platforms, computing and network based, real-time and automated, and integrated across the enterprise.

Today, information assets are distributed all over the enterprise, and therefore auditing strategies must acknowledge and accept this challenge and accommodate more robust and dicey requirements. As is the case when implementing distributed security control mechanisms, in the audit environment there are also many players and functional support areas involved in collecting, integrating, synthesizing, reporting, and reconciling audit trails and audit information. The list includes applications and applications developers and programs, data base management systems and data base administrators, operating systems and systems administrators, local area network (LAN) administrators and network operating systems (NOS), security administrators and security software products, problem reporting and tracking systems and helpline administrators, and others unique to the company's environment.

As well as real-time, the audit system should provide for tracking and alarming, both to the systems and network management systems, and via pagers to support personnel. Policies and procedures should be developed for handling alarms and problems, i.e., isolate and monitor, disconnect, etc.

There are many audit facilities available today, including special audit software products for the Internet, distributed client server environments,

WWW clients and servers, Internet firewalls, E-mail, News Groups, etc. The application of one or more of these must be consistent with your risk assessment, security requirements, technology availability, etc. The most important point to make here is the fundamental need to centralize distributed systems auditing (not an oxymoron). Centrally collect, sort, delete, process, report, take action and store critical audit information. Automate any and all steps and processes. It is a well-established fact that human beings cannot review large numbers of audit records and logs and reports without error. Today's audit function is an adjunct to the security function, and as such is more important and critical than ever before. It should be part of the overall security strategy and implementation plan.

The overall audit solutions set should incorporate the use of browser access logs, enterprise security server audit logs, network and firewall system authentication server audit logs, application and middle-ware audit logs, URL filters and access information, mainframe system audit information, distributed systems operating system audit logs, data base management system audit logs, and other utilities that provide audit trail information such as accounting programs, network management products, etc.

The establishment of auditing capabilities over WWW environments follows closely with the integration of all external WWW servers with the firewall, as previously mentioned. This is important when looking at the various options available to address a comprehensive audit approach.

WWW servers can offer a degree of auditability based on the operating system of the server on which they reside. The more time-tested environments such as UNIX are perceived to be difficult to secure, whereas the emerging NT platform with its enhanced security features supposedly make it a more secure and trusted platform with a wide degree of audit tools and capabilities (though the vote is still out on NT, as some feel it hasn't had the time and exposure to discover all the potential security holes, perceived or real). The point, though, is that in order to provide some auditing the first place to potentially implement the first audit is on the platform where the WWW server resides. Issues here are the use of privileged accounts and file logs and access logs for log-ins to the operating system, which could indicate a backdoor attack on the WWW server itself. If server-based log are utilized, they of course must be file protected and should be off-loaded to a nonserver-based machine to protect against after-the-fact corruption.

Though the server logs aren't the only defensive logs that should be relied upon in a public WWW server environment, the other components in the access architecture should be considered for use as audit log tools. As previously mentioned, the WWW server should be placed in respect to its required controls in relation to the network security firewall. If it is a S-HTTP server that is placed behind (Exhibit 4) the firewall then the firewall of

course has the ability to log all access to the S-HTTP server and provide a log separate from the WWW server-based logs, and is potentially more secure should the WWW server somehow become compromised.

The prevalent security architecture places externally accessible WWW servers wholly outside the firewall, thus virtually eliminating the capability of auditing access to the WWW server except from users internal to the enterprise. In this case, the network security audit in the form of the network management tool, which monitors the "health" of enterprise components can be called upon to provide a minimal degree of audit over the status of your external WWW server. This type of audit can be important when protecting data which resides on your external server from being subject to "denial of service" attacks, which are not uncommon for external devices. But by utilizing your network management tool to guard against such attacks, and monitoring log alerts on the status or health of this external server, you can reduce the exposure to this type of attack.

Other outside devices that can be utilized to provide audit include the network router between the external WWW server and the true external environment, though these devices are not normally readily set up for comprehensive audit logs, but in some critical cases they could be reconfigured with added hardware and minimal customized programming. One such example would be the "I/P Accounting" function on a popular router product line, which allows off-loading of addresses and protocols through its external interface. This could be beneficial to analyze traffic, and if an attack alert was generated from one of the other logs mentioned, then these router logs could assist in possibly identifying the origin of the attack.

Another possible source of audit logging could come from "back end" systems that the WWW server is programmed to "mine" data from. Many WWW environments are being established to serve as "front ends" for much larger data repositories, such as Oracle data bases, where the WWW server receives user requests for data over HTTP, and the WWW server launches SQL_Net queries to a back end Oracle data base. In this type of architecture the more developed logging inherent to the Oracle environment can be called upon to provide audits over the WWW queries. The detailed Oracle logs can specify the quantity, data type, and other activity over all the queries that the WWW server has made, thus providing a comprehensive activity log that can be consolidated and reviewed should any type of WWW server compromise be suspected. A site could potentially discover the degree of data exposure though these logs.

These are some of the major areas where auditing can be put in place to monitor the WWW environment while enhancing its overall security. It is important to note that the potential placement of audits encompasses the entire distributed computing infrastructure environment, not just the new WWW server itself. In fact, there are some schools of thought that consider

the more reliable audits to be those that are somewhat distanced from the target server, thus reducing the potential threat of compromise to the audit logs themselves. In general, the important point is to look at the big picture when designing the security controls and a supporting audit solution.

WWW/Internet Audit Considerations

After your distributed Internet, intranet, and WWW security policies are firmly established, distributed security architectures are updated to accommodate this new environment. When planning for audit, and security control mechanisms are designed and implemented, you should plan how you will implement the audit environment — not only which audit facilities to use to collect and centralize the audit function, but how much and what type of information to capture, how to filter and review the audit data and logs, and what actions to take on the violations or anomalies identified. Additional consideration should be given to secure storage and access to the audit data. Other considerations include:

- Timely resolution of violations
- Disk space storage availability
- Increased staffing and administration
- In-house developed programming
- Ability to alarm and monitor in real time

WWW SECURITY FLAWS

As with all new and emerging technology, many initial releases come with some deficiency. But this has been of critical importance when that deficiency can impact the access or corruption of a whole corporation or enterprise's display to the world. This can be the case with Web implementations utilizing the most current releases which have been found to contain some impacting code deficiencies, though up to this point most of these deficiencies have been identified before any major damage has been done. This underlines the need to maintain a strong link or connection with industry organizations that announce code shortcomings that impact a sites Web implementation. A couple of the leading organizations are CERT, the Computer Emergency Response Team, and CIAC, Computer Incident Advisory Capability.

Just a few of these types of code or design issues that could impact a sites Web security include initial issues with the Sun JAVA language and Netscape's JavaScript (which is an extension library of their HyperText Markup Language, HTML).

The Sun Java language was actually designed with some aspects of security in mind, though upon its initial release there were several functions that were found to be a security risk. One of the most impacting bugs in an

early release was the ability to execute arbitrary machine instructions by loading a malicious Java applet. By utilizing Netscape's caching mechanism a malicious machine instruction can be downloaded into a user's machine and Java can be tricked into executing it. This doesn't present a risk to the enterprise server, but the user community within one's enterprise is of course at risk.

Other Sun Java language bugs include the ability to make network connections with arbitrary hosts (though this has since been patched with the following release) and Java's ability to launch denial of service attacks though the use of corrupt applets.

These types of security holes are more prevalent than the security profession would like to believe, as the JavaScript environment also was found to contain capabilities that allowed malicious functions to take place. The following three are among the most current and prevalent risks:

- JavaScripts ability to trick the user into uploading a file on his local hard disk to an arbitrary machine on the Internet
- The ability to hand out the user's directory listing from the internal hard disk
- The ability to monitor all pages the user visits during a session

The following are among the possible protection mechanisms:

- Maintain monitoring through CERT or CIAC, or other industry organizations that highlight such security risks.
- Utilize a strong software distribution and control capability, so that early releases aren't immediately distributed, and that new patched code known to fix a previous bug is released when deemed safe.
- In sensitive environments it may become necessary to disable the browser's capability to even utilize or execute JAVA or JavaScript — a selectable function now available in many browsers.

In the last point, it can be disturbing to some in the user community to disallow the use of such powerful tools, because they can be utilized against trusted Web pages, or those that require authentication through the use of SSL or S-HTTP. This approach can be coupled with the connection to S-HTTP pages where the target page has to prove its identity to the client user. In this case, enabling Java or JavaScripts to execute on the browser (a user-selectable option) could be done with a degree of confidence.

Other perceived security risks exist in a browser feature referred to as HTTP "Cookies." This is a feature that allows servers to store information on the client machine in order to reduce the store and retrieve requirements of the server. The cookies file can be written to by the server, and that server, in theory, is the only one that can read back their cookies entry. Uses of the cookie file include storing user's preferences or browser history

on a particular server or page, which can assist in guiding the user on their next visit to that same page. The entry in the cookies file identifies the information to be stored and the uniform resource locator (URL) or server page that can read back that information, though this address can be masked to some degree so multiple pages can read back the information.

The perceived security concern is that pages impersonating cookies-readable pages could read back a user's cookies information without the user knowing it, or discover what information is stored in their cookie file. The threat depends on the nature of the data stored in the cookie file, which is dependent on what the server chooses to write into a user's cookie file. This issue is currently under review, with the intention of adding additional security controls to the cookie file and its function. At this point it is important that users are aware of the existence of this file, which is viewable in the Macintosh environment as a Netscape file and in the Win environment as a cookies.txt file. There are already some inherent protections in the cookie file: one is the fact that the cookie file currently has a maximum of 20 entries, which potentially limits the exposure. Also, these entries can be set up with expiration dates to they don't have an unlimited lifetime.

WWW SECURITY MANAGEMENT

Consider the overall management of the Internet, intranet, and WWW environment. As previously mentioned, there are many players in the support role and for many of them this is not their primary job or priority. Regardless of where the following items fall in the support infrastructure, also consider these points when implementing ongoing operational support:

- Implement WWW browser and server standards.
- Control release and version distribution.
- Implement secure server administration including the use of products and utilities to erase sensitive data cache (NSClean).
- Ensure prompt problem resolution, management, and notification.
- Follow industry and vendor discourse on WWW security flaws and bugs including CERT distribution.
- Stay current on new Internet and WWW security problems, Netscape encryption, JAVA, Cookies, etc.

WWW SUPPORT INFRASTRUCTURE

- WWW servers accessible from external networks should reside outside the firewall and be managed centrally.
- By special approval, decentralized programs can manage external servers, but must do so in accordance with corporate policy and be subjected to rigorous audits.

- Externally published company information must be cleared through legal and public relations departments (i.e., follow company procedures).
- External outbound http access should utilize proxy services for additional controls and audit.
- WWW application updates must be authenticated utilizing standard company security systems (as required).
- Filtering and monitoring software must be incorporated into the firewall.
- The use of discovery crawler programs must be monitored and controlled.
- Virus software must be active on all desktop systems utilizing WWW.
- Externally published information should be routinely updated or verified through integrity checks.

In conclusion, as information security practitioners embracing the technical challenges of the 21st century, we are continually challenged to integrate new technology smoothly into our existing and underlying security architectures. Having a firm foundation or set of security principles, frameworks, philosophies and supporting policies, procedures, technical architectures, etc. will assist in the transition and our success.

Approach new technologies by developing processes to manage the integration and update the security framework and supporting infrastructure, as opposed to changing it. The Internet, intranet, and the World Wide Web is exploding around us — what is new today is old technology tomorrow. We should continue to acknowledge this fact while working aggressively with other MIS and customer functional areas to slow down the train to progress, be realistic, disciplined, and plan for new technology deployment.

39

An Introduction to IPSec

Bill Stackpole, CISSP

The IP Security Protocol Working Group (IPSec) was formed by the Internet Engineering Task Force (IETF) in 1992 to develop a standardized method for implementing privacy and authentication services on IP version 4 and the emerging version 6 protocols. There were several specific goals in mind. For the architecture to be widely adopted it would have to be flexible. It must be able to accommodate changes in cryptographic technology as well as the international restrictions on cryptographic use. Second, the architecture must support all the client IP protocols (i.e., Transmission Control Protocol or TCP, User Datagram Protocol or UDP) in standard or cast (i.e., multicast) modes. Third, it must be able to secure communications between two hosts or multiple hosts, two subnets or multiple subnets, or a combination of hosts and subnets. Finally, there had to be a method for automatically distributing the cryptographic keys. This chapter will cover the key features of the IPSec security architecture, its major components, and the minimum mandatory requirements for compliance.

Features

The goals of IPSec were transformed into the following key architectural features.

Separate Privacy and Authentication Functions with Transform Independence

IPSec privacy and authentication services are independent of each other. This simplifies their implementation and reduces their performance impact upon the host system. It also gives end users the ability to select the appropriate level of security for their transaction. The security functions are independent of their cryptographic transforms. This allows new encryption technologies to be incorporated into IPSec without changing the base architecture and avoids conflicts with location-specific use and exportation restrictions. It also makes it possible for end users to implement transforms that best meet their specific security requirements. Users can select authentication services using hashed cryptography which have low implementation costs, minimal performance impacts, and few international use restrictions. These implementations can be widely distributed and they provide a substantial improvement in security for most of today's Internet transactions. Or, users can select privacy functions based on private key cryptography. These are more difficult to implement, have higher performance impacts, and are often subject to international use restrictions, so although they provide a much higher level of security, their distribution and use is often limited. Or they can combine these functions to provide the highest possible level of security.

Network Layer (IP) Implementation with Unidirectional Setup

Introducing security functionality at the network layer means all the client IP protocols can operate in a secure manner without individual customization. Routing protocols like Exterior Gateway Protocol (EGP) and Border Gateway Protocol (BGP) as well as connection and connectionless transport protocols like TCP and UDP can be secured. Applications using these client protocols require no modifications to take advantage of IPSec

security services. The addition of IPSec services makes it possible to secure applications with inherent security vulnerabilities (e.g., clear-text password) with a single system modification. And this modification will secure any such application regardless of the IP services or transports it utilizes.

This capability even extends to streaming services using multicast and unicast packets where the destination address is indeterminate. IPSec makes this possible by using a unidirectional initialization scheme to set up secure connections. The sending station passes a setup index to the receiving station. The receiving station uses this index to reference the table of security parameters governing the connection. The receiving station does not need to interact with the sending station to establish a secure unidirectional connection. For bidirectional connections the process is reversed. The receiving station becomes the sender, passing its setup index back to the originator. Sending and receiving stations can be either hosts or security gateways.

Host and Gateway Topologies

IPSec supports two basic connection topologies: host-to-host and gateway-to-gateway. In the host (sometimes called end-to-end) topology, the sending and receiving systems are two or more hosts that establish secure connections to transmit data among themselves. In the gateway (also called subnet-to-subnet) topology, the sending and receiving systems are security gateways that establish connection to external (untrusted) systems on behalf of trusted hosts connected to their own internal (trusted) subnetwork(s). A trusted subnet-work is defined as a communications channel (e.g., Ethernet) containing one or more hosts that trust each other not to be engaged in passive or active attacks. A gateway-to-gateway connection is often referred to as a tunnel or a virtual private network (VPN). A third scenario, host-to-gateway, is also possible. In this instance the security gateway is used to establish connection between external hosts and trusted hosts on an internal subnet(s). This scenario is particularly useful for traveling workers or telecommuters who require access to applications and data on internal systems via untrusted networks like the Internet.

Key Management

The ability to effectively manage and distribute encryption keys is crucial to the success of any cryptographic system. The IP Security Architecture includes an application-layer key management scheme that supports public and private key-based systems and manual or automated key distribution. It also supports the distribution of other principle session parameters. Standardizing these functions makes it possible to use and manage IPSec security functions across multiple security domains and vendor platforms.

Two other key features of the IPSec Security Architecture are support for systems with Multi-Level Security (MLS) and the use of IANA (Internet Assigned Numbers Authority) assigned numbers for all standard IPSec type codes.

Implementation and Structures

The IPSec Security Architecture is centered around two IP header constructs: the Authentication Header (AH) and the Encapsulation Security Payload (ESP) header. To fully understand how these mechanisms function it is first necessary to look at the concept of security associations. In order to achieve algorithm independence, a flexible method for specifying session parameters had to be established. Security associations (SAs) became that method.

Security Associations (SA)

A security association is a table or database record consisting of a set of security parameters that govern security operations on one or more network connections. Security associations are part of the unidirectional initialization scheme mentioned above. The SA tables are established on the receiving host and referenced by the sending host using an index parameter known as the Security Parameters Index (SPI). The most common entries in an SA are:

- T*he type and operating mode of the transform*, for example DES in block chaining mode. This is a required parameter. Remember that IPSec was designed to be transform independent so this information must be synchronized between the endpoints if any meaningful exchange of data is going to take place.

- *The key or keys used by the transform algorithm*. For obvious reasons this is also a mandatory parameter. The source of the keys can vary. They can be entered manually when the SAS is defined on the host or gateway. They can be supplied via a key distribution system or — in the case of asymmetric encryption — the public key is sent across the wire during the connection setup.
- *The encryption algorithm's synchronization or initialization vector*. Some encryption algorithms, in particular those that use chaining, may need to supply the receiving system with an initial block of data to synchronize the cryptographic sequence. Usually, the first block of encrypted data serves this purpose, but this parameter allows for other implementations. This parameter is required for all ESP implementations but may be designated as "absent" if synchronization is not required.
- *The life span of the transform key(s)*. The parameter can be an expression of duration or a specific time when a key change is to occur. There is no predefined life span for cryptographic keys. The frequency with which keys are changed is entirely at the discretion of the security implementers at the endpoints. Therefore, this parameter is only recommended, not required.
- *The life span of the security association*. There is no predefined life span for a security association. The length of time a security association remains in effect is at the discretion of the endpoint implementers. Therefore, this parameter is also recommended, but not required.
- *Source address of the security association*. A security association is normally established in one direction only. A communications session between two endpoints will usually involve two security associations. When more than one sending host is using this security association, the parameter may be set to a wild-card value. Usually this address is the same as the source address in the IP header; therefore, this parameter is recommended, but not required.
- *The sensitivity level of the protected data*. This parameter is required for hosts implementing multilevel security and recommended for all other systems. The parameter provides a method of attaching security labels (e.g., Secret, Confidential, Unclassified) to ensure proper routing and handling by the endpoints.

Security associations are normally set up in one direction only. Before a secure transmission can be established, the SAs must be created on the sending and receiving hosts. These security associations can be configured manually or automatically via a key management protocol. When a datagram destined for a (secure) receiving host is ready to be sent, the sending system looks up the appropriate security association and passes the resulting index value to the receiving host. The receiving host uses the SPI and the destination address to look up the corresponding SA on its system. In the case of multilevel security, the security label also becomes part of the SA selection process. The receiving system then uses those SA parameters to process all subsequent packets from the sending host. To establish a fully authenticated communications session, the sending and receiving hosts would reverse roles and establish a second SA in the reverse direction.

One advantage to this unidirectional SA selection scheme is support for broadcast types of traffic. Security associations can still be established even in this receive-only scenario by having the receiving host select the SPI. Unicast packets can be assigned a single SPI value, and multicast packets can be assigned an SPI for each multicast group. However, the use of IPSec for broadcast traffic does have some serious limitations. The key management and distribution is difficult, and the value of cryptography is diminished because the source of the packet cannot be positively established.

Security Parameters Index (SPI)

The Security Parameters Index is a 32-bit pseudo-random number used to uniquely identify a security association (SA). The source of an SPI can vary. They can be entered manually when the SA is defined on the host or gateway, or they can be supplied via an SA distribution system. Obviously for the security function to work properly, the SPIs must be synchronized between the endpoints. SPI values 1 through 255 have been reserved by the IANA for use with openly specified (i.e., standard) implementations. SPIs require minimal management but some precautions should be observed to ensure that previously assigned SPIs are not reused too quickly after their associated SA has been deleted. An SPI value of zero (0) specifies that no security association exists for this transaction. On host-to-host connections, the SPI is used by the receiving host to look up the security association. On a gateway-to-gateway, unicast, or multicast transaction, the receiving system combines the SPI with the destination address (and in an MLS system, with the security label) to determine the appropriate SA. Now we will look at how IPSec authentication and privacy functions utilize SAs and SPIs.

Authentication Function

IPSec authentication uses a cryptographic hashing function to provide strong integrity and authentication for IP datagrams. The default algorithm is keyed Message Digest version 5 (MD5), which does not provide non-repudiation. Non-repudiation can be provided by using a cryptographic algorithm that supports it (e.g., RSA). The IPSec authentication function does not provide confidentiality or traffic analysis protection.

The function is computed over the entire datagram using the algorithm and keys(s) specified in the security association (SA). The calculation takes place prior to fragmentation, and fields that change during transit (e.g., ttl or hop count) are excluded. The resulting authentication data is placed into the Authentication Header (AH) along with the Security Parameter Index (SPI) assigned to that SA. Placing the authentication data in its own payload structure (the AH) rather than appending it to the original datagram means the user datagram maintains its original format and can be read and processed by systems not participating in the authentication. Obviously there is no confidentiality, but there is also no need to change the Internet infrastructure to support the IPSec authentication function. Systems not participating in the authentication can still process the datagrams normally.

The Authentication Header (AH) is inserted into the datagram immediately following the IP header (IPv4) or the Hop-by-Hop Header (IPv6) and prior to the ESP header when used with the confidentiality function, as seen in Exhibit 39.1.

The header type is IANA assigned number 51 and is identified in the next header or the protocol field of the preceding header structure. There are five parameter fields in an authentication header, four of which are currently in use (see also Exhibit 39.2):

- The next header field — used to identify the IP protocol (IANA assigned number) used in the next header structure.
- The payload length — the number of 32-bit words contained in the authentication data field.
- The reserved field — intended for future expansion. This field is currently set to zero (0).
- The SPI field — the value that uniquely identifies the security association (SA) used for this datagram.
- The authentication data field — the data output by the cryptographic transform padded to the next 32-bit boundary.

IP version 4 systems claiming AH compliance must implement the IP Authentication Header with at least the MD5 algorithm using a 128-bit key. Implementation of AH is mandatory for all IP version 6 hosts and must also implement the MD5 algorithm with a 128-bit key. All AH implementations have an option to support other additional authentication algorithms (e.g., SHA-1). In fact, well-known weaknesses in the current MD5 hash functions (see Hans Dobbertin, Cryptanalysis of MD5 Compress) will undoubtedly lead to its replacement in the next version of the AH specification. The likely replacement is HMAC-MD5. HMAC is an enhanced method for calculating Hashed Message Authentication Codes that greatly increased the cryptographic strength of the underlying algorithm. Because HMAC is an enhancement rather than a replacement, it can be easily added to existing AH implementations with little impact upon the original algorithm's performance. Systems using MLS are required to implement AH on packets containing sensitivity labels to ensure the end-to-end integrity of those labels.

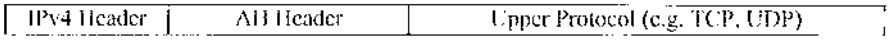


EXHIBIT 39.1 IPv4 placement example.

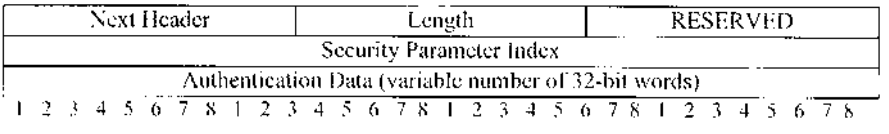


EXHIBIT 39.2 IP Authentication Header structure.

The calculation of hashed authentication data by systems using the Authentication Header does increase processing costs and communications latency; however, this impact is considerably less than that of a secret key cryptographic system. The Authentication Header function has a low implementation cost and is easily exportable because it is based on a hashing algorithm. Nevertheless , it would still represent a significant increase in security for most of the current Internet traffic.

Confidentiality Function

IPSec confidentiality uses keyed cryptography to provide strong integrity and confidentiality for IP datagrams. The default algorithm uses the Cipher Block Chaining mode of the U.S. Data Encryption Standard (DES CBC), which does not provide authentication or non-repudiation. It is possible to provide authentication by using a cryptographic transform that supports it. However, it is recommended that implementation requiring authentication or nonrepudiation use the IP Authentication Header for that purpose. The IPSec confidentiality function does not provide protection from traffic analysis attacks.

There are two modes of operation: tunnel and transport. In tunnel mode the entire contents of the original IP datagram are encapsulated into the Encapsulation Security Payload (ESP) using the algorithm and key(s) specified in the security association (SA). The resulting encrypted ESP along with the Security Parameter Index (SPI) assigned to this SA become the payload portion of a second datagram with a cleartext IP header. This cleartext header is usually a duplicate of the original header for host-to-host transfers, but in implementations involving security gateways the cleartext header usually addresses the gateway, while the encrypted header's addressing point is the endpoint host on an interior subnet. In transport mode only the transport layer (i.e., TCP, UDP) portion of the frame is encapsulated into the ESP so the cleartext portions of the IP header retain their original values. Although the term "transportmode" seems to imply a use limited to TCP and UDP protocols, this is a misnomer. Transport mode ESP supports all IP client protocols. Processing for both modes takes place prior to fragmentation on output and after reassembly on input.

The Encapsulation Security Payload (ESP) header can be inserted anywhere in the datagram after the IP Header and before the transport layer protocol. It must appear after the AH header when used with the authentication function (see Exhibit 39.3).

The header type is IANA-assigned number 50 and is identified in the next header or the protocol field of the preceding header structure. The ESP header contains three fields (Exhibit 39.4):

- The SPI field — the unique identifier for the SA used to process this datagram. This is the only mandatory ESP field.
- The opaque transform data field — additional parameters required to support the cryptographic transform used by this SA (e.g., an initialization vector). The data contained in this field is transform specific and therefore varies in length. The only IPSec requirement is that the field be padded so it ends on a 32-bit boundary.
- The encrypted data field — the data output by the cryptographic transform.

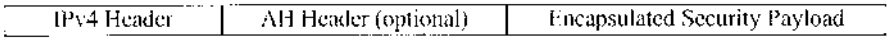


EXHIBIT 39.3 IPv4 Placement Example.

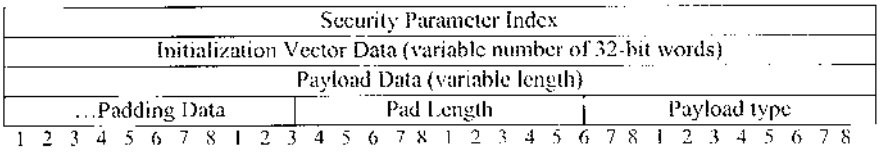


EXHIBIT 39.4 IP ESP Header Structure.

IP version 4 or version 6 systems claiming ESP compliance must implement the Encapsulation Security Protocol supporting the use of the DES CBC transform. All ESP implementations have an option to support other encryption algorithms. For example, if no valid SA exists for an arriving datagram (e.g., the receiver has no key), the receiver must discard the encrypted ESP and record the failure in a system or audit log. The recommended values to be logged are the SPI value, date/time, the sending and destination addresses, and the flow ID. The log entry may include other implementation-specific data. It is recommended that the receiving system not send immediate notification of failures to the send system because of the strong potential for easy-to-exploit denial-of-service attacks.

The calculation of the encrypted data by systems using the ESP does increase processing costs and communications latency. The overall impact depends upon the cryptographic algorithm and the implementation. Secret key algorithms require much less processing time than public key algorithms, and hardware-based implementations tend to be even faster with very little system impact.

The Encapsulation Security Payload function is more difficult to implement and subject to some international export and use restrictions, but its flexible structure, VPN capabilities, and strong confidentiality are ideal for businesses requiring secure communications across untrusted networks.

Key Management

Key management functions include the generation, authentication, and distribution of the cryptographic keys required to establish secure communications. The functions are closely tied to the cryptographic algorithms they are supporting but, in general, generation is the function that creates the keys and manages their life span and disposition; authentication is the process used to validate the hosts or gateways requesting keys services; and distribution is the process that transfers the keys to the requesting systems in a secure manner.

There are two common approaches to IP keying: host-oriented and user-oriented. Host-oriented keys have all users sharing the same key when transferring data between endpoint (i.e., hosts and gateways). User-oriented keying establishes a separate key for each user session that is transferring data between endpoints. The keys are not shared between users or applications. Users have different keys for Telnet and FTP sessions. Multilevel security (MLS) systems require user-oriented keying to maintain confidentiality between the different sensitivity levels. But it is not uncommon on non-MLS systems to have users, groups, or processes that do not trust each other. Therefore, the IETF Security Working Group strongly recommends the use of user-oriented keying for all IPSec key management implementations.

Thus far we have only mentioned traditional cryptographic key management. However, traditional key management functions are not capable of supporting a full IPSec implementation. IPSec's transform independence requires that all the elements of the security association, not just the cryptographic keys, be distributed to the participating endpoints. Without all the security association parameters, the endpoints would be unable to determine how the cryptographic key is applied. This requirement led to the development of the Internet Security Association and Key Management Protocol (ISAKMP). ISAKMP supports the standard key management functions and incorporates mechanisms to negotiate, establish, modify, and delete security associations and their attributes. For the remainder of this section we will use the term "SA management" to refer to the management of the entire SA structure (including cryptographic keys) and key management to refer to just the cryptographic key parameters of an SA. It is important to note that key management can take place separate from SA management. For example, host-oriented keying would use SA management to establish both the session parameters and the cryptographic keys, whereas user-oriented keying would use the SA management function to establish the initial session parameters and the key management function to supply the individual-use session keys.

The simplest form of SA or key management is manual management. The system security administrator manually enters the SA parameters and encryption keys for their system and the system(s) it communicates with. All IPv4 and IPv6 implementations of IPSec are required to support the manual configuration of security associations and keys. Manual configuration works well in small, static environments but is extremely difficult to scale to larger environments, especially those involving multiple administrative domains. In these environments the SA and key management functions must be automated and centralized to be effective. This is the functionality ISAKMP is designed to provide.

Internet Security Association and Key Management Protocol (ISAKMP)

ISAKMP provides a standard, flexible, and scalable methodology for distributing security associations and cryptographic keys. The protocol defines the procedures for authenticating a communicating peer, creating and managing security associations, techniques for generating and managing keys and security associations, and ways to mitigate threats like replay and denial-of-service attacks. ISAKMP was designed to support IPSec AH and ESP services, but it goes far beyond that. ISAKMP has the capability of supporting security services at the transport and applications layers for a variety of security mechanisms. This is possible because ISAKMP separates the security association management function from the key exchange mechanism. ISAKMP has key exchange protocol independence. It provides a common framework for negotiating, exchanging, modifying, and deleting SAs between dissimilar systems. Centralizing the management of the security associations with ISAKMP reduces much of the duplicated functionality within each security protocol and significantly reduces the connection setup time because ISAKMP can negotiate an entire set of services at once.

A detailed discussion of ISAKMP is beyond the scope of this chapter so only the operations and functional requirements of a security association and key management system will be covered. A security association and key management system is a service application that mediates between systems establishing secure connections. It does not actively participate in the transfer of data between these systems. It only assists in the establishment of a secure connection by generating, authenticating, and distributing the required security associations and cryptographic keys.

Two parameters must be agreed upon for the system to work properly. First, a trust relationship must be established between the endpoint systems and the SA manager. The SA manager can be a third-party system — similar to a Kerberos Key Distribution Center (KDC) — or integrated into the endpoint's IPSec implementation. Each approach requires a manually configured SA for each manager and the endpoints it communicates with. The advantage is these few manual SAs can be used to establish a multitude of secure connections. Most vendors have chosen to integrate ISAKMP into the endpoint systems and use a third-party (e.g., Certificate Authority) system to validate the initial trust relationship. The second requirement is for the endpoints to have a trusted third party in common. In other words, both endpoints must have an SA management system or Certificate Authority they both trust.

The operation is pretty straightforward. We will use systems with integrated SAs for this scenario. System A wishes to establish a secure communications session with System B and no valid security association currently exists between them. System A contacts the SA management function on System B. The process then reverses itself (remember that SAs are only established in one direction) as System B establishes a secure return path to System A. ISAKMP does have the capability of negotiating bidirectional SAs in a single transaction, so a separate return path negotiation is usually not required.

ISAKMP has four major functional components. They are:

1. Authentication of communications peers
2. Cryptographic key establishment and management
3. Security association creation and management
4. Threat mitigation

Authenticating the entity at the other end of the communication is the first step in establishing a secure communications session. Without authentication it is impossible to trust an entity's identification, and without a valid ID access control is meaningless. What value is there to secure communication with an unauthorized system?

ISAKMP mandates the use of public key digital signatures (e.g., DSS, RSA) to establish strong authentication for all ISAKMP exchanges. The standard does not specify a particular algorithm. Public key cryptography is a very effective, flexible, and scalable way to distribute shared secrets and session keys. However, to be completely effective, there must be a means of binding public keys to a specific entity. In larger implementations, this function is provided by a trusted third party (TTP) like a Certificate Authority (CA). Smaller implementations may choose to use manually configured keys. ISAKMP does not define the protocols used for communication with trusted third parties.

Key establishment encompasses the generation of the random keys and the transportation of those keys to the participating entities. In an RSA public key system, key transport is accomplished by encrypting the session

key with the recipient's public key. The encrypted session key is then sent to the recipient system, which decrypts it with its private key. In a Diffie–Hellman system, the recipient's public key would be combined with the sender's private key information to generate a shared secret key. This key can be used as the session key or for the transport of a second randomly generated session key. Under ISAKMP these key exchanges must take place using strong authentication. ISAKMP does not specify a particular key exchange protocol, but it appears that Oakley will become the standard.

Security association creation and management is spread across two phases of connection negotiation. The first phase establishes a security association between the two endpoint SA managers. The second phase establishes the security associations for the security protocols selected for that session. Phase one constitutes the trust between the managers and endpoints; the second phase constitutes the trust between the two endpoints themselves. Once phase two has been completed, the SA manager has no further involvement in the connection.

ISAKMP integrates mechanisms to counteract threats like denial of service, hijacking, and man-in-the-middle attacks. The manager service sends an anti-clogging token (cookie) to the requesting system prior to performing any CPU-intensive operation. If the manager does not receive a reply to this cookie, it assumes the request is invalid and drops it. Although this certainly is not comprehensive anti-clogging protection, it is quite effective against most common flooding attacks. The anti-clogging mechanism is also useful for detecting redirection attacks. Because multiple cookies are sent during each session setup, any attempt to redirect the data stream to a different endpoint will be detected.

ISAKMP links the authentication process and the SA/key exchange process into a single data stream. This makes attacks which rely on the interception or modification of the data stream (e.g., hijacking, man-in-the-middle) completely ineffective. Any interruption or modification of the data stream will be detected by the manager and further processing halted. ISAKMP also employs a built-in state machine to detect data deletions, thus ensuring that SAs based on partial exchanges will not be established. As a final anti-threat, ISAKMP specifies logging and notification requirements for all abnormal operations and limits the use of on-the-wire error notification.

Summary

As a standard, IPSec is quickly becoming the preferred method for secure communications on TCP/IP networks. Designed to support multiple encryption and authentication schemes and multi-vendor interoperability, IPSec can be adapted to fit the security requirements of large and small organizations alike. Industries that rely on extranet technologies to communicate with their business partners will benefit from IPSec's flexible encryption and authentication schemes; large businesses will benefit from IPSec's scalability and centralized management; and every company can benefit from IPSec's virtual private networking (VPN) capabilities to support mobile workers, telecommuters, or branch offices accessing company resources via the Internet.

The Internet Security Protocol Architecture was designed with the future in mind and is garnering the support it deserves from the security and computer communities. Recent endorsements by major manufacturing associations like the Automotive Industry Action Group, product commitments from major vendors like Cisco Systems, as well as the establishment of a compliance certification program through the International Computer Security Association are clear signs that IPSec is well on its way to becoming the industry standard for business-to-business communications in the 21st century.

Wireless Internet Security

Dennis Seymour Lee

RECALLING THE EARLY DAYS OF THE INTERNET, ONE CAN RECOUNT SEVERAL REASONS WHY THE INTERNET CAME ABOUT. Some of these include:

- providing a vast communication medium to share electronic information
- creating a multiple-path network that could survive localized outages
- providing a means for computers from different manufacturers and different networks to talk to one another

Commerce and security, at that time, were not high on the agenda (with the exception of preserving network availability). The thought of commercializing the Internet in the early days was almost unheard of. In fact, it was considered improper etiquette to use the Internet to sell products and services. Commercial activity and their security needs are a more recent development on the Internet, having come about strongly in the past few years.

Today, in contrast, the wireless Internet is being designed from the very beginning with commerce as its main driving force. Nations and organizations around the globe are spending millions, even billions of dollars to buy infrastructure, transmission frequencies, technology, and applications in the hopes of drawing business. In some ways, this has become the "land rush" of the new millennium. It stands to reason then that security must play a critical role early on as well — where money changes hands, security will need to accompany this activity.

Although the wireless industry is still in its infancy, the devices, infrastructure, and application development for the wireless Internet are rapidly growing on a worldwide scale. Those with foresight will know that security must fit in early into these designs. The aim of this chapter is to highlight some of the significant security issues in this emerging industry that need addressing. These are concerns that any business wishing to

deploy a wireless Internet service or application will need to consider to protect their own businesses and their customers, and to safeguard their investments in this new frontier.

Incidentally, the focus of this chapter is not about accessing the Internet using laptops and wireless modems. That technology, which has been around for many years, in many cases, is an extension of traditional wired Internet access. Neither will this chapter focus on wireless LANs and Bluetooth, which are not necessarily Internet based, but deserve chapters on their own. Rather, the concentration is on portable Internet devices, which inherently have far less computing resources than regular PCs, such as cell phones and PDAs (personal digital assistants). Therefore, these devices require different programming languages, protocols, encryption methods, and security perspectives to cope with the different technology. It is important to note, however, that despite their smaller sizes and limitations, these devices have a significant impact on information security, mainly because of the electronic commerce and intranet-related applications that are being designed for them.

WHO IS USING THE WIRELESS INTERNET?

Many studies and estimates are available today that suggest the number of wireless Internet users will soon surpass the millions of wired Internet users. The assumption is based on the many more millions of worldwide cell phone users who are already out there, a population that grows by the thousands every day. If every one of these mobile users chose to access the Internet through cell phones, indeed that population could easily exceed the number of wired Internet users by several times. It is this very enormous potential that has many businesses devoting substantial resources and investments in the hopes of capitalizing on this growing industry.

The wireless Internet is still very young. Many mobile phone users do not yet have access to the Internet through their cell phones. Many are taking a "wait-and-see" attitude to see what services will be available. Most who do have wireless Internet access are early adopters who are experimenting with the potential of what this service could provide. Because of the severe limitations in the wireless devices — the tiny screens, the extremely limited bandwidth, as well as other issues — most users who have both wired and wireless Internet access will admit that, for today, the wireless devices will not replace their desktop computers and notebooks anytime soon as their primary means of accessing the Internet. Many admit that "surfing the Net" using a wireless device today could become a disappointing exercise. Most of these wireless Internet users have expressed the following frustrations:

- It is too slow to connect to the Internet.
- Mobile users can be disconnected in the middle of a session when they are on the move.
- It is cumbersome to type out sentences using a numeric keypad.
- It is expensive to use the wireless Internet, especially when billed on a per-minute basis.
- There is very little or no graphics display capabilities on wireless devices.
- The screens are too small and users have to scroll constantly to read a long message.
- There are frequent errors when surfing Web sites (mainly because most Web sites today are not yet wireless Internet compatible).

At the time of this writing, the one notable exception to these disappointments is found in Japan. The telecommunications provider NTT DoCoMo has experienced phenomenal growth in the number of wireless Internet subscribers, using a wireless application environment called i-Mode (as opposed to wireless application protocol, or WAP). For many in Japan, connection using a wireless phone is their only means of accessing the Internet. In many cases, wireless access to the Internet is far cheaper than wired access, especially in areas where the wired infrastructure is expensive to set up. I-Mode users have the benefit of "always online" wireless connections to the Internet, color displays on their cell phones, and even graphics, musical tones, and animation. Perhaps Japan's success with the wireless Internet will offer an example of what can be achieved in the wireless arena, given the right elements.

WHAT TYPES OF APPLICATIONS ARE AVAILABLE?

Recognizing the frustrations and limitations of today's wireless technology, many businesses are designing their wireless devices and services, not necessarily as replacements for wired Internet access, but as specialized services that extend what the wired Internet could offer. Most of these services highlight the attractive convenience of portable informational access, anytime and anywhere, without having to sit in front of a computer — essentially, Internet services one can carry in one's pocket. Clearly, the information would have to be concise, portable, useful, and easy to access. Examples of mobile services available or being designed today include:

- shopping online using a mobile phone; comparing online prices with store prices while inside an actual store
- getting current stock prices, trading price alerts, trade confirmations, and portfolio information anywhere
- performing bank transactions and obtaining account information
- obtaining travel schedules and booking reservations

- obtaining personalized news stories and weather forecasts
- receiving the latest lottery numbers
- obtaining the current delivery status for express packages
- reading and writing e-mail "on the go"
- accessing internal corporate databases such as inventory, client lists, etc.
- getting map directions
- finding the nearest ATM machines, restaurants, theaters, and stores, based on the user's present location
- dialing 911 and having emergency services quickly triangulate the caller's location
- browsing a Web site and speaking live with the site's representative, all within the same session

Newer and more innovative services are in the works. As any new and emerging technology, wireless services and applications are often surrounded by much hope and hype, as well as some healthy skepticism. But as the technology and services mature over time, yesterday's experiments can become tomorrow's standards. The Internet is a grand example of this evolving progress. Development of the wireless Internet will probably go through the same evolutionary cycle, although probably at an even faster pace.

Like any new technology, however, security and safety issues can damage its reputation and benefits if they are not included intelligently into the design from the very beginning. It is with this purpose in mind that this chapter is written.

Because the wireless Internet covers a lot of territory, the same goes for its security as well. This chapter discusses security issues as they relate to the wireless Internet in a few select categories, starting with transmission methods to the wireless devices and ending with some of the infrastructure components themselves.

HOW SECURE ARE THE TRANSMISSION METHODS?

For many years, it was public knowledge that analog cell phone transmissions are fairly easy to intercept. It has been a known problem for as long as analog cell phones have been available. They are easily intercepted using special radio scanning equipment. For this reason, as well as many others, many cell phone service providers have been promoting digital services to their subscribers and reducing analog to a legacy service.

Digital cell phone transmissions, on the other hand, are typically more difficult to intercept. It is on these very same digital transmissions that most of the new wireless Internet services are based.

However, there is no single method for digital cellular transmission. In fact, there are several different methods for wireless transmission available today. For example, in the United States, providers such as Verizon and Sprint primarily use CDMA (Code Division Multiple Access), whereas AT&T primarily uses TDMA (Time Division Multiple Access) and Voicestream uses GSM (Global Systems for Mobile Communications). Other providers, such as Cingular, offer more than one method (TDMA and GSM), depending on the geographic location. All these methods differ in the way they use the radio frequencies and the way they allocate users on those frequencies. This chapter discusses each of these in more detail.

Cell phone users are generally not concerned with choosing a particular transmission method if they want wireless Internet access, nor do they really care to. Instead, most users select their favorite wireless service provider when they sign up for service. It is generally transparent to the user which transmission method their provider has implemented. It is an entirely different matter for the service provider, however. Whichever method they implement has significant bearing on its infrastructure. For example, the type of radio equipment they use, the location and number of transmission towers to deploy, the amount of traffic they can handle, and the type of cell phones to sell to their subscribers are all directly related to the digital transmission method chosen.

Frequency Division Multiple Access (FDMA) Technology

All cellular communications, analog or digital, are transmitted using radio frequencies that are purchased by, or allocated to, the wireless service provider. Each service provider typically purchases licenses from the respective government to operate a spectrum of radio frequencies.

Analog cellular communications typically operate on what is called Frequency Division Multiple Access (or FDMA) technology. With FDMA, each service provider divides its spectrum of radio frequencies into individual frequency channels. Each channel is a specific frequency that supports a one-way communication session; and each channel has a width of 10 to 30 kilohertz (kHz). For a regular two-way phone conversation, every cell phone caller would be assigned two frequency channels: one to send and one to receive.

Because each phone conversation occupies two channels (two frequencies), it is not too difficult for specialized radio scanning equipment to tap into a live analog phone conversation once the equipment has tuned into the right frequency channel. There is very little privacy protection in analog cellular communications if no encryption is added.

Time Division Multiple Access (TDMA) Technology

Digital cellular signals, on the other hand, can operate on a variety of encoding techniques, most of which are resistant to analog radio frequency scanning. (Note that the word "encoding" in wireless communications does not mean encryption. "Encoding" here usually refers to converting a signal from one format to another; for example, from a wired signal to a wireless signal.)

One such technique is called time division multiple access, or TDMA. Similar to FDMA, TDMA typically divides the radio spectrum into multiple 30-kHz frequency channels (sometimes called frequency carriers). Every two-way communication requires two of these frequency channels: one to send and one to receive. But in addition, TDMA further subdivides each frequency channel into three to six time slots called voice/data channels, so that now up to six digital voice or data sessions can take place using the same frequency. With TDMA, a service provider can handle more calls at the same time compared to FDMA. This is accomplished by assigning each of the six sessions a specific time slot within the same frequency. Each time slot (or voice/data channel) is approximately seven milliseconds in duration. The time slots are arranged and transmitted over and over again in rapid rotation. Voice or data for each caller is placed into the time slot assigned to that caller and then transmitted. Information from the corresponding time slot is quickly extracted and reassembled at the receiving cellular base station to piece together the conversation or session. Once that time slot (or voice/data channel) is assigned to a caller, it is dedicated to that caller for the duration of the session, until it terminates. In TDMA, a user is not assigned an entire frequency, but shares the frequency with other users, each with an assigned time slot.

As of the writing of this chapter, there have not been many publicized cases of eavesdropping of TDMA phone conversations and data streams as they travel across the wireless space. Access to special types of equipment or test equipment would probably be required to perform such a feat. It is possible that an illegally modified TDMA cell phone could also do the job.

However, this does not mean that eavesdropping is unfeasible. With regard to a wireless Internet session, consider the full path that such a session takes. For a mobile user to communicate with an Internet Web site, a wireless data signal from the cell phone will eventually be converted into a wired signal before traversing the Internet itself. As a wired signal, the information can travel across the Internet in clear text until it reaches the Web site. Although the wireless signal itself may be difficult to intercept, once it becomes a wired signal, it is subject to the same interception vulnerabilities as all unencrypted communications traversing the Internet.

Always as a precaution, if there is confidential information being transmitted over the Internet, regardless of the method, it is necessary to encrypt that session from end-to-end. Encryption is discussed in a later chapter section.

Global Systems for Mobile Communications (GSM)

Another method of digital transmission is Global Systems for Mobile Communications (GSM). GSM is actually a term that covers more than just the transmission method alone. It covers the entire cellular system, from the assortment of GSM services to the actual GSM devices themselves. GSM is primarly used in European nations.

As a digital transmission method, GSM uses a variation of TDMA. Similar to FDMA and TDMA, the GSM service provider divides the allotted radio frequency spectrum into multiple frequency channels. This time, each frequency channel has a much larger width of 200 kHz. Again, similar to FDMA and TDMA, each GSM cellular phone uses two frequency channels: one to send and one to receive.

Like TDMA, GSM further subdivides each frequency channel into time slots called voice/data channels. However, with GSM, there are eight time slots, so that now up to eight digital voice or data sessions can take place using the same frequency. As for TDMA, once that time slot (or voice/data channel) is assigned to a caller, it is dedicated to that caller for the duration of the session, until it terminates.

GSM has additional features that enhance security. Each GSM phone uses a subscriber identity module (or SIM). A SIM can look like a credit-card sized smart card or a postage-stamp sized chip. This removable SIM is inserted into the GSM phone during usage. The smart card or chip contains information pertaining to the subscriber, such as the cell phone number belonging to the subscriber, authentication information, encryption keys, directory of phone numbers, and short saved messages belonging to that subscriber. Because the SIM is removable, the subscriber can take this SIM out of one phone and insert it into another GSM phone. The new phone with the SIM will then take on the identity of the subscriber. The user's identity is not tied to a particular phone but to the removable SIM itself. This makes it possible for a subscriber to use or upgrade to different GSM phones, without changing phone numbers. It is also possible to rent a GSM phone in another country, even if that country uses phones that transmit on different GSM frequencies. This arrangement works, of course, only if the GSM service providers from the different countries have compatible arrangements with each other.

The SIM functions as an authentication tool because the GSM phones are useless without it. Once the SIM is inserted into a phone, users are

prompted to put in their personal identification numbers (PINs) associated with that SIM (if the SIM is PIN-enabled). Without the correct PIN number, the phone will not work.

In addition to authenticating the user to the phone, the SIM is also used to authenticate the phone to the phone network itself during connection. Using the authentication (or Ki) key in the SIM, the phone authenticates to the service provider's Authentication Center during each call. The process employs a challenge-response technique, similar in some respects to using a token card to remotely log a PC onto a network.

The keys in the SIM have another purpose in addition to authentication. The encryption (or Kc) key generated by the SIM can be used to encrypt communications between the mobile phone and the service provider's transmission equipment for confidentiality. This encryption prevents eavesdropping, at least between these two points.

GSM transmissions, similar to TDMA, are difficult, but not impossible, to intercept using radio frequency scanning equipment. A frequency can have up to eight users on it, making the digital signals difficult to extract. By adding encryption using the SIM card, GSM can add yet another layer of security against interception.

However, when it comes to wireless Internet sessions, this form of encryption does not provide end-to-end protection. Only part of the path is actually protected. This is similar to the problem mentioned previously with TDMA Internet sessions. A typical wireless Internet session takes both a wireless and a wired path. GSM encryption protects only the path between the cell phone and the service provider's transmission site — the wireless portion. The remainder of the session through the wired Internet — from the service provider's site to the Internet Web site — can still travel in the clear. One would need to add end-to-end encryption if one needs to keep the entire Internet session confidential.

Code Division Multiple Access (CDMA) Technology

Another digital transmission method is called code division multiple access, or CDMA. CDMA is based on spread spectrum, a transmission technology that has been used by the U.S. military for many years to make radio communications more difficult to intercept and jam. Qualcomm is one of the main pioneers incorporating CDMA spread spectrum technology into the area of cellular phones.

Instead of dividing a spectrum of radio frequencies into narrow frequency bands or time slots, CDMA uses a very large portion of that radio spectrum, also called a frequency channel. The frequency channel has a wide width of 1.25 megahertz (MHz). For duplex communication, each cell

phone uses two of these wide CDMA frequency channels: one to send and one to receive.

During communication, each voice or data session is first converted into a series of data signals. Next, the signals are marked with a unique code to indicate that they belong to a particular caller. This code is called a pseudo-random noise (PN) code. Each mobile phone is assigned a new PN code by the base station at the beginning of each session. These coded signals are then transmitted by spreading them out across a very wide radio frequency spectrum. Because the channel width is very large, it has the capacity to handle many other user sessions at the same time, each session again tagged by unique PN codes to associate them to the appropriate caller.

A CDMA phone receives transmissions using the appropriate PN code to pick out the data signals that are destined for it and ignores all other encoded signals.

With CDMA, cell phones communicating with the base stations all share the same wide frequency channels. What distinguishes each caller is not the frequency used (as in FDMA), nor the time slot within a particular frequency (as in TDMA or GSM), but the PN noise code assigned to that caller. With CDMA, a voice/data channel is a data signal marked with a unique PN code.

Intercepting a single CDMA conversation would be difficult because its digital signals are spread out across a very large spectrum of radio frequencies. The conversation does not reside on just one frequency alone, making it difficult to scan. Also, without knowledge of the PN noise code, an eavesdropper would not be able to extract the relevant session from the many frequencies used. To further complicate interception, the entire channel width is populated by many other callers at the same time, creating a vast amount of noise for anyone trying to intercept the call.

However, as seen earlier with the other digital transmission methods, Internet sessions using CDMA cell phones are not impossible to intercept. As before, although the CDMA digital signals themselves can be difficult to intercept, once these wireless signals are converted into wired signals, the latter signals can be intercepted as they travel across the Internet. Without using end-to-end encryption, wireless Internet sessions are as vulnerable as other unencrypted communications traveling over the Internet.

Other Methods

There are additional digital transmission methods, many of which are derivatives of the types already discussed, and some of which are still under development. Some of these that are under development are called

third-generation or 3G transmission methods. Second-generation (2G) technologies, such as TDMA, GSM, and CDMA, offer transmission speeds of 9.6 to 14.4 Kbps (kilobits per second), which is slower than today's typical modem speeds. 3G technologies, on the other hand, are designed to transmit much faster and carry larger amounts of data. Some will be capable of providing high-speed Internet access as well as video transmission. Below is a partial listing of other digital transmission methods, including those in the 3G category.

- *iDEN* (Integrated Digital Enhanced Network) is based on TDMA and is a 2G transmission method. In addition to sending voice and data, it can also be used for two-way radio communications between two iDEN phones, much like walkie-talkies.
- *PDC* (Personal Digital Communications) is based on TDMA and is a 2G transmission method widely used in Japan.
- *GPRS* (General Packet Radio Service) is a 2.5G (not quite 3G) technology based on GSM. It is a packet-switched data technology that provides "always online" connections, which means that the subscriber can stay logged on to the phone network all day but uses it only if there is actual data to send or receive. Maximum data rates are estimated to be 115 Kbps.
- *EDGE* (Enhanced Data rates for Global Evolution) is a 3G technology based on TDMA and GSM. Like GPRS, it features "always online" connections using packet-switched data technologies. Maximum data rates are estimated to be 384 Kbps.
- *UMTS* (Universal Mobile Telecommunications System) is a 3G technology based on GSM. Maximum data rates are estimated at 2 Mbps (megabits per second).
- *CDMA2000* and *W-CDMA* (Wideband CDMA) are two 3G technologies based on CDMA. CDMA2000 is a more North American design, whereas W-CDMA is more European and Japanese oriented. Both provide maximum data rates estimated at 384 Kbps for slow-moving mobile units, and at 2 Mbps for stationary units.

Regardless of the methods or the speeds, the need for end-to-end encryption will still be a requirement if confidentiality is needed between the mobile device and the Internet or intranet site. Because wireless Internet communications encompass both wireless and wired-based transmissions, encryption features covering just the wireless portion of the communication is clearly not enough. For end-to-end privacy protection, the applications and the protocols have a role to play, as discussed later in this chapter.

HOW SECURE ARE WIRELESS DEVICES?

Internet security, as many have seen it applied to corporate networks today, can be difficult to implement on wireless phones and PDAs for a

variety of reasons. Most of these devices have limited CPUs, memory, bandwidth, and storage abilities. As a result, many have disappointingly slow and limited computing power. Robust security features that can take less than a second to process on a typical workstation can take potentially many minutes on a wireless device, making them impractical or inconvenient for the mobile user. Because many of these devices have merely a fraction of the hardware capabilities found on typical workstations, the security features on portable devices are often lightweight or even nonexistent — from an Internet security perspective. However, these same devices are now being used to log into sensitive corporate intranets, or to conduct mobile commerce and banking. Although these wireless devices are smaller in every way, their security needs are just as significant as before. It would be a mistake for corporate IT and information security departments to ignore these devices as they start to populate the corporate network. After all, these devices do not discriminate; they can be designed to tap into the same corporate assets as any other node on a network. Some of the security aspects as they relate to these devices are examined here.

Authentication

The process of authenticating wireless phone users has gone through many years of implementation and evolution. It is probably one of the most reliable security features digital cell phones have today, given the many years of experience service providers have had in trying to reduce the theft of wireless services. Because the service providers have a vested interest in knowing who to charge for the use of their services, authenticating the mobile user is of utmost importance.

As previously mentioned, GSM phones use SIM cards or chips that contain authentication information about the user. SIMs typically carry authentication and encryption keys, authentication algorithms, identification information, phone numbers belonging to the subscriber, etc. They allow users to authenticate to their own phones and to the phone network to which they are subscribed.

In North America, TDMA and CDMA phones use a similarly complex method of authentication as in GSM. Like GSM, the process incorporates keys, Authentication Centers, and challenge-response techniques. However, because TDMA and CDMA phones do not generally use removable SIM cards or chips, instead, these phones rely on the authentication information embedded into the handset. The user's identity is therefore tied to the single mobile phone itself.

The obvious drawback is that for authentication purposes, TDMA and CDMA phones offer less flexibility when compared to GSM phones. To

deploy a new authentication feature with a GSM phone, in many cases, all that is needed is to update the SIM card or chip. On the other hand, with TDMA and CDMA, deploying new authentication features would probably require users to buy new cell phones — a more expensive way to go. Because it is easier to update a removable chip than an entire cell phone, it is likely that one will find more security features and innovations being offered for GSM as a result.

One important note, however, is that this form of authentication does not necessarily apply to Internet-related transactions. It merely authenticates the mobile user to the service provider's phone network, which is only one part of the transmission if one is talking about Internet transactions. For securing end-to-end Internet transactions, mobile users still need to authenticate the Internet Web servers they are connecting to, to verify that indeed the servers are legitimate. Likewise, the Internet Web servers need to authenticate the mobile users that are connecting to it, to verify that they are legitimate users and not impostors. The wireless service providers, however, are seldom involved in providing full end-to-end authentication service, from mobile phone to Internet Web site. That responsibility usually falls to the owners of the Internet Web servers and applications.

Several methods for providing end-to-end authentication are being tried today at the application level. Most secure mobile commerce applications are using IDs and passwords, an old standby, which of course has its limitations because it provides only single-factor authentication. Other organizations are experimenting with GSM SIMs by adding additional security ingredients such as public/private key pairs, digital certificates, and other public key infrastructure (PKI) components into the SIMs. However, because the use of digital certificates can be process intensive, cell phones and hand-held devices typically use lightweight versions of these security components. To accommodate the smaller processors in wireless devices, the digital certificates and their associated public keys may be smaller or weaker than those typically deployed on desktop Web browsers, depending on the resources available on the wireless device.

Additionally, other organizations are experimenting with using elliptic-curve cryptography (ECC) for authentication, digital certificates, and public key encryption on the wireless devices. ECC is an ideal tool for mobile devices because it can offer strong encryption capabilities but requires less computing resources than other popular forms of public key encryption. Certicom is one of the main pioneers incorporating ECC for use on wireless devices.

As more and more developments take place with wireless Internet authentication, it becomes clear that, in time, these Internet mobile devices will become full-fledged authentication devices, much like tokens,

smart cards, and bank ATM cards. If users begin conducting Internet commerce using these enhanced mobile devices, securing those devices themselves from loss or theft now becomes a priority. With identity information embedded into the devices or the removable SIMs, losing these could mean that an impostor can now conduct electronic commerce transactions using that stolen identity. With a mobile device, the user, of course, plays the biggest role in maintaining its overall security. Losing a cell phone that has Internet access and an embedded public/private key pair can be potentially as disastrous as losing a bank ATM card with its associated PIN written on it, or worse. If a user loses such a device, contacting the service provider immediately about the loss and suspending its use is a must.

Confidentiality

Preserving confidentiality on wireless devices poses several interesting challenges. Typically, when one accesses a Web site with a browser and enters a password to gain entry, the password one types is masked with asterisks or some other placeholder to prevent others from seeing the actual password on one's screen. With cell phones and hand-held devices, masking the password could create problems during typing. With cell phones, letters are often entered using the numeric keypad, a method that is cumbersome and tedious for many users. For example, to type the letter "R," one must press the number 7 key three times to get to the right letter. If the result is masked, it is not clear to the user what letter was actually submitted. Because of this inconvenience, some mobile Internet applications do away with masking so that the entire password is displayed on the screen in the original letters. Other applications initially display each letter of the password for a few seconds as they are being entered, before masking each with a placeholder afterward. This gives the user some positive indication that the correct letters were indeed entered, while still preserving the need to mask the password on the device's screen for privacy. The latter approach is probably the more sensible of the two, and should be the one that application designers adopt.

Another challenge to preserving confidentiality is making sure that confidential information such as passwords and credit card numbers are purged from the mobile device's memory after they are used. Many times, such sensitive information is stored as variables by the wireless Internet application and subsequently cached in the memory of the device. There have been documented cases in which credit card numbers left in the memory of cell phones were reusable by other people who borrowed the same phones to access the same sites. Once again, the application designers are the chief architects in preserving the confidentiality here. It is important that programmers design an application to clear the mobile

device's memory of sensitive information when the user finishes using that application. Although leaving such information in the memory of the device may spare the user of having to re-enter it the next time, it is, however, as risky as writing the associated PIN or password on a bank ATM card itself.

Yet another challenge in preserving confidentiality is making sure that sensitive information is kept private as it travels from the wireless device to its destination on the Internet, and back. Traditionally, for the wired Internet, most Web sites use Secure Sockets Layer (SSL) or its successor, Transport Layer Security (TLS), to encrypt the entire path end-to-end, from the client to the Web server. However, many wireless devices, particularly cell phones, lack the computing power and bandwidth to run SSL efficiently. One of the main components of SSL is RSA public key encryption. Depending on the encryption strength applied at the Web site, this form of public key encryption can be processor and bandwidth intensive, and can tax the mobile device to the point where the communication session itself becomes too slow to be practical.

Instead, wireless Internet applications that are developed using the Wireless Application Protocol (WAP) use a combination of security protocols. Secure WAP applications use both SSL and WTLS (Wireless Transport Layer Security) to protect different segments of a secure transmission. Typically, SSL protects the wired portion of the connection and WTLS primarily protects the wireless portion. Both are needed to provide the equivalent of end-to-end encryption.

WTLS is similar to SSL in operation. However, although WTLS can support either RSA or ECC, ECC is probably preferred because it provides strong encryption capabilities but is more compact and faster than RSA.

WTLS has other differences from SSL as well. WTLS is built to provide encryption services for a slower and less resource-intensive environment, whereas SSL could tax such an environment. This is because SSL encryption requires a reliable transport protocol, particularly TCP (Transmission Control Protocol, a part of TCP/IP). TCP provides error detection, communication acknowledgments, and retransmission features to ensure reliable network connections back and forth. But because of these features, TCP requires more bandwidth and resources than what typical wireless connections and devices can provide. Most mobile connections today are low bandwidth and slow, and not designed to handle the constant, back and forth error-detection traffic that TCP creates.

Realizing these limitations, the WAP Forum, the group responsible for putting together the standards for WAP, designed a supplementary protocol stack that is more suitable for the wireless environment. Because this environment typically has low connection speeds, low reliability, and low

bandwidth, in order to compensate, the protocol stack uses compressed binary data sessions and is more tolerant of intermittent coverage. The WAP protocol stack resides in layers 4, 5, 6, and 7 of the OSI reference model. The WAP protocol stack works with UDP (User Datagram Protocol) for IP-based networks and WDP (Wireless Datagram Protocol) for non-IP networks. WTLS, which is the security protocol from the WAP protocol stack, can be used to protect UDP or WDP traffic in the wireless environment.

Because of these differences between WTLS and SSL, as well as the different underlying environments that they work within, an intermediary device such as a gateway is needed to translate the traffic going from one environment into the next. This gateway is typically called a WAP gateway. The WAP gateway is discussed in more detail in the infrastructure section below.

Malicious Code and Viruses

The number of security attacks on wireless devices has been small compared to the many attacks against workstations and servers. This is due, in part, to the very simple fact that most mobile devices, particularly cell phones, lack sufficient processors, memory, or storage that malicious code and viruses could exploit. For example, a popular method for spreading viruses today is by hiding them in file attachments to e-mail. However, many mobile devices, particularly cell phones, lack the ability to store or open e-mail attachments. This makes mobile devices relatively unattractive as targets because the damage potential is relatively small.

However, mobile devices are still vulnerable to attack and will become increasingly more so as they evolve with greater computing, memory, and storage capabilities. With greater speeds, faster downloading abilities, and better processing, mobile devices can soon become the equivalent of today's workstations, with all their exploitable vulnerabilities. As of the writing of this chapter, cell phone manufacturers were already announcing that the next generation of mobile phones will support languages such as Java so that users can download software programs such as organizers, calculators, and games onto their Web-enabled phones. However, on the negative side, this also opens up more opportunities for users to unwittingly download malicious programs (or "malware") onto their own devices. The following adage applies to mobile devices: "The more brains they have, the more attractive they become as targets."

HOW SECURE ARE THE NETWORK INFRASTRUCTURE COMPONENTS?

As many of us who have worked in the information security field know, security is usually assembled using many components, but its overall strength is only as good as its weakest link. Sometimes it does not matter

if one is using the strongest encryption available over the network and the strongest authentication at the devices. If there is a weak link anywhere along the chain, attackers will focus on this vulnerability and may eventually exploit it, choosing a path that requires the least effort and the least amount of resources.

Because the wireless Internet world is still relatively young and a work in progress, vulnerabilities abound, depending on the technology one has implemented. This chapter section focuses on some infrastructure vulnerabilities for those who are using WAP (Wireless Application Protocol).

The "Gap in WAP"

Encryption has been an invaluable tool in the world of E-commerce. Many online businesses use SSL (Secure Sockets Layer) or TLS (Transport Layer Security) to provide end-to-end encryption to protect Internet transactions between the client and the Web server.

When using WAP however, if encryption is activated for the session, there are usually two zones of encryption applied, each protecting the two different halves of the transmission. SSL or TLS is generally used to protect the first path, between the Web server and an important network device called the WAP gateway that was previously mentioned. WTLS (Wireless Transport Layer Security) is used to protect the second path, between the WAP gateway and the wireless mobile device.

The WAP gateway is an infrastructure component needed to convert wired signals into a less bandwidth-intensive and compressed binary format, compatible for wireless transmissions. If encryption such as SSL is used during a session, the WAP gateway will need to translate the SSL-protected transmission by decrypting this SSL traffic and re-encrypting it with WTLS, and vice versa in the other direction. This translation can take just a few seconds; but during this brief period, the data sits in the memory of the WAP gateway decrypted and in the clear before it is re-encrypted using the second protocol. This brief period in the WAP gateway — some have called it the "gap in WAP" — is an exploitable vulnerability. It depends on where the WAP gateway is located, how well it is secured, and who is in charge of protecting it.

Clearly, the WAP gateway should be placed in a secure environment. Otherwise, an intruder attempting to access the gateway can steal sensitive data while it transitions in clear text. The intruder can also sabotage the encryption at the gateway, or even initiate a denial-of-service or other malicious attack on this critical network component. In addition to securing the WAP gateway from unauthorized access, proper operating procedures should also be applied to enhance its security. For example, it is wise not to save any of the clear-text data onto disk storage during the

decryption and re-encryption process. Saving this data onto log files, for example, could create an unnecessarily tempting target for intruders. In addition, the decryption and re-encryption should operate in memory only and proceed as quickly as possible. Furthermore, to prevent accidental disclosure, the memory should be properly overwritten, thereby purging any sensitive data before that memory is reused.

WAP Gateway Architectures

Depending on the sensitivity of the data and the liability for its unauthorized disclosure, businesses offering secure wireless applications (as well as their customers) may have concerns about where the WAP gateway is situated, how it is protected, and who is protecting it. Three possible architectures and their security implications are examined:

WAP Gateway at the Service Provider. In most cases, the WAP gateways are owned and operated by the wireless service providers. Many businesses that deploy secure wireless applications today rely on the service provider's WAP gateway to perform the SSL-to-WTLS encryption translation. This implies that the business owners of the sensitive wireless applications, as well as their users, are entrusting the wireless service providers to keep the WAP gateway and the sensitive data that passes through it safe and secure. Exhibit 8-1 provides an example of such a setup, where the WAP gateway resides within the service provider's secure environment. If encryption is applied in a session between the user's cell phone and the application server behind the business' firewall, the path between the cell phone and the service provider's WAP gateway is typically encrypted using WTLS. The path between the WAP gateway and the business host's application server is encrypted using SSL or TLS.

A business deploying secure WAP applications using this setup should realize, however, that it cannot guarantee end-to-end security for the data because it is decrypted, exposed in clear text for a brief moment, and then re-encrypted, all at an outside gateway that is away from its control. The WAP gateway is generally housed in the wireless service provider's data center and attended by those who are not directly accountable to the businesses. Of course, it is in the best interest of the service provider to maintain the WAP gateway in a secure manner and location.

Sometimes, to help reinforce that trust, businesses may wish to conduct periodic security audits on the service provider's operation of the WAP gateways to ensure that the risks are minimized. Bear in mind, however, that by choosing this path, the business may need to inspect many WAP gateways from many different service providers. A service provider sets up the WAP gateway primarily to provide Internet access to its own wireless phone subscribers. If users are dialing into a business' secure Web

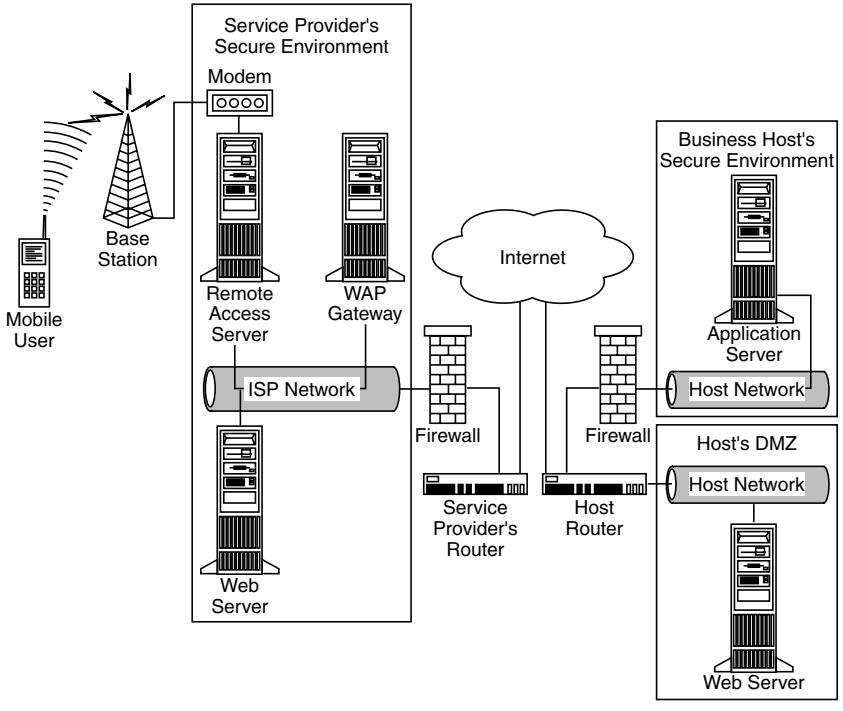


Exhibit 8-1. WAP gateway at the service provider.

site, for example, from 20 different wireless service providers around the world, then the business may need to audit the WAP gateways belonging to these 20 providers. This, unfortunately, is a formidable task and an impractical method of ensuring security. Each service provider might apply a different method for protecting its own WAP gateway — if protected at all. Furthermore, in many cases, the wireless service providers are accountable to their own cell phone subscribers, not necessarily to the countless businesses that are hosting secure Internet applications, unless there is a contractual arrangement to do so.

WAP Gateway at the Host. Some businesses and organizations, particularly in the financial, healthcare, and government sectors, may have legal requirements to keep their customers' sensitive data protected. Having such sensitive data exposed outside the organization's internal control may pose an unnecessary risk and liability. To some, the "gap in WAP" presents a broken pipeline, an obvious breach of confidentiality that is just waiting to be exploited. For those who find such a breach unacceptable, one possible solution is to place the WAP gateway at the business host's own protected network, bypassing the wireless service provider's

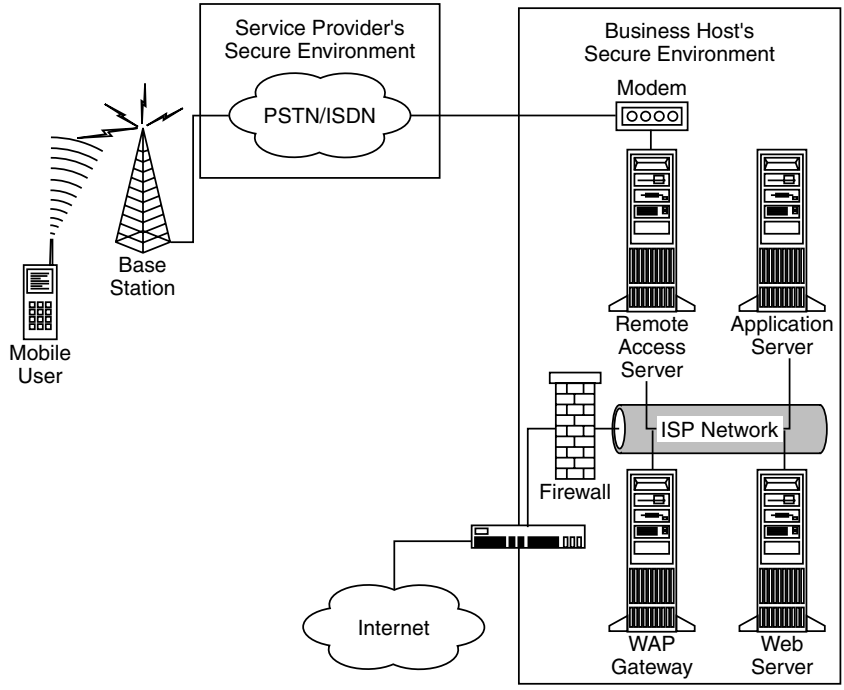


Exhibit 8-2. WAP gateway at the host.

WAP gateway entirely. Exhibit 8-2 provides an example of such a setup. Nokia, Ericsson, and Ariel Communications are just a few of the vendors offering such a solution.

This approach has the benefit of keeping the WAP gateway and its WTLS-SSL translation process in a trusted location, within the confines of the same organization that is providing the secure Web applications. Using this setup, users are typically dialing directly from their wireless devices, through their service provider's Public Switched Telephone Network (PSTN), and into the business' own Remote Access Servers (RAS). Once they reach the RAS, the transmission continues onto the WAP gateway, and then onward to the application or Web server, all of these devices within the business host's own secure environment.

Although it provides better end-to-end security, the drawback to this approach is that the business host will need to set up banks of modems and RAS so users have enough access points to dial in. The business will also need to reconfigure the users' cell phones and PDAs to point directly to the business' own WAP gateway instead of typically to the service provider's. However, not all cell phones allow this reconfiguration by the

user. Furthermore, some cell phones can point to only one WAP gateway, while others are fortunate enough to point to more than one. In either case, individually reconfiguring all those wireless devices to point to the business' own WAP gateway may take significant time and effort.

For users whose cell phones can point to only a single WAP gateway, this reconfiguration introduces yet another issue. If these users now want to access other WAP sites across the Internet, they still must go through the business host's WAP gateway first. If the host allows outgoing traffic to the Internet, the host then becomes an Internet service provider (ISP) to these users who are newly configured to point to the host's own WAP gateway. Acting as a makeshift ISP, the host will inevitably need to attend to service- and user-related issues, which to many businesses can be an unwanted burden because of the significant resources required.

Pass-Through from Service Provider's WAP Gateway to Host's WAP Proxy. For those businesses that want to provide secure end-to-end encrypted transactions, yet want to avoid the administrative headaches of setting up their own WAP gateways, there are other approaches. One such approach, as shown in Exhibit 8-3, is to keep the WTLS-encrypted data unchanged as it goes from the user's mobile device and through the service provider's WAP gateway. The WTLS-SSL encryption translation will not occur until the encrypted data reaches a second WAP gateway-like device residing within the business host's own secure network. One vendor developing such a solution is Openwave Systems (a combination of Phone.com and Software.com). Openwave calls this second WAP gateway-like device the Secure Enterprise Proxy. During an encrypted session, the service provider's WAP gateway and the business' Secure Enterprise Proxy negotiate with each other, so that the service provider essentially passes the encrypted data unchanged onto the business that is using this Proxy. This solution utilizes the service provider's WAP gateway because it is still needed to provide proper Internet access for the mobile users, but it does not perform the WTLS-SSL encryption translation there and thus is not exposing confidential data. The decryption is passed on and occurs, instead, within the confines of the business' own secure network, either at the Secure Enterprise Proxy or at the application server.

One drawback to this approach, however, is its proprietary nature. At the time of this writing, to make the Openwave solution work, three parties would need to implement components exclusively from Openwave. The wireless service providers would need to use Openwave's latest WAP gateway. Likewise, the business hosting the secure applications would need to use Openwave's Secure Enterprise Proxy to negotiate the encryption pass-through with that gateway. In addition, the mobile devices themselves would need to use Openwave's latest Web browser,

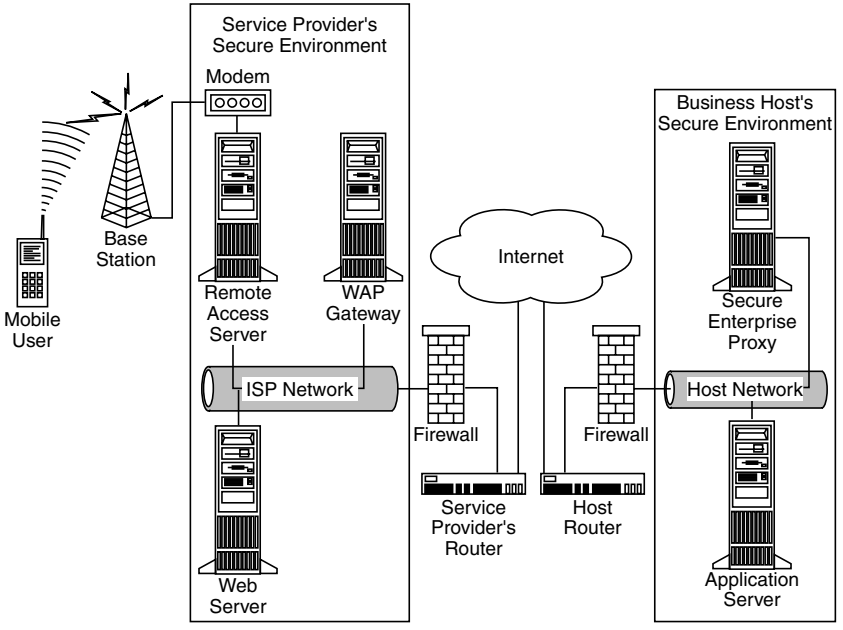


Exhibit 8-3. Pass-through from service provider's WAP gateway to host's WAP proxy.

at least Micro-browser version 5. Although approximately 70 percent of WAP-enabled phones throughout the world are using some version of Openwave Micro-browser, most of these phones are using either version 3 or 4. Unfortunately, most of these existing browsers are not upgradable by the user, so most users may need to buy new cell phones to incorporate this solution. It may take some time before this solution comes to fruition and becomes popular.

These are not the only solutions for providing end-to-end encryption for wireless Internet devices. Other methods in the works include applying encryption at the applications level, adding encryption keys and algorithms to cell phone SIM cards, and adding stronger encryption techniques to the next revisions of the WAP specifications, perhaps eliminating the "gap in WAP" entirely.

CONCLUSION

Two sound recommendations for the many practitioners in the information security profession are:

- Stay abreast of the wireless security issues and solutions.
- Do not ignore the wireless devices.

Many in the IT and information security professions regard the new wireless Internet devices diminutively as personal gadgets or executive toys. Many are so busy grappling with the issues of protecting their corporate PCs, servers, and networks that they cannot imagine worrying about yet another class of devices. Many corporate security policies make no mention about securing mobile hand-held devices and cell phones, although some of these same corporations are already using these devices to access their own internal e-mail. The common fallacy heard is: because these devices are so small, what harm can such a tiny device create?

Security departments have had to wrestle with the migration of information assets from the mainframe world to distributed PC computing. Many corporate attitudes have had to change during that evolution regarding where to apply security. With no exaggeration, corporate computing is undergoing yet another significant phase of migration. It is not so much that corporate information assets can be accessed through wireless means, because wireless notebook computers have been doing that for years; rather, the means of access will become ever cheaper and, hence, greater in volume. Instead of using a $3000 notebook computer, users (or intruders) can now tap into a sensitive corporate network from anywhere, using just a $40 Internet-enabled cell phone. Over time, these mobile devices will have increasing processing power, memory, bandwidth, storage, ease of use, and finally, popularity. It is this last item that will inevitably draw upon the corporate resources.

Small as these devices may be, once they access the sensitive assets of an organization, they can do as much good or harm as any other computer. Ignoring or disallowing these devices from an information security perspective has two probable consequences. First, the business units or executives within the organization will push, and often successfully, to deploy wireless devices and services anyway, but shutting out any involvement or guidance from the information security department. Inevitably, information security will be involved at a much later date, but reactively and often too late to have any significant impact on proper design and planning.

Second, by ignoring the wireless devices and their capabilities, the information security department will give attackers just what they need — a neglected and unprotected window into an otherwise fortified environment. Such an organization will be caught unprepared when an attack using wireless devices surfaces.

Wireless devices should not be treated as mere gadgets or annoyances. Once they tap into the valued assets of an organization, they are indiscriminate and equal to any other node on the network. To stay truly informed and prepared, information security practitioners should stay

abreast of the news developments and security issues regarding wireless technology. In addition, they need to work with the application designers as an alliance to ensure that applications designed for wireless take into consideration the many points discussed in this chapter. And finally, organizations need to expand the categories of devices protected under their information security policies to include wireless devices because they are, effectively, yet another infrastructure component of the organization.

Bibliography

Books:

1. Blake, Roy, *Wireless Communication Technology*, Delmar Thomson Learning, 2001.
2. Harte, Lawrence et al., *Cellular and PCS: The Big Picture*, McGraw-Hill, 1997.
3. Howell, Ric et al., *Professional WAP,* Wrox Press Ltd., 2000.
4. Muller, Nathan J., *Desktop Encyclopedia of Telecommunications, second edition*, McGraw-Hill, 2000.
5. Tulloch, Mitch, *Microsoft Encyclopedia of Networking,* Microsoft Press, 2000.
6. Van der Heijden, Marcel and Taylor, Marcus, *Understanding WAP: Wireless Applications, Devices, and Services,* Artech House Publishers, 2000.

Articles and white papers:

1. Saarinen Markku-Juhani, *Attacks Against the WAP WTLS Protocol,* University of Jvyskyl, Finland.
2. Saita, Anne, Case Study: Securing Thin Air, Academia Seeks Better Security Solutions for Handheld Wireless Devices, http://www.infosecuritymag.com, April 2001.
3. Complete WAP Security from Certicom, http://www.certicom.com.
4. Radding, Alan, Crossing the Wireless Security Gap, http://www.computerworld.com, Jan. 1, 2001.
5. Does Java Solve Worldwide WAP Wait?, http://www.unstrung.com, April 9, 2001.
6. DeJesus, Edmund X., "Locking Down the… Wireless Devices Are Flooding the Airwaves with Millions of Bits of Information. Securing Those Transmissions Is the Next Challenge Facing E-Commerce, http://www.infosecuritymag.com, Oct. 2000.
7. Izarek, Stephanie, Next-Gen Cell Phones Could Be Targets for Viruses, http://www.foxnews.com, June 1, 2000.
8. Nobel, Carmen, Phone.com Plugs WAP Security Hole, *eWEEK,* September 25, 2000.
9. Secure Corporate WAP Services: Nokia Activ Server, http://www.nokia.com.
10. Schwartz, Ephraim, Two-Zone Wireless Security System Creates a Big Hole in Your Communications, http://www.infoworld.com, Nov. 6, 2000.
11. Appleby, Timothy P., WAP — The Wireless Application Protocol (White Paper), Global Integrity.
12. Wireless Devices Present New Security Challenges — Growth in Wireless Internet Access Means Handhelds Will Be Targets of More Attacks, CMP Media, Inc., Oct 21, 2000.

40

VPN Deployment and Evaluation Strategy

Keith Pasley, CISSP

VPN technology has rapidly improved in recent years in the areas of performance, ease of use, deployment, and management tool effectiveness. The market demand for virtual private network (VPN) technology is also rapidly growing. Similarly, the number of different VPN products is increasing. The promise of cost savings is being met. However, there is a new promise that approaches VPNs from both a technical and business perspective. In today's fast-paced business environment, the promises of ease of management, deployability, and scalability of VPN systems are the critical success factors when it comes to selecting and implementing the right VPN system. From a business perspective, the realized benefits include:

- Competitive advantage due to closer relationships with business partners and customers
- New channels of service delivery
- Reaching new markets with less cost
- Offering higher-value information with removal of security concerns that have hampered this effort in the past

With so many choices, how does one determine the best fit? Objective criteria are needed to make a fair assessment of vendor product claims. What should one look for when evaluating a vendor's performance claims? What else can add value to VPN systems? In some cases, outsourcing to a managed security service provider is an option. Managed security service providers are service outsourcers that typically host security applications and offer transaction-based use of the hosted security application. Many businesses are now seriously considering outsourcing VPNs to managed security service providers that can provide deployment and management. The perception is that managed service providers have the expertise and management infrastructure to operate large-scale VPNs better than in-house staff.

VPN performance has consistently improved in newer versions of VPN products. Although performance is important, is it the most important criterion in selecting a VPN solution? No. A fast but exploitable VPN implementation will not improve security. Performance is also difficult to evaluate, and many performance tests do a poor job of mimicking real-world situations. Vendor performance claims should be evaluated very closely due to overly optimistic marketing-oriented performance claims that do not pan out in real-world implementations. It is important to understand the test methodologies used by vendors as the basis for such performance claims.

This chapter provides answers to a number of issues that information security professionals face when selecting products and implementing VPNs.

What is a VPN?

VPNs allow private information to be transferred across a public network such as the Internet. A VPN is an extension of the network perimeter, and therefore must have the ability to uniformly enforce the network security policy across all VPN entry points. Through the use of encapsulation and encryption, the confiden-

tiality of the data is protected as it traverses a public network. Technical benefits of proper use of this technology include reduced business operational costs, increased security of network access, in-transit data integrity, user and data authentication, and data confidentiality. However, some of the financial benefits can be negated by the real costs of a VPN system, which are incurred after the purchase of a VPN solution, during deployment, ongoing management, and support. The new promise of manageability, deployability, and scalability offers vendors an opportunity to differentiate their products from their competitors'. This type of product differentiation is increasingly important because most vendors' VPN products use the same VPN protocol — IPSec — and other underlying technologies. IPSec is an international standard that defines security extensions to the Internet Protocol. Although there are other secure tunneling protocols used to implement VPNs, IPSec has taken the leadership position as the protocol of choice. This standard specifies mandatory features that provide for a minimal level of vendor interoperability. This chapter will help information security professionals sort out a set of criteria that can be used when evaluating IPSec VPN solutions. The discussion begins with an examination of VPN applications.

IPSec VPN Applications

Enterprises have typically looked to virtual private networks (VPNs) to satisfy four application requirements: remote access, site-to-site intranet, secure extranet, and secured internal network. The technical objective, in most cases, is to provide authorized users with controlled access to protected network data resources (i.e., server files, disk shares, etc.). A companion business objective is to manage down network infrastructure costs and increase the efficiency of internal and external business information flow, increasing user productivity, competitive advantage, or strength of business partner relationships.

It is a good idea to define the tasks involved in a VPN evaluation project. A task list will help keep the evaluation focused and help anticipate the resources needed to complete the evaluation. Exhibit 40.1 gives an example list of VPN evaluation project tasks.

Remote Access VPN

There are two parts to a remote access VPN: the server and the client. They have two different roles and therefore two different evaluation criteria.

- B*usiness goal:* lower telecom costs, increased employee productivity
- T*echnical goal:* provide secured same-as-on-the-LAN access to remote workers

Both roles and criteria are discussed in this chapter section.

Remote access IPSec VPNs enable users to access corporate resources whenever, wherever, and however they require. Remote access VPNs encompass analog, dial, ISDN, digital subscriber line (DSL), mobile IP, and cable Internet access technologies, combined with security protocols such as IPSec to securely connect mobile users and telecommuters.

EXHIBIT 40.1 VPN evaluation project tasks

Assess data security requirements.
Classify users.
Assess user locations.
Determine the networking connectivity and access requirements.
Choose product or a service provider.
Assess hardware/software needs.
Set up a test lab.
Obtain evaluation devices.
Test products based on feature requirements.
Implement a pilot program.

The Client Software

Remote access users include telecommuters, mobile workers, traveling employees, and any other person who is an employee of the company whose data is being accessed. The most frequently used operating systems are MS Windows based, due to its market acceptance as a corporate desktop standard. IPSec VPN system requirements may indicate support for other operating systems, such as Macintosh, UNIX, PalmOS, or Microsoft Pocket PC/Windows CE. Preferably, the IPSec VPN vendor offers a mix of client types required by company. Mobile workers sometimes require access to high-value/high-risk corporate data such as sales forecasts, confidential patient or legal information, customer lists, and sensitive but unclassified DoD or law enforcement information. Remote access can also mean peer-to-peer access for information collaboration across the Internet (e.g., Microsoft NetMeeting) and can also be used for remote technical support.

The client hardware platforms for this application include PDAs, laptops, home desktop PC, pagers, data-ready cell phones, and other wired and wireless networked devices. As hardware platform technology evolves, there are sure to be other devices that can be used to remotely access company data. An interesting phenomenon that is increasing in popularity is the use of wireless devices such as personal digital assistants, cell phones, and other highly portable network-capable devices as access platforms for remote access IPSec VPN applications. The issues facing wireless devices include the same basic issues that wired IPSec VPN platforms face, such as physical security and data security, with the added issue of implementing encryption in computationally challenged devices.

Another issue with wireless IPSec VPN platforms, such as PDAs, is compatibility with wired-world security protocols. The Wireless Application Protocol (WAP) Forum, a standards body for wireless protocols, is working to improve compatibility between the WAP-defined security protocol — Wireless Transport Layer Security (WTLS) — and wired-world security protocols, such as SSL. Industry observers estimate that wireless devices such as PDAs and data-ready cell phones will be the platform of choice for applications that require remote, transactional data access. However, these devices are small and can easily be stolen or lost. This emphasizes the need to include hardware-platform physical security as part of the evaluation criteria when analyzing the features of IPSec VPN client software. Physical security controls for these platforms can include cables and locks, serial number tracking, motion sensors, location-based tracking (via the use of Global Positioning Systems), and biometric authentication such as finger scan with voice verification.

The communications transport for remote access continues to be predominately via dial-up. Wireless and broadband access continue to grow in usage. However, early complexities in broadband implementations and certain geographic constraints have recently been mitigated, and it is likely that broadband and wireless may grow in usage beyond dial-up use.

One issue with broadband (DSL, cable modem) usage is that as it becomes a commodity, broadband providers may try to segment allowable services on their networks. One tactic that is being used by cable services that provide Internet access is to prohibit the use of IPSec VPNs by residential users. According to one cable company, based on the U.S. West Coast, the network overhead generated by residential IPSec VPN users was affecting its available bandwidth to other home-based users. Therefore, this cable company had prohibited all VPNs from being used by its residential service customers through the use of port and protocol packet filter rules in the cable modem. Obviously, this benefits the cable company because it can then charge higher business-class fees to route VPNs from home users through the Internet. Some vendors of proprietary VPN solutions have responded by using encapsulation of VPN payloads into allowed protocols, within HTTP packets for example, to bypass this cable company constraint. How this issue will be resolved remains to be seen, but it does identify another criterion when selecting a VPN: will it work over the end user's ISP or network access provider network? Will the remote end users use their own residential class ISP? Or will the company purchase business-class access to ensure consistent and reliable connectivity?

End users are focused on getting done the work they are paid to do. Users, in general, are not incentived to really care about the security of their remote access connection. Users are primarily concerned with ease of use, reliability, and compatibility with existing applications on their computers.

Therefore, a part of a comprehensive evaluation strategy is that the VPN client should be fully tested on the same remote platform configuration as will be used by the users in real life. For example, some vendors' personal firewall may cause a conflict with another vendor's IPSec VPN client. This type of incompatibility may or may not be resolvable by working with the vendor and may result in disqualification from a list of potential solutions. Another example of IPSec VPN client incompatibility is the case in which one vendor's

IPSec VPN client does not support the same parameters as, say, the IPSec VPN server or another IPSec VPN client. The thing to keep in mind here is that standards usually define a minimum level of mandatory characteristics. Vendors, in an effort to differentiate their products, may add more advanced features, features not explicitly defined by a standard. Also, vendors may optimize their IPSec VPN client to work most effectively with their own IPSec VPN server. This leaves a mixed vendor approach to use a "lowest common denominator" configuration that may decrease the level of security and performance of the overall IPSec VPN system. For example, some IPSec VPN server vendors support authentication protocols that are not explicitly defined as mandatory in the standard. Obviously, if the IPSec VPN client that is selected is not from the same vendor as the IPSec VPN server and acceptable interoperability cannot be attained, then a compromise in criteria or vendor disqualification would be the decision that would have to be made.

As Internet access becomes more pervasive and subscribers stay connected longer or "always," there are resultant increases in attack opportunity against the remote VPN user's computer. Therefore, if there is valuable data stored on the remote user's computer, it may make sense to use some form of file or disk encryption. Because encryption is a processor-intensive activity, the computing resources available to the remote computer may need to be increased. The goal here is to also protect the valuable data from unauthorized viewing, even if it is stored on a portable computing device. Some VPN client software includes virus protection, distributed desktop firewall, desktop intrusion protection, and file/disk encryption. This type of solution may be more than is required for certain applications, but it does illustrate the principle of defense in depth, even at a desktop level. Add to this mix strong authentication and digital signing and the security risk decreases, assuming the application of a well-thought-out policy along with proper implementation of the policy. The aforementioned applies to dialup users as well; any time one connects via dialup, one receives a publicly reachable and hence attackable IP address.

VPN client integrity issues must also be considered. For example, does the VPN client have the ability to authenticate a security policy update or configuration update from the VPN server? Does the user have to cooperate in some way for the update to be successfully completed? Users can be a weak link in the chain if they have to be involved in the VPN client update process. Consider VPN clients that allow secured auto-updates of VPN client configuration without user participation. Antivirus protection is a must due to the potential of a Trojan horse or virus, for example, to perform unauthorized manipulation of VPN system. Is the VPN client compatible with (or does it include) desktop antivirus programs? We are witnessing an increase in targeted attacks, that is, where the attacker select targets for a particular reason rather than blindly probing for a vulnerable host. These kinds of attacks include the ability of attackers to coordinate and attack through VPN entry points. This is plausible for a determined attacker who systematically subverts remote VPN user connections into the central site. Therefore, one may have a requirement to protect the VPN client from subversion through the use of distributed desktop firewalls and desktop intrusion-detection systems.

The key differentiator of a distributed desktop firewall is that firewall policy for all desktops within an organization are managed from a central console. Personal firewalls, as the name implies, are marketed to individual consumers. The individual user is responsible for policy maintenance on personal firewalls. A distributed firewall is marketed to businesses that need to centrally enforce a consistent network security policy at all entry points to the internal network, including the remote VPN user connection. By deploying an IPSec VPN client in conjunction with a distributed firewall and an intrusion-detection system that reports back to a central management console, ongoing network attacks can be coalesced and correlated to provide an enterprise view of the security posture. Ideally, an IPSec vendor could provide a VPN client that includes anti-virus, desktop intrusion detection, and a distributed firewall along with the IPSec VPN client. A product that provides that level of integration would certainly enhance the efficiency of desktop security policy management.

Deploying the Client

Remote access VPN client software deployment issues are primarily operational issues that occur with any distributed software, such SQL client software. There is a wide body of software administration knowledge and methodologies that can be adapted to deploying remote access VPN client software.

Several issues must be sorted out when examining the deployability of a VPN client. One such issue is the VPN client software file size. This becomes an important issue if the selected mode of client software distribution is via low-speed dial-up, currently the most widely used remote access method. If the file takes too long to download, say, from a distribution FTP server, it is possible that affected users will be resistant to downloading the file or future updates. Resistant users may increase the likelihood of protracted implementation of the VPN, thus increasing total implementation cost. However, promise of pervasive high-speed access is on

the horizon. A deployment strategy that could resolve this issue is to distribute the VPN client initially by portable media, such as diskette or CD-ROM. Data compression can also help shrink VPN client distribution size. Most vendors supply some sort of client configuration utility that allows an administrator to preconfigure some initial settings, then distribute the installation file to each remote user. Possible VPN client distribution methods include posting to a Web or FTP site. If using Web, FTP, or other online file transfer method, it is important that the security professional anticipate possible scenarios that include the case of unauthorized access to the VPN client installation file. Some companies may decide that they will only distribute the installation files in person. Others are prepared to accept the risk of distribution via postal or electronic mail. Others may elect to set up a secured file transfer site, granting access via a PIN or special passphrase. When it comes to the initial distribution of the VPN client, the possibilities are limited only by the level of risk that is acceptable based on the value of loss if breached. This is especially the case if the initial VPN client software contains is preconfigured with information that could be used as reconnaissance information by an attacker.

Client Management Issues

VPN client management pertains to operational maintenance of the client configuration, VPN client policy update process, and upgrading of the VPN client software. Again, there are many approaches that can be adapted from the general body of knowledge and software management methodologies used to manage other types of software deployed by enterprises today. The additional factors are user authentication of updates, VPN availability, update file integrity, and confidentiality. The ability to manage user credentials is discussed in the chapter section on VPN server management issues.

Because the VPN client represents another access point into the internal network, such access requires rigorous user authentication and stringently controlled VPN configuration information. Many would argue that the highest practical level of strong authentication is biometrics based. If a PIN is used in conjunction with biometrics, it can be considered two-factor authentication. The next choice by many security professionals is the digital certificate stored on a smart card with PIN combination. The use of time-based calculator cards (tokens) and simple passwords is falling into legacy usage. However, many IPSec vendors are implementing the XAUTH extension to the IKE/IPSec standard. The XAUTH extension allows the use of legacy user authentication methods such as RADIUS, currently the most widely used authentication method in use, when validating user identity during IPSec tunnel setup. An added benefit is that XAUTH allows a company to leverage existing legacy authentication infrastructure, thus extending the investment in the older technology. The result: less changes to the network and a potential for decreased implementation time and costs due to reuse of existing user accounts. Another result of XAUTH use is relatively weaker authentication, given the increased vulnerability of passwords and token use.

A question that bears consideration due to the possibility of spoofing the VPN update server is "How does the client software confirm the sender of its receipt of the configuration update file?" With many forms of configuration distribution, an opportunity exists for an attacker to send an unauthorized update file to users. One control against this threat is the use of cryptography and digital signatures to digitally sign the update file, which can then be verified by the VPN client before acceptance. An additional protection would be to encrypt the actual configuration file as it resides on the remote user computer. One common method is to use a secured path to transfer updates, for example, LDAP over SSL (LDAPs).

Exhibit 40.2 shows a sample evaluation profile for remote access VPN client software. This is a list of items that may be considered when developing evaluation criteria for a VPN client.

The Remote Access Server

The major processing of encryption tunnel traffic is done at the remote access (VPN) server. The VPN server becomes a point of tunnel aggregation: the remote access client uses the server as a tunnel endpoint. There are basically two ways to verify that the VPN server has the capacity to efficiently process the VPN traffic. The first is to use bigger, faster hardware devices to overcome processing limitations; solutions based on monolithic hardware are tied directly to performance advances in hardware. If performance enhancements are slow to arrive, so will the ability to scale upward. This approach is commonly referred to as vertical scalability. The second alternative is load balancing, or distributing, the VPN connections across a VPN server farm. Load balancing requires special processors and software, either through dedicated load balancing hardware or via policy and state replication among multiple VPN servers. In terms of connections and economies, a load balanced VPN server farm will always offer better scalability because more servers can be added as needed. Load balancing will also offer redundancy; if any VPN server fails, the load will be distributed among the remaining VPN servers. (Some HA solutions can do this without disrupting sessions; other are more disrup-

EXHIBIT 40.2 Evaluation Criteria for Remote Access VPN Client

Assumption: VPN client is subject to the management of the central site
File/disk encryption may be needed for security of mobile user desktop
High-performance laptops/notebooks may be needed if using extensive disk/file encryption
Desktop intrusion detection with alerting integrated into centralized VPN manager
Distributed desktop firewall with alerting integrated into centralized VPN manager
Ability to lock down VPN client configuration
Transparent-to-user VPN client update
Authenticated VPN client update over an encrypted link
Adherence to current industry VPN standards if interoperability is a requirement

tive.) Encryption accelerators — hardware-based encryption cards — can be added to a VPN server to increase the speed of tunnel processing at the server. Encryption acceleration is now being implemented at the chip level of network interface cards as well. Encryption acceleration is more important for the VPN server than on the individual VPN client computer, again due to the aggregation of tunnels.

When evaluating the VPN server's capability, consider ease of management. Specifically, how easy is it for an administrator to perform and automate operational tasks? For example, how easy is it to add new tunnels? Can additional tunnel configurations be automatically "pushed" or "pulled" down to the VPN client? Logging, reporting, and alerting is an essential capability that should be integrated into the VPN server management interface. Can the VPN logs be exported to existing databases and network management systems? Does the VPN server provide real-time logging and alerting? Can filters be immediately applied to the server logs to visually highlight user-selectable events? If using digital certificates, what certificate authorities are supported? Is the certificate request and acquisition process an automated online procedure? Or does it require manual intervention? Repetitive tasks such as certificate request and acquisition are natural candidates for automation. Does the VPN server automatically request certificate revocation lists to check the validity of user certificates?

Exhibit 40.3 shows a sample evaluation profile for remote access VPN servers.

Intranet VPN

An intranet VPN connects fixed locations and branch and home offices within an enterprise WAN. An intranet VPN uses a site-to-site, or VPN gateway-to-VPN gateway, topology. The business benefits of an intranet VPN include reduced network infrastructure costs and increased information flow within an organization. Because the nature of an intranet is site to site, there is little impact on end-user desktops. The key criteria in evaluating VPN solutions for an intranet application are performance, interoperability with preexisting network infrastructure, and manageability. The technical benefits of an intranet VPN include reduced WAN bandwidth costs, more flexible topologies (e.g., fully meshed), and quick and easy connection of new sites.

EXHIBIT 40.3 Evaluation Profile for Remote Access VPN Server

Scalability (can the server meet connectivity requirements?)
Supports high-availability options
Integrates with preexisting user authentication systems
Hardware-based tunnel processing, encryption/decryption acceleration
Automated management of user authentication process
Supports industry VPN standards for interoperability
What authentication types are supported?
Does the VPN server run on a hardened operating system?
Is firewall integration on both the VPN client and server side possible?
Centralized client management features
Broad client support for desktop operating systems

The use of remotely configurable VPN appliances, a vendor-provided VPN hardware/software system, is indicated when there will be a lack of on-site administration and quick implementation timeframe. The value of VPN appliances becomes clear when comparing the time and effort needed to integrate hardware, operating system, and VPN server software using the more traditional "build-it-yourself" approach.

Class-of-service controls can be useful when performing traffic engineering to prioritize certain protocols over others. This becomes an issue, for example, when business requirements mandate that certain types of VPN traffic must have less latency than others. For example, streaming video or voice traffic requires a more continuous bit rate than a file transfer or HTTP traffic due to the expectations of the end user or the characteristics of the type of application.

Two limiting factors for general use of intranet VPNs that tunnel through the Internet are latency and lack of guaranteed bandwidth. Although these factors can also affect internationally deployed private WAN-based intranet VPNs, most companies cannot afford enough international private WAN bandwidth to compete against the low cost of VPN across the Internet. Performing a cost/benefit analysis may help in deciding whether to use a private WAN, an Internet-based intranet VPN, or an outsourced VPN service. Multi-Protocol Label Switching (MPLS) is a protocol that provides a standard way to prioritize data traffic. MPLS could be used to mitigate latency and guaranteed bandwidth issues. With MPLS, traffic can be segregated and prioritized so as to allow certain data to traverse across faster links than other data traffic. The benefit of using MPLS-enabled network components in IPSec VPN applications is that VPN traffic could be given priority over other data traffic, thereby increasing throughput and decreasing latency.

The topology of the VPN is an important consideration in the case of the intranet VPN. Many intranet VPNs require a mesh topology, due to the decentralized nature of an organization's information flow. In other cases, a hub-and-spoke topology may be indicated, in the case of centralized information flow, or in the case of a "central office" concept that needs to be implemented. If it is anticipated that network changes will be frequent, VPN solutions that support dynamic routing and dynamic VPN configuration are indicated. Dynamic routing is useful in the case where network addressing updates need to be propagated across the VPN quickly, with little to no human intervention required. Routing services ensure cost-effective migration to VPN infrastructures that provide robust bandwidth management without impacting existing network configurations. Dynamic VPN technology is useful where it is anticipated that spontaneous, short-lived VPN connectivity is a requirement. There is much ongoing research in the area of dynamic VPNs that promise to ease the administrative burden of setting up VPN tunnels in large-scale deployments.

Building an intranet VPN using the Internet is, in general, the most cost-effective means of implementing VPN technology. Service levels, however, as mentioned before, are generally not guaranteed on the Internet. While the lack of service level guarantees is true for general IP traffic, it is not universally so for intranet VPNs. While some ISPs and private-label IP providers (e.g., Digital Island) offer service level guarantees, this technology is only now maturing; and to get the most benefit from such service offerings, customers will typically build their intranets on top of a single ISP's IP network. When implementing an intranet VPN, businesses need to assess which trade-off they are willing to make between guaranteed service levels, pervasiveness of network access, and transport cost. Enterprises requiring guaranteed throughput levels should consider deploying their VPNs over a network service provider's private end-to-end IP network, or, potentially, Frame Relay, or build one's own private backbone.

Exhibit 40.4 provides a list of items that can be used when developing a set of evaluation criteria for an intranet VPN.

EXHIBIT 40.4 Evaluation Profile for a Site-to-Site Intranet VPN

Assumption: none
Support for automatic policy distribution and configuration
Mesh topology automatic configuration, support for hub-and-spoke topology
Network and service monitoring capability
Adherence to VPN standards if used in heterogeneous network
Class-of-service controls
Dynamic routing and tunnel setup capability
Scalability and high availability

Extranet VPN

Extranet VPNs allows for selective flow of information between business partners and customers, with an emphasis on highly granular access control and strong authentication. For example, security administrators may grant user-specific access privileges to individual applications using multiple parameters, including source and destination addresses, authenticated user ID, user group, type of authentication, type of application (e.g., FTP, Telnet), type of encryption, the day/time window, and even by domain.

An extranet VPN might use a user-to-central site model, in which a single company shares information with supply-chain and business partners, or a site-to-site model, such as the Automotive Network Exchange. If using the user-to-site model, then evaluation criteria are similar to remote access VPNs with the exception that the user desktop will not be under the control of the central site. Because the extranet user's computer is under the control of its own company security policy, there may be a conflict in security policy, implemented on the users' computer. In general, extranet partners in the user-to-site model will need to work together to reach an agreement as to security policy implementation at the user desktop, VPN client installation issues, help desk, ongoing maintenance if one partner is mandating the use of a particular VPN client, and liability issues should one partner's negligence lead to the compromise of the other partner's network. The hardware platforms supported by a vendor's VPN client will also be an issue that will require a survey of possible platforms that remote extranet partners will be using. For the most part, Web-based access is often used as the software client of choice in extranet environments, and SSL is often chosen as the security protocol. This greatly simplifies the configuration and maintenance issues that will need to be confronted. With an extranet VPN, it really does not matter whether all the participants use the same ISP, assuming acceptable quality of service is provided by whichever ISP is chosen. All that is required is for each member of the group to have some type of access to the Internet. The VPN software or equipment in each site must be configured with the IP address of the VPN equipment in the main site of the extranet.

Because the appeal of an extranet VPN is largely one of the ability to expand markets and increased strength of business relationships, from a marketing perspective it may be desirable to brand the extranet client software. This can be done, with some extranet VPN software and service providers, either at the Web page that is the extranet entry point (if using a Web browser as the software platform) or within the VPN client (if using the traditional client/server software model). In the consumer market, extranet VPNs can be used as an alternative to Web browser-based SSL. A situation in which IPSec VPNs would be preferable to Web browser-based SSL is when the customer is known and is likely to come back to the site many times. In other words, an extranet VPN would not necessarily work well in a consumer catalog environment where people might come once to make a purchase with a credit card.

A Web browser-based SSL is fine for spontaneous, simple transactional relationships, but an IPSec VPN client/server solution using digital certificates-based mutual authentication may be more appropriate for persistent business relationships that entail access to high-value data. Browser-based SSL could be appropriate for this kind of application if client-side certificates are used. The main idea is that once the user is known by virtue of a digital certificate, the access control features of a VPN can then be used to give this person access to different resources on the company's network. This level of control and knowledge of who the user is has led many companies to use digital certificates. Obviously, this is a concern in large-scale extranet VPN implementations. The issues related to the PKI within the extranet VPN are beyond the scope of this chapter.

Should an existing intranet VPN be used as the basis for implementing an extranet VPN? It depends on the level of risk acceptance and additional costs involved. Enabling an intranet to support extranet connections is a fairly simple undertaking that can be as basic as defining a new class of users with limited rights on a network. There are, however, several nuances to designing an extranet VPN that can directly impact the security of the data. One approach to enabling an extranet, for example, is to set up a demilitarized zone (for example, on a third interface of a perimeter firewall) to support outside users. This solution provides firewall protection for the intranet and the extranet resources, as well as data integrity and confidentiality via the VPN server.

Exhibit 40.5 shows a sample evaluation profile for an extranet VPN application. Below is a list of items that can be used when developing a set of evaluation criteria for an extranet VPN.

Securing the Internal Network

Due to constant insider threat to data confidentiality, companies now realize that internal network compartmentalization through the use of VPNs and firewalls is not just a sales pitch by security vendors trying to sell

EXHIBIT 40.5 Evaluation Profile for an Extranet VPN

Prefer strong mutual authentication over simple username/passwords
Access control and logging are very important
Prefer solutions that allow client customization for branding
Minimal desktop footprint (because the desktop is not under the control of the partner)
Minimal intrusiveness to normal application use
Silent installation of preconfigured VPN client and policy
Ease-of-use of the VPN client is key
Service level monitoring and enforcement support

more products. Although external threat is growing, the internal threat to data security remains constant. Therefore, an emerging VPN application is to secure the internal network.

There are many ways that a network can be partitioned from a network security perspective. One approach is to logically divide the internal network. Another approach is to physically partition the network. VPN technology can be used in both approaches. For example, physical compartmentalization can be accomplished by placing a target server directly behind a VPN server. Here, the only way the target server can be accessed is by satisfying the access control policy of the VPN server. The benefits here include simplicity of management, clearly defined boundaries, and a single point of access. An example of logical compartmentalization would be the case in which users who need access to a target server are given VPN client software. The users can be physically located anywhere on the internal network, locally or remote. The VPN client software automatically establishes an encrypted session with the target server, either directly or through an internal VPN gateway. The internal network is thereby logically "partitioned" via access control. Another logical partitioning scenario would be the case in which peer-to-peer VPN sessions need to be established on the internal network. In this case, two or more VPN clients would establish VPN connectivity, as needed, on an ad hoc basis. The benefit of this configuration is that dynamic VPNs could be set up with little user configuration needed, along with data privacy. The downside of this approach would be decreased user authentication strength if the VPN clients do not support robust user authentication in the peer-to-peer VPN.

There appears to be a shift in placement emphasis regarding where VPN functionality is implemented within the network hierarchy. With the introduction of Microsoft Windows 2000, VPN technology is being built into the actual operating system as opposed to being added later using specialized hardware and software. With this advent, the level of VPN integration that can be used to secure the internal network becomes much deeper, if implemented properly. VPN technology is being implemented at the server level as well in Microsoft Windows and with various versions of UNIX. Although this does not mean that this level of VPN integration is all that is needed to secure the internal network, it does encourage the concept of building in security from the beginning, and using it end-to-end. Implementation of a VPN directly on the target application server, to date, has a considerable impact on performance; thus, hardware acceleration for cryptographic functions is typically required.

The requirement to provide data confidentiality within the internal network can be met using the same deployment and management approaches used in implementing remote access VPN. The user community is generally the same. The hardware platform could be the same, especially with so many companies issuing laptop and other portable computers to their employees. One difference that must be considered is the security policy to be implemented on the VPN client while physically inside the internal network versus the policy needed when using the same hardware platform to remotely access the internal network via remote access VPN. The case might exist where it is prudent to have a tighter security policy when users are remotely logging in due to increased risk of unauthorized access to company data as it traverses a public transport such as the Internet. Although the risks are the same on internal or external access, the opportunity for attack is much greater when using the remote access VPN. There is another application of VPN technology on internal networks, which is to provide data confidentiality for communications across LANs. Due to the operational complexity of managing potentially *n*-squared VPN connections in a Microsoft File Sharing/SMB environment, however, some companies are investigating whether a single "group" or LAN key is sufficient — in such deployments, data confidentiality in transport is more important than authentication.

A sample evaluation profile for securing the internal network VPN application in Exhibit 9-6.

EXHIBIT 40.6 Evaluation Profile for Securing the Internal Network VPN Application

Strong user authentication
Strong access control
Policy-based encryption for confidentiality
In-transit data integrity
Low impact to internal network performance
Low impact on the internal network infrastructure
Low impact to user desktop
Ease of management
Integration with preexisting network components
Operational costs (may not be a big issue when weighed against the business objective)
VPN client issues:
User transparency (does the user have to do anything different?)
Automatic differentiation between remote access and internal VPN policy (can the VPN client auto adapt to internal/external security policy changes?)

VPN Deployment Models

There are four VPN server deployment models discussed in this chapter section: dedicated hardware/appliance, software based, router based, and firewall based. The type of VPN platform used depends on the level of security needed, performance requirements, network infrastructure integration effort, and implementation and operational costs. This discussion now concentrates on VPN server deployment considerations, as VPN client deployment was discussed in earlier chapter sections.

Dedicated Hardware VPN Appliance

An emerging VPN server platform of choice is that of a dedicated hardware appliance, or purpose-built VPN appliance. Dedicated hardware appliance usage has become popular due to fact that its single-purpose, highly optimized design is shown to be (in some respects) easier to deploy, easier to manage, easier to understand, and in many cases cost effective. The idea behind this type of platform is similar to the example of common household appliances. For example, very few people buy a toaster and then attempt to modify it after bringing it home. The concept to grasp here is turnkey.

These units are typically sold in standard hardware configurations that are not meant to be modified by the purchaser. Purpose-built VPN appliances often have the advantage over other platforms when it comes to high performance due to the speed efficiency of performing encryption in hardware. Most purpose-built VPN appliances are integrated into a specialized real-time operating system optimized to efficiently run on specially designed hardware. Many low-end VPN appliances use a modified Linux or BSD operating system running on an Intel platform. Many VPN appliances can be preconfigured, shipped to a remote site, and easily installed and remotely managed. The advantage here is quick implementation in large-scale deployments. This deployment model is used by large enterprises with many remote offices, major telecom carriers, ISPs, and managed security service providers. If an enterprise is short of field IT personnel, VPN appliances can greatly reduce the human resource requirement for implementing a highly distributed VPN.

One approach to rolling out a large-scale, highly distributed VPN using hardware VPN devices is to: (1) preconfigure the basic networking parameters that will be used by the appliance, (2) pre-install the VPN appliance's digital certificate, (3) ship the appliance to its remote location, and (4) then have someone at the remote location perform the rudimentary physical installation of the appliance. After the unit is plugged into the power receptacle and turned on, the network cables can be connected and the unit should then be ready for remote management to complete the configuration tasks as needed. Drawbacks to the use of the VPN appliances approach include the one-size-fits-all design concept of VPN appliance products, which does not always allow for vendor support of modifications of the hardware in a VPN appliance. Additionally, VPN appliances that use proprietary operating systems may mean learning yet another operating system and may

not cleanly interoperate with existing systems management tools. The bottom line is: if planning to modify the hardware of a VPN appliance oneself, then VPN appliances may not be the way to go.

Many carrier-class VPN switches — VPN gateways that are capable of maintaining tens of thousands of separate connections — are another class of VPN component that fits the requirements of large-scale telecommunications networks such as telcos, ISPs, or large enterprise business networks. Features of carrier-class VPN gateways include quick and easy setup and configuration, allowing less-experienced personnel to perform installations. High throughput, which means it can meet the needs of a growing business, and easy-to-deploy client software are also differentiators for carrier-class VPN gateways.

Software-Based VPNs

Software-based VPN servers usually require installation of VPN software onto a general-purpose computer running on a general-purpose operating system. Typical operating systems that are supported tend to be whatever operating system is the market leader at the time. This has included both Microsoft Windows-based and UNIX-based operating systems. Some software-based VPNs will manipulate the operating system during installation to provide security hardening, some level of performance optimization, or fine-tuning of network interface cards. Software-based VPNs may be indicated if the VPN strategy is to upgrade or "tweak" major components of the VPN hardware in some way due to the turnkey concept of the appliance approach. Also, the software VPN approach is indicated if one plans to minimize costs by utilizing existing general-purpose computing hardware.

Disadvantages of software-based VPN servers are typically performance degradation when compared to purpose-built VPN appliances, the server hardware and operating system must be acquired if not available, the additional cost for hardware encryption cards, and the additional effort required to harden the operating system. Applying appropriate scalability techniques such as load balancing and using hardware encryption add-on cards can mitigate these disadvantages. Also, the VPN software-only approach has a generally less-expensive upfront purchase price. Sometimes, the software is built into the operating system; for example, Microsoft Windows 2000 Server includes an IPSec VPN server.

Some vendors' software VPN products are supported on multiple platforms that cannot be managed using a central management console or have a different look and feel on each platform. To ensure consistent implementation and manageability, it makes sense to standardize on hardware platforms and operating systems. By standardizing on platforms, the learning curve can be minimized and platform-based idiosyncrasies can be eliminated.

Router-Based VPNs

One low-cost entry point into deploying a VPN is to use existing routers that have VPN functionality. By leveraging existing network resources, implementation costs can be lowered, and integration into network management infrastructure can more easily be accomplished. Many routers today support VPN protocols and newer routers have been enhanced to more efficiently process VPN traffic. However, a router's primary function is to direct network packets from one network to another network; therefore, a trade-off decision may have to be made between routing performance and VPN functionality. Some router models support hardware upgrades to add additional VPN processing capability. The ability to upgrade existing routers provides a migration path as the VPN user community grows. Many router-based VPNs include support for digital certificates. In some cases, the digital certificates must be manually requested and acquired through the use of cutting and pasting of text files. Depending on the number of VPN nodes, this may affect scalability. VPN-enabled routers require strong security management tools — the same kinds of tools normally supplied with hardware appliance and software VPNs.

Where should the router-based VPN tunnel terminate? The tunnel can be terminated in either of two places: outside the network perimeter when adding VPN to an access router, or terminating tunneled traffic behind the firewall when adding VPN to an interior router.

Firewall-Based VPNs

Firewalls are designed to make permit/deny decisions on traffic entering a network. Many companies are have already implemented firewalls at the perimeter of their networks. Many firewalls have the ability to be upgraded

for use as VPN endpoints. If this is the case, for some organizations it may make sense to investigate the VPN capability of their existing firewall. This is another example of leveraging existing network infrastructure to reduce upfront costs. A concern with using firewalls as a VPN endpoint would be performance. Because all traffic entering or leaving a network goes through a firewall, the firewall may already be overloaded. Some firewall vendors, however, offer hardware encryption add-ons. As with any configurable security device, any changes made to a firewall can compromise its security. VPN management is enhanced through use of a common management interface provided by the firewall. As the perimeter firewall, this is an ideal location for the VPN because it isolates the ingress/egress to a single point. Adding the VPN server to the firewall eliminates the placement issues associated with hardware, software, and router VPNs; for example, should encrypted packets be poked through a hole in the firewall, what happens if the firewall performs NAT, etc.?

The firewall/VPN approach also allows for termination of VPN tunnels at the firewall, decryption, and inspection of the data. A scenario in which this capability is advantageous is when firewall-based anti-virus software needs to be run against data traversing the VPN tunnel.

General Management Issues for Any VPN

The question arises as to who should manage the software-based VPN. Management can be divided between a network operations group, a security group, and the data owner. The network operations will need to be included in making implementation and design decisions, as this group is usually charged with maintaining the availability of a company's data and data integrity. The security group would need to analyze the overall system design and capability to ensure conformance to security policy. The data owner, in this case, refers to the operational group that is using the VPN to limit access. The data owner could be in charge of access control and user account setup. In an ideal situation, this division of labor would provide a distributed management approach to VPN operations. In practice, there is rarely the level of cooperation required for this approach to be practical.

Evaluating VPN Performance

To this point, we have discussed the criteria for evaluating VPNs from end-user and administrator perspectives. However, it is also insightful to understand how VPN vendors establish benchmarks for performance as a marketing tool. Many vendors offer VPN products that they classify by the number of concurrent VPN connections, by the maximum number of sessions, or by throughput. Most security professionals are interested in how secure the implementation is; most network operations staff, especially ISP staff, are interested in how many clients or remote user tunnels are supported by a VPN gateway. An IPSec remote user tunnel can be defined as the completion of IKE phase 1 and phase 2 key exchanges. These phases must be completed to create a secure tunnel for each remote communications session, resulting in four security associations. This is a subjective definition because vendors typically establish various definitions to put their performance claims in the best possible light.

Although many vendors provide a single number to characterize VPN throughput, in real-world deployments, performance will vary depending on many conditions. This chapter section provides a summary of the factors that affect throughput in real-world deployments.

Packet Size

Most VPN operations, such as data encryption and authentication, are performed on a per-packet basis. CPU overhead is largely independent of packet size. Therefore, larger packet sizes typically result in higher data throughput figures. The average size of IP packets on the Internet is roughly 300 bytes. Unfortunately, most vendors state VPN throughput specifications based on relatively large average packet sizes of 1000 bytes or more. Consequently, organizations should ask vendors for throughput specifications over a range of average packet sizes to better gauge expected performance.

Encryption and Authentication Algorithms

Stronger encryption algorithms require greater system resources to complete mathematical operations, resulting in lower data throughput. For example, VPN throughput based on DES (56-bit strength) encryption may

be greater than that based on 3DES (168-bit strength) encryption. Stream ciphers are typically faster than block ciphers.

Data authentication algorithms can have a similar effect on data throughput. For example, using MD5 authentication may result in a slightly greater throughput when compared with SHA1.

Host CPU

Software-based VPN solutions provide customers with a choice of central processors, varying in class and clock speed. Host processing power is especially critical with VPN products not offering optional hardware-based acceleration. VPN testing has shown that performance does not linearly increase by adding additional general-purpose CPUs to VPN servers. One vendor claims that on a Windows NT server, if one processor is 100 percent loaded, adding a second processor frees CPU resources by only 5 percent. The vendor claims a sevenfold increase in throughput when using encryption acceleration hardware instead of adding general-purpose CPUs to the server. In other cases, the price/performance of adding general-purpose CPUs compared to adding hardware acceleration weighs against the former. In one case, the cost of adding the general-purpose CPU was approximately twice the price of a hardware acceleration card, with substantially less performance increase. Speed is not just a factor of CPU, but also a factor of I/O bus, RAM, and cache. Reduced Instruction Set CPUs, RISC processors, are faster than general-purpose CPUs, and Application-Specific Integrated Circuits, ASICs, are typically faster at what they are designed to do than RISC processors.

Operating System and Patch Levels

Many software-based VPN solutions provide customers with a choice of commercial operating systems. Although apples-to-apples comparisons of operating systems are difficult, customers should make sure that performance benchmarks are specific to their target operating system. Also, operating system patch levels can have a significant throughput impact. Usually, the most current operating system patch levels deliver better performance. If the VPN requirement is to use operating system-based VPN technology, consider software products that perform necessary "hardening" of operating systems, as most software firewalls do. Consider subscribing to ongoing service plans that offer software updates, security alerts, and patch updates.

Network Interface Card Drivers

Network interface card (NIC) version levels can affect throughput. Usually, the most updated network interface card drivers deliver the best performance. A number of network interface card manufacturers now offer products that perform complementary functions to IPSec-based VPNs. NICs can be installed in user computers or IPSec VPN gateways that perform encryption/decryption, thereby increasing system performance while decreasing CPU utilization. This is achieved by installing a processor directly on the NIC, which allows the NIC to share a greater load of network traffic processing so the host system can focus on servicing applications.

Memory

The ability of a VPN to scale on a remote user tunnel basis depends on the amount of system memory installed in the gateway server. Unlike many VPN appliance solutions (which are limited by a fixed amount of memory), a software-based VPN is limited in its support of concurrent connections and remote user tunnels by maximum number of concurrent connections established by the kernel. In some cases, concurrent connections are limited by VPN application proxy connection limits, which are independent of the host's kernel limits. However, it is important to understand that most VPN deployments are likely to run into throughput limitations before reaching connection limitations. Only by combining the memory extensibility of software-based VPN platforms and throughput benefits of dedicated hardware can the best of both worlds be achieved. Consider the following hypothetical example. An organization has a 30-Mbps Internet link connected to a software-based VPN with a hardware accelerator installed. For this organization, the required average data rate for a single remote user is approximately 40K. In this scenario, the VPN will support approximately 750 concurrent remote users (30 Mb/40K.) Once the number of users increases beyond 750 users, average data rates and the corresponding user experience will begin to decline. It is clear from this example that reliable, concurrent user support is more likely to be limited by software-based VPN gateway throughput than by limitations in the number of

connections established. From this perspective, the encryption accelerator card is a key enabler in scaling a software-based VPN deployment to support thousands of users.

A single number does not effectively characterize the throughput performance of a VPN. The size of packets being transferred by the system, for example, has a major impact on throughput. System performance degrades with smaller packet sizes. The smaller the packet size, the greater the number of packets processed per second, the higher the overhead, and thus the lower the effective throughput. An encryption accelerator card can be tuned for both large and small packets to ensure performance is optimized for all packet sizes. Other factors that can affect performance include system configuration (CPU, memory, cache, etc.), encryption algorithms, authentication algorithms, operating system, and traffic types. Most of these factors apply to all VPN products. Therefore, do not assume that performance specifications of competitive VPN products mean that those numbers can be directly compared or achieved in all environments.

Data Compression Helps

To boost performance and improve satisfaction among end users, a goal to reach for is to minimize delay across the VPN. One way to minimize delay is to send less traffic. This goal can be achieved by compressing data before it is put on the VPN. Performance gains from compression vary, depending on what kind of data is being sent; but, in general, once data is encrypted, it just does not compress as well as it would have unencrypted. Data compression is an important performance enhancer, especially when optimizing low-bandwidth analog dialup VPN access, where MTU size and fragmentation can be factors.

Is Raw Performance the Most Important Criteria?

According to a recent VPN user survey whose goal was to discover which features users think are most important in evaluating VPNs and what they want to see in future offerings, performance was rated higher than security as a priority when evaluating VPNs. This marks a shift in thinking from the early days of VPN when few were convinced of the security of the underlying technology of VPN.

This is particularly true among security professionals who rate their familiarity with VPN technologies and products as "high." Those who understand VPNs are gaining confidence in security products and realize that performance and management are the next big battles. According to the survey results, many users feel that while the underlying security components are a concern, VPN performance must not be sacrificed. According to the survey, users are much more concerned about high-level attributes, such as performance, security of implementation, and usability, and less concerned about the underlying technologies and protocols of the VPN.

Outsourcing the VPN

Outsourcing to a knowledgeable service provider can offer a sense of security that comes from having an expert available for troubleshooting. Outsourcing saves in-house security managers from the problems associated with physically upgrading their VPNs every time branch offices are setting and testing remote users who need to be added to the network. Unless a company happens to have its own geographically dispersed backbone, then at least the transit portion of the VPN will have to be outsourced to an Internet access service provider or private IP network provider. However, a generic Internet access account does not provide much assurance that the VPN traffic will not get bogged down with the rest of the traffic on the Internet during peak hours. The ISP or VPN service provider can select and install the necessary hardware and software, as well as assume the duties of technical support and ongoing maintenance.

Exhibit 40.7 lists some factors to consider when evaluating a VPN service provider.

Reliability

If users are not able to get on to the network and have an efficient connection, then security is irrelevant. If the goal of the VPN is to provide remote access for mobile workers, then a key aspect of performance is going to be the number of points of presence the service provider has in the geographic regions that will require service, as well as guarantees the service provider can make in terms of its success rates for dialup access. For example, one VPN service provider (provides transport and security services) offers 97 percent busy-free dialing

EXHIBIT 40.7 Evaluating a VPN service provider

Factors to consider when evaluating a VPN service provider include:
Quality of service
Reliability
Security
Manageability
Securities of the provider's own networks and network operations centers
Investigate the hiring practices of the provider (expertise, background checks)
What pre- and post-deployment services does the provider offer (vulnerability assessment, forensics)

for remote access, with initial modem connect speeds of 26.4 KBps or higher, 99 percent of the time. Another VPN service provider (provides transport and security services) promotes 100 percent network availability and a 95 percent connection success rate for dialup service. When such guarantees are not met, the service provider typically promises some sort of financial compensation or service credit. VPN transport and security services can be outsourced independently.

However, if the main goal is to provide a wide area network for a company, overall network availability and speed should be a primary concern. Providers currently measure this by guaranteeing a certain level of performance, such as throughput, latency, and availability, based on overall network averages. Providers that build their own backbones use them to support many customer VPNs. Some VPN service providers provide private WAN service via Asynchronous Transfer Mode or Frame Relay transport for customer VPNs. This way, VPN traffic does not have to compete for bandwidth with general Internet traffic, and the VPN service provider can do a better job of managing the network's end-to-end performance.

Quality of Service

VPN service providers are beginning to offer guarantees for performance-sensitive traffic such as voice data and multimedia. For example, a network might give higher priority to a streaming video transmission than to a file download because the video requires speedy transmission for it to be usable. The current challenge is to be able to offer this guarantee across network boundaries. While it is currently possible with traffic traveling over a single network, it is almost impossible to do for traffic that must traverse several networks. This is because, although standards like MPLS are evolving, there is no current single standard for prioritizing traffic over a network, much less the Internet.

To ensure better performance, many VPN service providers offer service level agreements. For an extra charge, commensurate with the quality of service, a VPN service provider can offer its customers guarantees on throughput, dial-in access, and network availability. Some VPN service providers have their own private Frame Relay or Asynchronous Ttransfer mode networks over which much of the VPN traffic is routed, enhancing performance.

Security

A VPN provides security through a combination of encryption, tunneling, and authentication/authorization. A firewall provides a perimeter security defense by allowing only trusted, authorized packets or users access to the corporate network. Companies can opt to have their VPN service provider choose the security method for their VPN and can either manage it in-house or allow the service provider to manage this function. Another option is for the customer to handle the security policy definition of the VPN entirely. Most security managers prefer to retain some control over their network's security, mainly in the areas of end-user administration, policy, and authentication. A company might opt to do its own encryption, for example, or administer its own security server, but use the VPN service provider for other aspects of VPN management, such as monitoring and responding to alerts. The decision of whether or not to outsource security, for some, has to do with the size and IT resources of a company. For others, outsource decisions have more to do with the critical nature of the corporate data and the confidence the IT manager has in outsourcing in general.

Exhibit 40.7 enumerates factors to be considered when evaluating outsourced VPNs.

Manageability

Another issue to consider is the sort of management and reporting capabilities that are needed from the VPN service provider. Many VPN service providers offer subscribers some sort of Web-based access to network performance data and customer usage reports. Web-based tools allow users to perform tasks such as conducting updates of remote configurations, adding/deleting users, controlling the issuance of digital certificates, and monitoring performance-level data. Check if the VPN service provider offers products that allow split administration so that customers can add and delete users and submit policy changes at a high level.

Summary

Establishing a VPN evaluation strategy will allow security professionals to sort out vendor hype from actual features that meet a company's own VPN system requirements. The key is to develop a strategy and set of criteria that match the VPN application type that is needed. The evaluation criteria should define exactly what is needed. A hands-on lab evaluation will help the security professional understand exactly what will be delivered. Pay particular attention to the details of the VPN setup and be vigilant with any VPN service provider or product vendor that is selected.

Similarly, a well-thought-out VPN deployment strategy will help keep implementation costs down, increase user acceptance, and accelerate the return on investment. The deployment strategy will vary, depending on the type of VPN application and deployment model chosen.

Vendors traditionally want to streamline the sales cycle by presenting as few decision points as possible to customers. One way this is done is to oversimplify VPN product performance characteristics. Do you want a size small, medium, or large? Do you want a 10-user VPN server, 100-user VPN server, or mega-user VPN server? Do you want the 100-MHz or 1-Gigabit model? Insist that VPN vendors provide the parameters used to validate their claims. It is important that security professionals understand the metrics and validation methodologies used by vendors. Armed with this knowledge, security professionals can make informed decisions when selecting products.

There are many options available for implementing VPNs. Managed security service providers can ease some of the burden and help implement VPNs quickly. However, security professionals will do well to exercise due diligence when selecting a service provider.

Glossary

ATM (Asynchronous Transfer Mode) A means of digital communications that is capable of very high speeds; suitable for transmission of images or voice or video as well as data. Commonly deployed in backbone networks.

DSL (Digital Subscriber Line) A generic name for a family of high-speed digital lines being provided by competitive local exchange carriers and local phone companies to provide broadband access to their subscribers.

FTP (File Transfer Protocol) A protocol that allows users to copy files between their local system and any system they can reach on a network. Consists of FTP client and FTP server.

IKE (Internet Key Exchange) A protocol used in IPSec VPNs to establish security parameters for use during an IPSec VPN session (referred to as a security association).

IPSec (IP Security protocol) A standard suite of protocols used in VPNs which defines encryption and data integrity algorithms and rules determining the format and transmission of secure IP packets.

Kbps Kilobits per second.

Mbps Megabits per second.

MSP (Managed Security Service Provider) A class of network infrastructure provider that offers to assume various network security tasks on behalf of its customers. VPN service providers provide VPN server/client deployment assistance and operational management of VPNs.

SSL (Secure Socket Layer) A security protocol that was originally developed by Netscape. SSL has been universally accepted on the World Wide Web for authenticated and encrypted communication between clients and servers. SSL is usually associated with browsers, although it can be used to secure other TCP/IP protocols, such as FTP. SSL has evolved into TLS.

TLS (Transport Layer Security protocol) An IETF draft standard protocol that provides communications privacy over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery.

VPN client Software that resides on individual users computer that establishes a VPN tunnel to a VPN server.

VPN server A device (IPSec security gateway) that resides at a central location and terminates a VPN tunnel. Communicates with VPN clients and other VPN servers. Can be hardware or software based.

VPN (Virtual Private Network) A network that provides the ability to transmit data that ensures confidentiality, authentication, and data integrity.

41

HOW TO PERFORM A SECURITY REVIEW OF A CHECKPOINT FIREWALL

Ben Rothke, CISSP

Altered States was not just a science fiction movie about a research scientist who experimented with altered states of human consciousness; it is also a metaphor for many firewalls in corporate enterprises.

In general, when a firewall is initially installed, it is tightly coupled to an organization's security requirements. After use in a corporate environment, the firewall rule base, configuration, and underlying operating system often gets transformed into a radically different arrangement. This altered firewall state is what necessitates a firewall review.

A firewall is only effective to the degree that it is properly configured. And in today's corporate environments, it is easy for a firewall to become misconfigured. By reviewing the firewall setup, management can ensure that its firewall is enforcing what it expects, and in a secure manner.

This chapter focuses on performing a firewall review for a Checkpoint Firewall-1.[1] Most of the information is sufficiently generic to be germane to any firewall, including Cisco PIX, NAI Gauntlet, Axent Raptor, etc. One caveat: it is important to note that a firewall review is not a penetration test. The function of a firewall review is not to find exploits and gain access into the firewall; rather, it is to identify risks that are inadvertently opened by the firewall.

Finally, it must be understood that a firewall review is also not a certification or guarantee that the firewall operating system or underlying network operating system is completely secure.

The Need for a Firewall Review

Firewalls, like people, need to be reviewed. In the workplace, this is called a performance review. In the medical arena, it is called a physical. The need for periodic firewall reviews is crucial, as a misconfigured firewall is often worse than no firewall. When organizations lack a firewall, they understand the risks involved and are cognizant of the fact that they lack a fundamental security mechanism. However, a misconfigured firewall gives an organization a false sense of security.

In addition, because the firewall is often the primary information security mechanism deployed, any mistake or misconfiguration on the firewall trickles into the entire enterprise. If a firewall is never reviewed, any of these mistakes will be left unchecked.

Review, Audit, Assessment

Firewall reviews are often called audits. An audit is defined as "a methodical examination and review." As well, the terms "review," "assessment," and "audit" are often synonymous. It is interesting to note that when security groups from the Big Five[2] accounting firms perform a security review, they are specifically prohibited from using the term "audit." This is due to the fact that the American Institute of Certified Public Accounts (www.aicpa.org), which oversees the Big Five, prohibits the use of the term "audit" because there is no set of official information security standards in which to audit the designated environment.

On the other hand, financial audits are performed against the Generally Accepted Accounting Principles (GAAP). While not a fixed set of rules, GAAP is a widely accepted set of conventions, standards, and procedures for reporting financial information. The Financial Accounting Standards Board (www.fasb.org) established GAAP in 1973. The mission of the Financial Accounting Standards Board is to establish and improve standards of financial accounting and reporting for the guidance and education of the public, including issuers, auditors, and users of financial information.

As of January 2001, the Generally Accepted System Security Principles (GASSP) Committee was in the early stages of drafting a business plan that reflects their plans for establishing and funding the International Information Security Foundation (IISF).[3] While there is currently no set of geerally accepted security principles (in which a firewall could truly be *audited* against), work is underway to create such a standard. Working groups for the GASSP are in place. Work is currently being done to research and complete the Authoritative Foundation and develop and approve the framework for GASSP. The committee has developed a detailed plan for completing the GASSP Detailed Principles and plans to implement that plan upon securing IISF funding.

The lack of a GASSP means that there is no authoritative reference on which to maintain a protected infrastructure. If there were a GAASP, there would be a way to enforce a level of compliance and provide a vehicle for the authoritative approval of reasonably founded exceptions or departures from GASSP.

Similar in theory to GASSP is the Common Criteria Project (http://csrc. nist.gov/cc). The Common Criteria is an international effort that is being developed as a way to evaluate the security properties of information technology (IT) products and systems. By establishing such a common criteria base, the results of an IT security evaluation will be meaningful to a wider audience.

The Common Criteria will permit comparability between the results of independent security evaluations. It facilitates this by providing a common set of requirements for the security functions of IT products and systems and for assurance measures applied to them during a security evaluation. The evaluation process establishes a level of confidence that the security functions of such products and systems, and the assurance measures applied to them, meet these requirements. The evaluation results help determine whether the information technology product or system is secure enough for its intended application and whether the security risks implicit in its use are tolerable.

Steps in Reviewing a Firewall

A comprehensive review of the firewall architecture, security plans, and processes should include:

- Procedures governing infrastructure access for employees and business partners accessing the infrastructure
- Physical and logical architecture of the infrastructure
- Hardware and software versions of the infrastructure and underlying network operating systems
- Infrastructure controls over access control information
- Review of log event selection and notification criteria
- All access paths, including those provided for maintenance and administration

- Security policies and administrative procedures (i.e., addition or deletion of users and services, review of device and system audit logs, system backup and retention of media, etc.)
- Access controls over the network operating system, including user accounts, file system permissions, attributes of executable files, privileged programs, and network software
- Emergency Response Plans for the infrastructure in the event of an intrusion, denia-of-service attack, etc.
- Access to and utilization of published security alert bulletins

There are many methodologies with which to perform a firewall review. Most center around the following six steps:

1. Analyze the infrastructure and architecture.
2. Review corporate firewall policy.
3. Run hosts and network assessment scans.
4. Review Firewall-1 configuration.
5. Review Firewall-1 Rule Base.
6. Put it all together in a report.

The following discussion expands on each step.

Step 1: Analyze the Infrastructure and Architecture

An understanding of the network infrastructure is necessary to ensure that the firewall is adequately protecting the network. Items to review include:

- Internet access requirements
- Understanding the business justifications for Internet/extranet access
- Validating inbound and outbound services that are allowed
- Reviewing firewall design (i.e., dual-homed, multi-homed, proxy)
- Analyzing connectivity to internal/external networks:
  - Perimeter network and external connections
  - Electronic commerce gateways
  - Inter- or intra-company LAN-WAN connectivity
  - Overall corporate security architecture
  - The entire computing installation at a given site or location
- Interviewing network and firewall administrators

If there is a fault in the information security architecture that does not reflect what is corporate policy, then the firewall can in no way substitute for that deficiency.

From a firewall perspective, to achieve a scalable and distributable firewall system, Checkpoint has divided the functionality of its Firewall-1 product into two components: a Firewall Module and a Management Module. The interaction of these components makes up the whole of the standard Checkpoint Firewall architecture.

The management module is a centralized controller for the other firewall modules and is where the objects and rules that define firewall functionality exist. The rules and objects can be applied to one or all of the firewall modules. All logs and alerts generated by other firewall modules are sent to this management system for storage, querying, and review.

The firewall module itself is the actual gateway system in which all traffic between separate zones must pass. The firewall module is the system that inspects packets, applies the rules, and generates logs and alerts. It relies on one or more management modules for its rule base and log storage, but may continue to function independently with its current rule base if the management module is not functioning.

An excellent reference to use in the design of firewall architectures is *Building Internet Firewalls* by Elizabeth Zwicky (O'Reilly & Assoc. ISBN: 1565928717).[4]

Step 2: Review Corporate Information System Security Polices

Policy is a critical element of the effective and successful operation of a firewall. A firewall cannot be effective unless deployed in the context of working policies that govern use and administration.

Marcus Ranum defines a firewall as "the implementation of your Internet security policy. If you haven't got a security policy, you haven't got a firewall. Instead, you've got a thing that's sort of doing something, but you don't know what it's trying to do because no one has told you what it should do." Given that, if an organization expects to have a meaningful firewall review in the absence of a set of firewall policies, the organization is in for a rude awakening.

Some policy-based questions to ask during the firewall review include:

- Is there a published firewall policy for the organization?
- Has top management reviewed and approved policies that are relevant to the firewall infrastructure?
- Who has responsibility for controlling the organization's information security?
- Are there procedures to change the firewall policies? If so, what is the process?
- How are these policies communicated throughout the organization?

As to the management of the firewall, some of the issues that must be addressed include:

- Who owns the firewalls, and is this defined?
- Who is responsible for implementing the stated policies for each of the firewalls?
- Who is responsible for the day-to-day management of the firewall?
- Who monitors the firewall for compliance with stated policies?
- How are security-related incidents reported to the appropriate information security staff?
- Are CERT, CIAC, vendor-specific, and similar advisories for the existence of new vulnerabilities monitored?
- Are there written procedures that specify how to react to different events, including containment and reporting procedures?

Change control is critically important for a firewall. Some change controls issues are:

- Ensure that change control procedures documents exist.
- Ensure that test plans are reviewed.
- Review procedures for updating fixes.
- Review the management approval process.
- Process should ensure that changes to the following components are documented:
  - — Any upgrades or patches require notification and scheduling of downtime
  - — Electronic copies of all changes
  - — Hard-copy form filled out for any changes

Finally, backup and contingency planning is crucial when disasters occur. Some issues are:

- *Maintain a golden copy of Firewall-1.* A golden copy is full backup made before the host is connected to the network. This copy can be used for recovery and also as a reference in case the firewall is somehow compromised.
- *Review backup procedures and documentation.* Part of the backup procedures must also include restoration procedures. A backup should only be considered complete if one is able to recover from the backups made. Also, the backups must be stored in a secure location.[5] Should the firewall need to be rebuilt or replaced, there are several files that will need to be restored (see Exhibit 41.1). These files can be backed up via a complete system backup, utilizing an external device such as a

EXHIBIT 41.1 Critical Firewall-1 Configuration Files to Backup

Management Module
$FWDIR/conf/fw.license
$FWDIR/conf/objects.C
$FWDIR/conf/*.W
$FWDIR/conf/rulebases.fws
$FWDIR/conf/fwauth.NDB*
$FWDIR/conf/fwmusers
$FWDIR/conf/gui-clients
$FWDIR/conf/product.conf
$FWDIR/conf/fwauth.keys
$FWDIR/conf/serverkeys.*
Firewall Module
$FWDIR/conf/fw.license
$FWDIR/conf/product.conf
$FWDIR/conf/masters
$FWDIR/conf/fwauth.keys
$FWDIR/conf/product.conf
$FWDIR/conf/smtp.conf
$FWDIR/conf/fwauthd.conf
$FWDIR/conf/fwopsec.conf
$FWDIR/conf/product.conf
$FWDIR/conf/serverkeys.*

See www.phoneboy.com/fw1/faq/0196.html.

tape drive or other large storage device. The most critical files for firewall functionality should be able to fit on a floppy disk.

- *Review backup schedule.*
- *Determine if procedures are in place to recover the firewall system should a disruption of service occur.*
- *Review contingency plan.*
- *Contingency plan documentation.*

Information Security Policies and Procedures (Thomas Peltier, Auerbach Publications) is a good place to start a policy roll-out. While not a panacea for the lack of a comprehensive set of policies, *Information Security Policies and Procedures* enables an organization to quickly roll-out policies without getting bogged down in its composition.

It must be noted that all of this analysis and investigation should be done in the context of the business goals of the organization. While information systems security is about risk management, if it is not implemented within the framework of the corporate strategy, security is bound to fail.

Step 3: Perform Hosts Software Assessment Scan

A firewall misconfiguration can allow unauthorized parties, outsiders, to break into the network despite the firewall's presence. By performing software scans against the individual firewall hosts, specific vulnerabilities can be detected. These scanning tools can identify security holes, detail system weaknesses, validate policies, and enforce corporate security strategies. Such tools are essential for checking system vulnerabilities.

Some of the myriad checks that scanners can identify include:

- Operating system misconfiguration
- Inappropriate security and password settings
- Buffer overflow
- Detection of SANS Top 10 Internet Security Threats

- Segmentation fault affecting FreeBSD
- Detection of unpassworded NT guest and administrator accounts

Some popular scanning tools[6] include:

- NAI Cybercop, http://www.pgp.com/products/cybercop-scanner
- ISS Internet Scanner, http://www.iss.net/internet_scanner/index.php
- SAINT, http://www.wwdsi.com/saint
- Symantec (formerly Axent) NetRecon, http://enterprisesecurity.symantec.
- com/products
- Netcat, http://www.l0pht.com/~weld/netcat/
- nmap, http://www.insecure.org/nmap/index.html

It must be noted that running a host software assessment scan on a firewall is just one aspect of a firewall review. Tools such as Cybercop are extremely easy to run; as such, there is no need to bring in a professional services firm to run the tools. The value added by security professional service firms is in the areas of comprehensive architecture design, analysis, and fault amelioration. Any firm that would run these tools and simply hand the client the output is doing the client a serious injustice.

This only serves to reiterate the point that a security infrastructure must be architected from the onset. This architecture must take into consideration items such as security, capacity, redundancy, and management. Without a good architecture, system redesign will be a constant endeavor.

Step 4: Review Firewall-1 Configuration

While Firewall-1 affords significant security, that security can be compromised if Firewall-1 is misconfigured. Some of the more crucial items to review are listed below (not in any specific order).

IP Forwarding.

Set to *Control IP Forwarding*. IP Forwarding should be disabled in the operating system kernel. This ensures that IP Forwarding will be never be enabled unless Firewall-1 is operating.

Firewall Administrators.

Ensure that the number of Firewall-1 administrators is limited only to those who truly need it. The purpose of every account on the firewall (both for the operating system and the firewall operating system) must be justified. Exhibit 41.2 provides a list of firewall administrators and their permissions.

Trained Staff.

A firewall cannot be effective unless the staff managing the firewall infrastructure is experienced with security and trained in Firewall-1 operations. If a person is made responsible for a firewall simply because he or she has experience with networking, the firewall should be expected to be filled with misconfigurations, which in turn will make it much easier for adversaries to compromise the firewall.

SYN Flood Protection.

Denial-of-service (DoS) attacks enable an attacker to consume resources on a remote host to the degree it cannot function properly. SYN flood attacks are one of the most common types of DoS attacks.

Ensure that SYN flood protection is activated at the appropriate level: None, SYN Gateway, or Passive SYN Gateway (see Exhibit 41.3).

Operating System Version Control.

For both the Checkpoint software and network operating system, ensure that the firewall is running a current and supported version of Firewall-1. While the latest version does not specifically have to be loaded, ensure that current patches are installed.

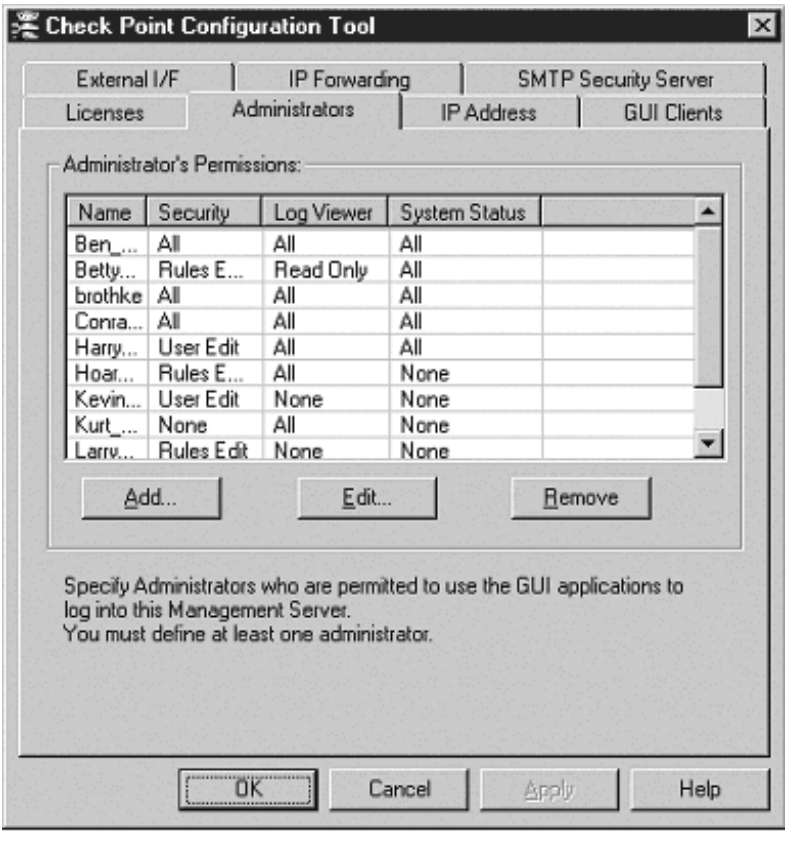


EXHIBIT 41.2 Firewall administrators and their permissions.

Physical Security.

The firewall must be physically secured. It should be noted that all network operating systems base their security models on a secure physical infrastructure. A firewall must be located in areas where access is restricted only to authorized personnel; specifically:

- The local console must be secure.
- The management console should not be open to the external network.
- The firewall configuration should be fully protected and tamper-proof (except from an authorized management station).
- Full authentication should be required for the administrator for local administration.
- Full authentication and an encrypted link are required for remote administration.

Remove Unneeded System Components.

Software such as compilers, debuggers, security tools, etc. should be removed from the firewall.

Adequate Backup Power Supplies.

If the firewall lacks a UPS, security will not be completely enforced in the event of a power disruption.

Log Review.

The logs of both the firewall and network operating system need to be reviewed and analyzed. All events can be traced to the logs, which can be used for debugging and forensic analysis.

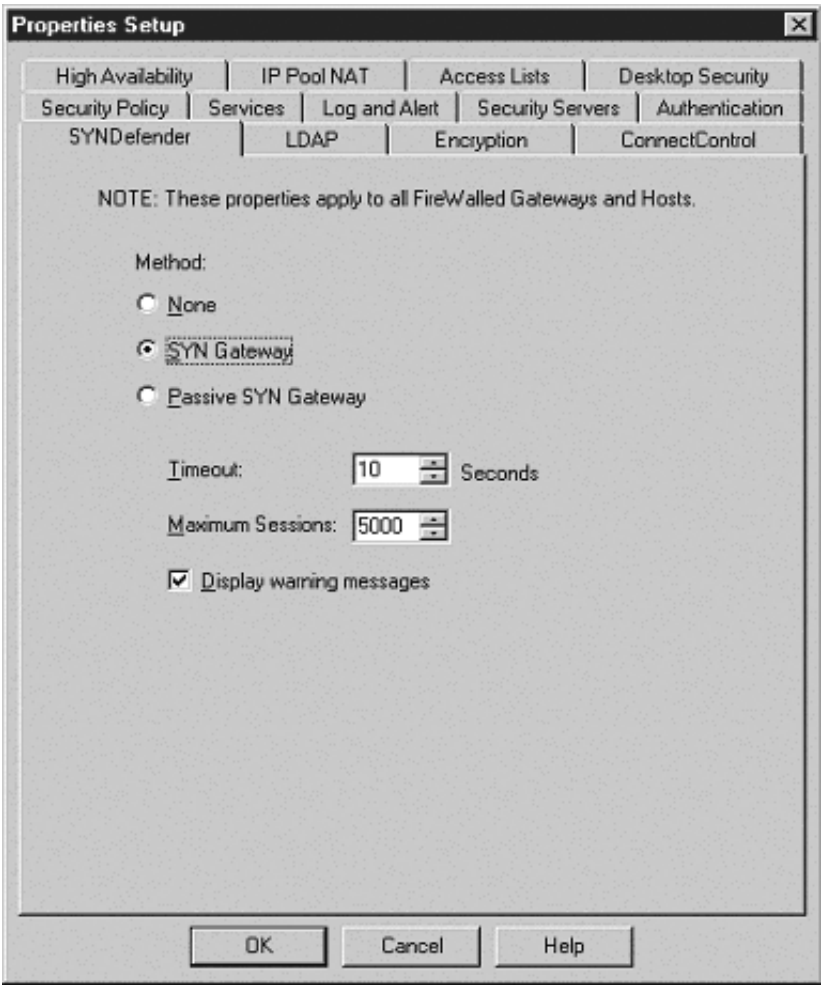


EXHIBIT 41.3 Setting the SYN flood protection.

Ideally, logs should be written to a remote log host or separate disk partition. In the event of an attack, logs can provide critical documentation for tracking several aspects of the incident. This information can be used to uncover exploited holes, discover the extent of the attack, provide documented proof of an attack, and even trace the attack's origin. The first thing an attacker will do is cover his or her tracks by modifying or destroying the log files. In the event that these log files are destroyed, backups will be required to track the incident. Thus, frequent backups are mandatory.

Time Synchronization.

Time synchronization serves two purposes: to ensure that time-sensitive events are executed at the correct time and that different log files can be correlated. Logs that reference an incorrect time can potentially be excluded as evidence in court and this might thwart any effort to prosecute an attacker.

The Network Time Protocol (NTP) RFC 1305 is commonly used to synchronize hosts. For environments requiring a higher grade and auditable method of time synchronization, the time synchronization offerings from Certified Time (www.certifiedtime.com) should be investigated.

Integrity Checking.

Integrity checking is a method to notify a system administrator when something on the file system has changed to a critical file. The most widely known and deployed integrity checking application is Tripwire (www.tripwire.com).

Limit the Amount of Services and Protocols.

A firewall should have nothing installed or running that is not absolutely required by the firewall. Unnecessary protocols open needless communication links. A port scan can be used to see what services are open. Too many services can hinder the efficacy of the firewall, but each service should be authorized; if not, it should be disabled.

Dangerous components and services include:

- X or GUI related packages
- NIS/NFS/RPC related software
- Compilers, Perl, TCL
- Web server, administration software
- Desktop applications software (i.e., Microsoft Office, Lotus Notes, browsers, etc.)

On an NT firewall, only the following services and protocols should be enabled:

- TCP/IP
- Firewall-1
- Protected Storage
- UPS
- RPC
- Scheduler
- Event log
- Plug-and-Play
- NTLM Security Support provider

If other functionality is needed, add them only on an as-needed basis.

Harden the Operating System.

Any weakness or misconfiguration in the underlying network operating system will trickle down to Firewall-1. The firewall must be protected as a bastion host to be the security stronghold. A firewall should never be treated as a general-purpose computing device.

The following are excellent documents on how to harden an operating system:

- *Armoring Solaris*, www.enteract.com/~lspitz/armoring.html
- *Armoring Linux*, www.enteract.com/~lspitz/linux.html
- *Armoring NT*, www.enteract.com/~lspitz/nt.html

Those needing a pre-hardened device should consider the Nokia firewall appliance (www.nokia.com/securitysolutions/network/firewall.html). The Nokia firewall is a hardware solution bundled with Firewall-1. It runs on the IPSO operating system that has been hardened and optimized for firewall functionality.

Firewall-1 Properties.

Exhibit 41.4 shows the Security Policies tab. One should uncheck the Accept boxes that are not necessary:

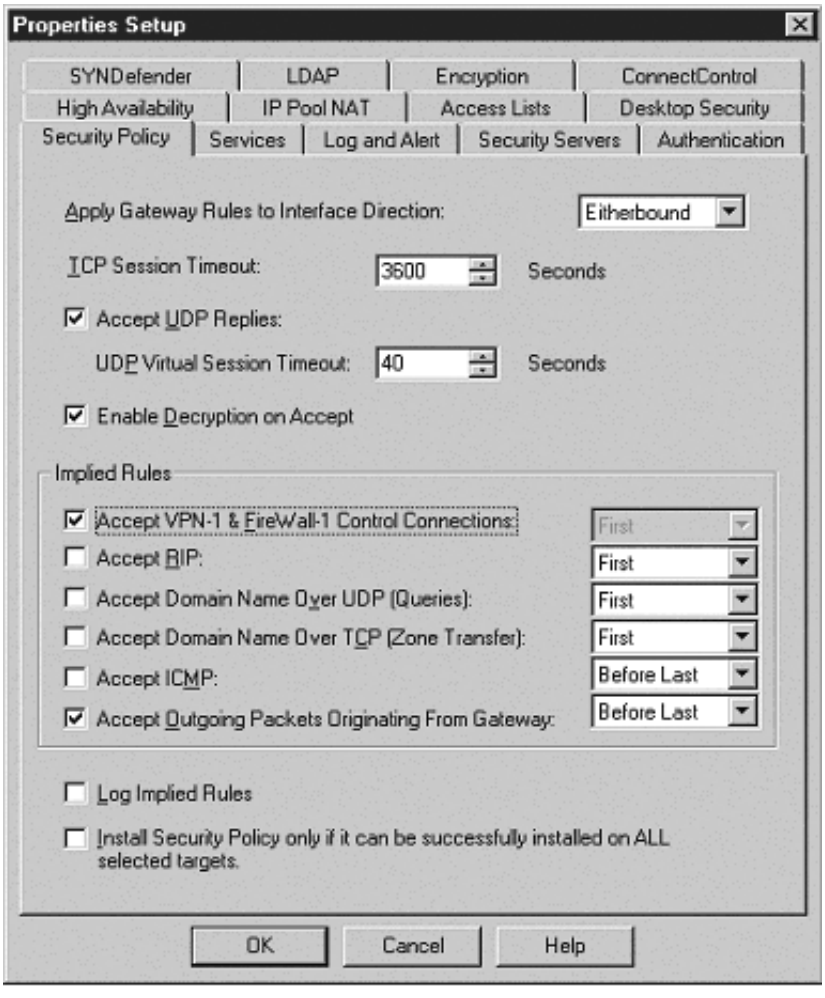


EXHIBIT 41.4 The Security Policy tab.

- *ICMP.* In general, one can disable this property, although one will need to leave it enabled to take advantage of Checkpoint's Stateful Inspection for ICMP in 4.0.
- *Zone transfer.* Most sites do not allow users to perform DNS downloads. The same is true for RIP and DNS lookup options.

Firewall-1 Network Objects.

A central aspect of a Firewall-1 review includes the analysis of all of the defined network objects. Firewall-1 network objects are logical entities that are grouped together as part of the security policy. For example, a group of Web servers could be a simple network object to which a rule is applied. Every network object has a set of attributes, such as network address, subnet mask, etc. Examples of entities that can be part of a network object include:

- Networks and sub-networks
- Servers
- Routers
- Switches
- Hosts and gateway

- Internet domains
- Groups of the above

Firewall-1 allows for the creation of network objects within the source and destination fields. These network objects can contain and referenceanywhere from a single device to entire networks containing thousands of devices. The latter creates a significant obstacle when attempting to evaluate the security configuration and security level of a Firewall-1 firewall. The critical issue is how to determine the underlying security of the network object when it contains numerous objects.

This object-oriented approach to managing devices on Firewall-1 allows the firewall administrator to define routers or any other device as network objects, and then to use those objects within the rules of the firewall security policy. The main uses of network objects are for efficiency in referencing a large amount of network devices. This obviates the need to remember such things as the host name, IP address, location, etc. While network objects provide a significant level of ease of use and time-saving by utilizing such objects, an organization needs to determine if it inherently trusts all of the devices contained within the object. Exhibit 41.5 shows the Network Objects box that shows some of the existing objects. Exhibit 41.6 shows an example of a Network Object with a number of workstations in the group.

As stated, such use of network objects is time-saving from an administrative perspective; but from a security perspective, there is a problem in that any built-in trust that is associated with the network object is automatically created for every entity within that network object. This is due to the fact that in large networks, it is time-consuming to inspect every individual entity defined in the network object. The difficulty posed by such a configuration means that in order to inspect with precision and accuracy the protection that the firewall rule offers, it is essential to inspect every device within the network object.

Step 5: Review Firewall-1 Rule Base

The purpose of a rule base review is to actually see what services and data the firewall permits. An analysis of the rule base is also meant to identify any unneeded, repetitive, or unauthorized rules. The rule base should be made as simple as possible. One way to reduce the number of rules is by combining rules, because sometimes repetitive rules can be merged.

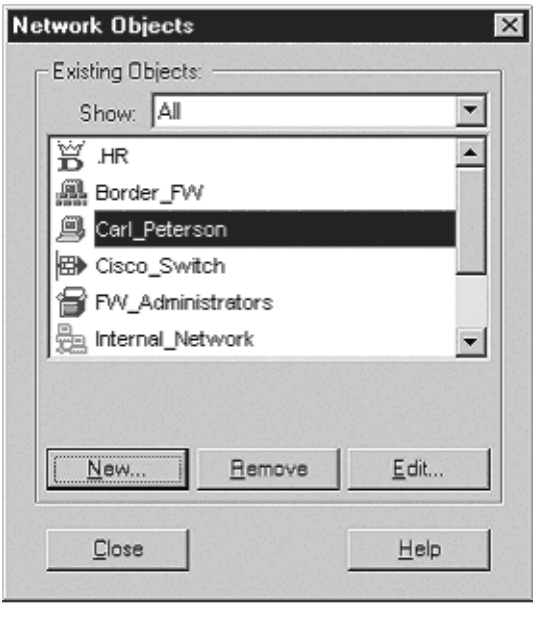


EXHIBIT 41.5 Existing objects.

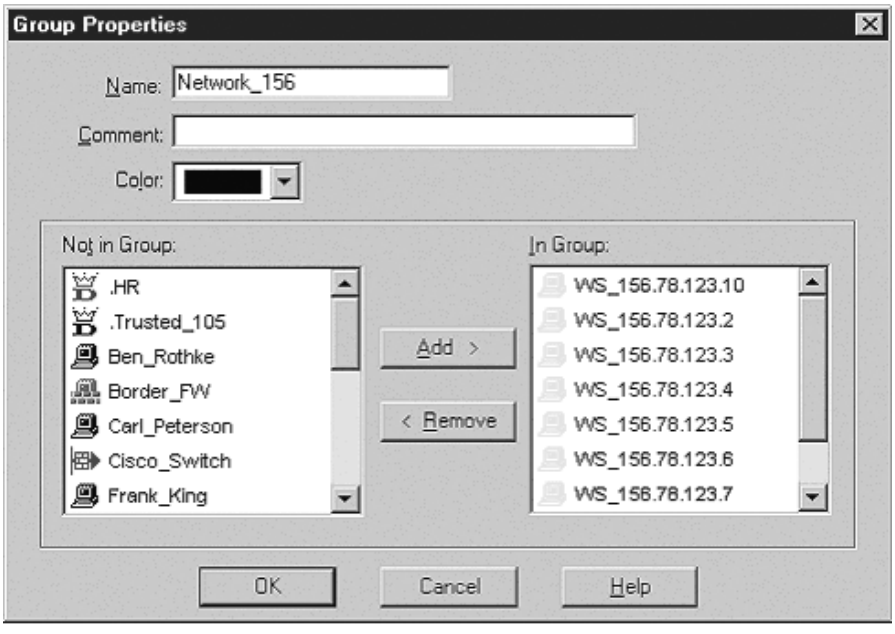


EXHIBIT 41.6 A network object with a number of workstations in the group.

The function of a rule base review is to ensure that the firewall is enforcing what it is expected to. Lance Spitzner writes in *Building Your Firewall Rule Base*[7] that "building a solid rule base is a critical, if not the most critical, step in implementing a successful and secure firewall. Security administrators and experts all often argue what platforms and applications make the best firewalls. However, all of this is meaningless if the firewall rule base is misconfigured."

The rule base is the heart and soul of a Checkpoint firewall. A rule base is a file stored on the firewall that contains an ordered set of rules that defines a distinct security policy for each particular firewall. Access to the rule base file is restricted to those that are either physically at the firewall or a member of the GUI client list specified in the configuration settings.

A rule describes a communication in terms of its source, destination, and service. The rule also specifies whether the communication should be accepted or rejected and whether a log entry is created.

The Firewall-1 inspection engine is a "first-fit" as opposed to a "best-fit" device. This means that if one has a rule base containing 20 rules, and the incoming packet matches rule #4, the inspection engine stops immediately (because rules are examined sequentially for each packet) and does not go through the remainder of the rule base.

As for the rule base review, security expert Lance Spitzer recommends that the goal is to have no more than 30 rules. Once there are more then 30 rules, things exponentially grow in complexity and mistakes then happen.

Each rule base has a separate name. It is useful to standardize on a common naming convention. A suggested format is: firewall-name_administrators-initials_date-of-change; for example, fw1_am_071298.

The result of this naming convention is that the firewall administrator knows exactly which firewall the rule base belongs to; when the rule base was last changed; and who last modified the current configuration. For the rule base review, each and every rule must be examined.

An example of a simple rule base with six rules is as shown in Exhibit 41.7:

- **Rules 1 and 2** enforce the concept of the stealth rule, in that nothing should be able to connect directly to the firewall, other than administrators that are GUI authorized. Rule 1 tells Firewall-1 to drop any packet unless it is from a member of the FW_Administrators group. The Firewall-1 service is predefined and defines all the Firewall-1 administrative ports. For the stealth rule, one

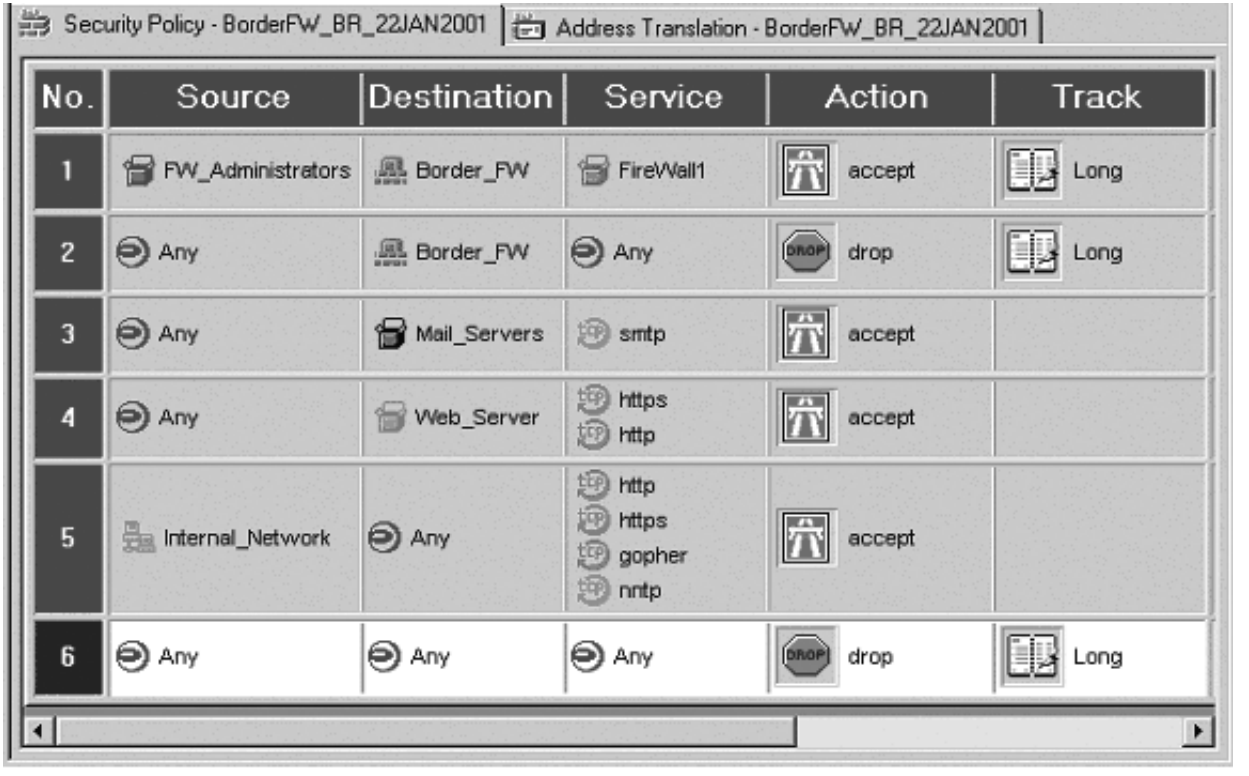


EXHIBIT 41.7 A simple rule base.

specifically wants to drop the packet, as opposed to rejecting it. A rejected packet tells the sender that there is something on the remote side, while a dropped packet does not necessarily indicate a remote host. In addition, this rule is logged; thus, detailed information can be gleaned about who is attempting to make direct connections to the firewall.

- **Rule 3** allows any host e-mail connectivity to the internal mail servers.
- **Rule 4** allows any host HTTP and HTTPS connectivity to internal Web servers.
- **Rule 5** allows internal host connectivity to the Internet for the four specified protocols.
- **Rule 6** is the cleanup rule. Any packet not handled by the firewall at this point will be dropped and logged. The truth is that any packet not handled by the firewall at that point would be dropped anyway. The advantage to this cleanup rule is that these packets will be logged. In this way, one can see which packets are not being handled by the firewall. This can be of assistance in designing a more scalable firewall architecture.

The above rule base example had only six rules and was rather simple. Most corporate rule bases are more detailed and complex. Going through a rule base containing 50 rules and thousands of network objects could take a while to complete.

Exhibit 41.8 displays a rule base that is a little more involved:

- **Rule 1** enforces the stealth rule.
- **Rules 2–4** allow mail traffic between the mail servers and clients.
- **Rule 5** allows any host HTTP connectivity to internal Web servers.
- **Rule 6** stops traffic between the DMZ and an intranet.
- **Rules 7–8** stop incoming and outgoing traffic between the DMZ and an intranet.
- **Rule 9** drop protocols that cause a lot of traffic — in this case, nbdatagram, nbname, and nbsession.
- **Rule 10** is the cleanup rule.

When performing a review and there is doubt that a specific rule is needed, it can be disabled. As a general rule, if a rule is disabled and no one complains, then the rule can be deleted. Exhibit 41.9 shows an example of a disabled rule.

Implied Pseudo-Rules

Implied pseudo-rules are rules that do not appear in the normal rule base, but are automatically created by Firewall-1 based on settings in the Properties Setup of the Security Policy.[8] These rules can be viewed along with the rule base in the Security Policy GUI application. Exhibit 41.10 displays an example of the implied pseudo-rules from a rule base with a single rule.

Although the single and only rule implicitly drops all traffic, there is a lot of traffic that can still pass through the firewall. As seen from these implied pseudo-rules, most of the connectivity deals with the internal operations of the firewall.

Step 6: Put It All Together in a Report

After all the work has been completed, the firewall review needs to be documented. The value in a post-review report is that it can be used as a resource to correct the anomalies found.

As previously stated, the ease of use afforded by scanning tools makes the creation of a huge report effortless. But for a firewall review to have value for a client, it should contain the following:

- Current security state: detail the baseline of the current networking environment and current security posture; this must reference the corporate risk assessment to ensure synchronization with the overall security goals of the organization
- Identification of all security vulnerabilities

Security Policy - DMZ2_BR_12JAN2001 | Address Translation - DMZ2_BR_12JAN2001

No.	Source	Destination	Service	Action	Track
1	Any	Main_FW	Any	drop	Alert
2	Intranet_NY	Mail_Server	pop-3	accept	Long
3	Any	Mail_Server	smtp	accept	Long
4	Mail_Server	Any	smtp	accept	Long
5	Any	Web_Servers	http	accept	Long
6	DMZ_Net	Intranet_NY	Any	reject	Alert
7	Intranet_NY	DMZ_Net	Any	reject	Alert
8	Intranet_NY	Any	Permitted_Internet_Services	accept	Long
9	Any	Any	Chatty_Protocols	drop	
10	Any	Any	Any	drop	Alert

EXHIBIT 41.8 Complex rule base.

EXHIBIT 41.9 Disabled rule.

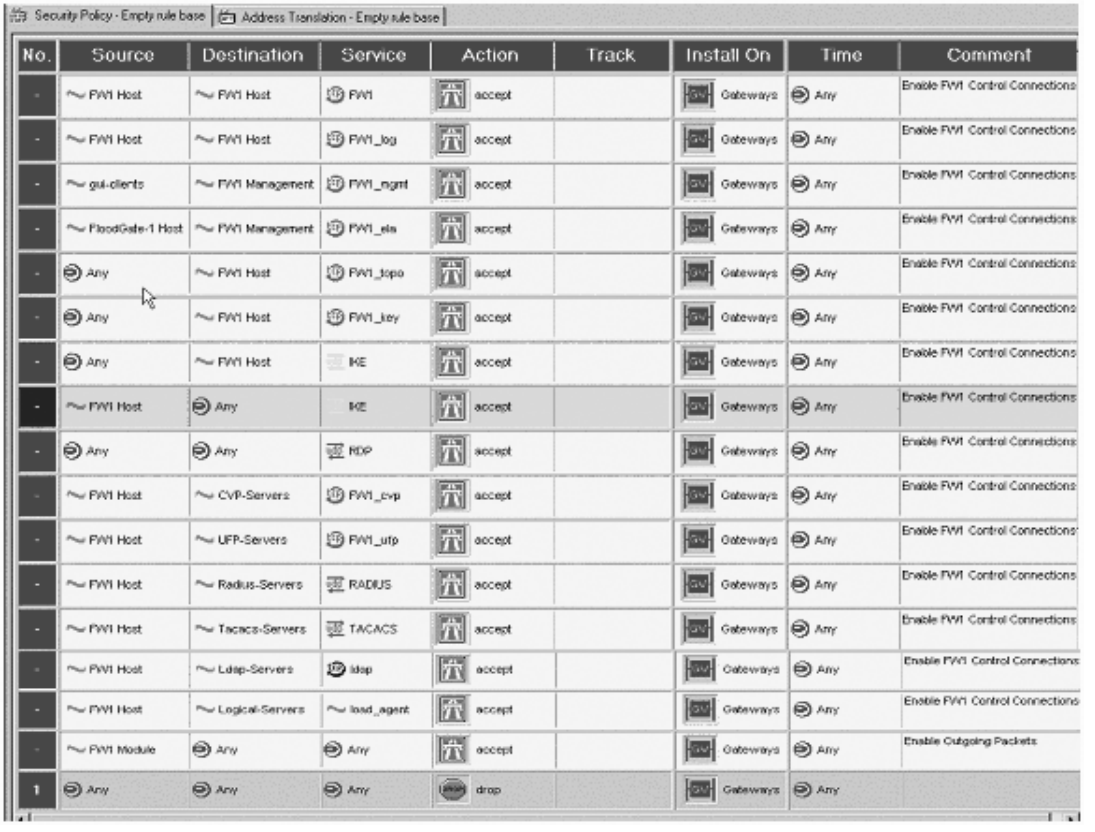


EXHIBIT 41.10 Implied pseudo-rules.

- Recommend corrections, solutions, and implementation priorities; a detailed implementation plan must be provided, showing how all of the solutions and fixes will coalesce
- Detailed analysis of security trade-offs, relating the risks to cost, ease of use, business requirements, and acceptable levels of risk
- Provide baseline data for future reference and comparison to ensure that systems are rolled out in a secure manner

Conclusion

A firewall is effective to the degree that it is properly implemented. And in today's corporate environments, it is easy for a firewall to become misconfigured. By reviewing the firewall setup, firewall administrators can ensure that their firewall is enforcing what they expect it to, in a secure manner. This makes for good sense and good security.

Notes

1. Screen shots in this chapter are from Firewall-1 v4.1 for Windows NT, but are germane for all platforms and versions. See www.phoneboy.com/fw1/docs/4.0-summary.html for the new features and up-grades in Firewall-1 version 4.x.
2. PricewaterhouseCoopers, Ernst & Young, Deloitte & Touche, Arthur Andersen, KPMG.
3. See http://web.mit.edu/security/www/gassp1.html for more information about GASSP.
4. Also see *Choosing the Right Firewall Architecture Environment* by B. Rothke (June 1998, *Enterprise Systems Journal*, http://www.esj.com/library/1998/june/0698028.htm).
5. It should be noted that while many safes will physically protect backup media, they will not protect this media against the heat from a fire. The safe must be specifically designed for data storage of media such as tapes, floppies, and hard drives.
6. A comprehensive list of tools can be found at www.hackingexposed.com/tools/tools.html.
7. See www.enteract.com/~lspitz.
8. See www.phoneboy.com/fw1/faq/0345.html for a comprehensive list of what the Firewall-1 control connections allow by default.

References

Checkpoint Knowledge Base, http://support.checkpoint.com/public/.

Checkpoint resource library, http://cgi.us.checkpoint.com/rl/resourcelib.asp.

Phoneboy, www.phoneboy.com, excellent Firewall-1 resource with large amounts of technical information.

Auditing Your Firewall Setup, Lance Spitzner, www.enteract.com/~lspitz/audit.html, www. csiannual. com/pdf/f7f8.pdf.

Building Your Firewall Rule Base, Lance Spitzner, www.enteract.com/~lspitz.

Firewall-1 discussion threads, http://msgs.securepoint.com/fw1/.

SecurityPortal, www.securityportal.com; latest and greatest firewall products and security news.

Marcus Ranum, Publications, Rants, Presentations & Code.

Pragmatic security information, http://web.ranum.com/pubs/index.shtml.

Internet Firewalls Frequently Asked Questions, www.interhack.net/pubs/fwfaq/.

SecurityFocus.com, www.securityfocus.com.

ICSA Firewall-1 Lab Report, www.icsa.net/html/communities/firewalls/certification/vendors/ checkpoint/ firewall1/nt/30a_report.shtml.

WebTrends Firewall Suite, www.webtrends.com/products/firewall/default.htm.

Intrusion Detection for FW-1, http://www.enteract.com/~lspitz.

Further Reading

Zwicky, Elizabeth, Building Internet Firewalls, O'Reilly & Assoc., 2000, ISBN: 1565928717.
Cheswick, William and S. Bellovin, *Firewalls and Internet Security*, Addison Wesley, 2001, ISBN: 020163466X.
Garfinkel, Simson and G. Spafford, *Practical UNIX and Internet Security*, O'Reilly & Associates, 1996, ISBN 1-56592-148-8.
Norberg, Stefan, Securing Windows NT/2000 Server, O'Reilly & Associates, 2001, ISBN 1-56592-768-0.
Scambray, Joel, S. McClure, and G. Kurtz, Hacking Exposed: Network Security Secrets and Solutions, McGraw-Hill, 2000, ISBN: 0072127481.

Resources and Mailing Lists

CERT/CC Advisories, www.cert.org/contact_cert/certmaillist.html.
@stake, http://www.atstake.com/research/advisories/index.html.
CIAC, http://ciac.llnl.gov/.
Firewall-1 mailing list, www.checkpoint.com/services/mailing.html.
Firewalls mailing list, http://lists.gnac.net/firewalls/.
Firewall Wizards list, www.nfr.com/forum/firewall-wizards.html.
CERIAS, www.cerias.purdue.edu/.
Bugtraq, Bugtraq-request@fc.net.
NTBugtraq, Ntbugtraq-request@fc.net.
ISS X-Force Advisories, www.iss.net/mailinglist.php.
Sun, www.sun.com/security/siteindex.html.
Microsoft, www.microsoft.com/security.
SANS, www.sans.org.

42

Comparing Firewall Technologies

Per Thorsheim

In early January 2001, a new Web page was launched. It was named Netscan,[1] and the creators had done quite a bit of work prior to launching their Web site. Actually, the work was quite simple, but time-consuming. They had pinged the entire routed IPv4 address space; or to be more exact, they pinged every IP address ending with .0 or .255. For each PING sent, they expected one PING REPLY in return. And for each network that replied with more than one packet, they counted the number of replies and put the data into a database. All networks that did reply with more than one packet for each packet sent were considered to be an amplifier network. After pinging the entire Internet (more or less), they published on their Web site a list of the 1024 worst networks, including the e-mail address for the person responsible for the IP address and its associated network. The worst networks were those networks that gave them the highest number of replies to a single PING, or the best amplification effect.

The security problem here is that it is rather easy to send a PING request to a network, using a spoofed source IP address. And when the recipient network replies, all those replies will be sent to the source address as given in the initial PING. As shown in Exhibit 42.1, the attacker can flood the Internet connection of the final recipient by repeating this procedure continuously.

In fact, the attacker can use an ISDN connection to create enough traffic to jam a T3 (45-Mbit) connection, using several SMURF amplifier networks to launch the attack. And as long as there are networks that allow such amplification, a network can be the target of the attack even if the network does not have the amplification problem itself, and there is not much security systems such as firewalls can do to prevent the attack.

This type of attack has been used over and over again to attack some of the biggest sites on the Internet, including the February 2000 attacks against Yahoo, CNN, Ebay, and Amazon.

Today, there are several Web sites that search for SMURF amplifier networks and publish their results publicly. In a presentation given in March 2001, this author pointed out the fact that the number of networks not protected from being used as such amplifiers had increased more than 1000 percent since January 2001.

One of the interesting findings from these attacks was that routers got blamed for the problems — not firewalls. And they were correct; badly configured Internet routers were a major part of the problem in these cases. Even worse is the fact that the only requirement for blocking this specific PING-based attack was to set one parameter in all routers connecting networks to the Internet. This has now become the recommended default in RFC 2644/BCP 34, "Changing the Default for Directed Broadcast in Routers." Security professionals should also read RFC 2827/BCP 0038, "Network Ingress Filtering: Defeating Denial-of-Service Attacks Which Employ IP Source Address Spoofing," to further understand spoofing attacks.

Another interesting observation after these attacks was President Clinton's announcement of a National Plan for Information Systems Protection, with valuable help from some of the top security experts in the United States. In this author's opinion, this serves as the perfect example of who should be at the top and responsible for security — the board of directors and the CEO of a company.

Finally, Web sites such as CNN, Yahoo, and Amazon all had firewalls in place, yet that did not prevent these attacks. Thus, a discussion of firewall technologies and what kind of security they can actually provide is in order.

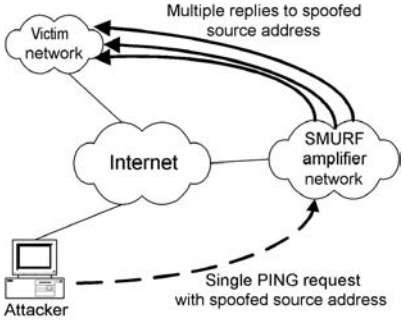


EXHIBIT 42.1 Attacker using spoofed PING packets to flood a network by using a vulnerable intermediary network.

Firewall Technologies Explained

The Internet Firewalls FAQ[2] defines two basic types of firewalls: network-layer firewalls and application-layer firewalls (also referred to as application proxy firewalls, or just proxies). For this chapter, stateful inspection firewalls are defined as a mix of the first two firewall types, in order to make it easier to understand the similarities and differences between them.

The reader may already be familiar with the OSI layer model, in which the network layer is layer 3 and the application layer is at layer 7, as shown in Exhibit 42.2.

A firewall can simply be illustrated as a router that transmits packets back and forth between two or more networks, with some kind of security filtering applied on top.

Network-Level Firewalls: Packet Filters

Packet filter firewalls are very often just a router with access lists. In its most basic form, a packet filter firewall controls traffic based on the source and destination IP address of each IP packet and the destination port. Many packet filter firewalls also allow checking the packets based on the incoming interface (is it coming from the Internet, or the internal network?). They may also allow control of the IP packet based on the source port, day and time, protocol type (TCP, UDP, or ICMP), and other IP options as well, depending on the product.

The first thing to remember about packet filter firewalls is that they inspect every IP packet by itself; they do not see IP packets as part of a session. The second thing to remember about packet filter firewalls is that many of them, by default, have a fail-open configuration, meaning that, by default, they will let packets through unless specifically instructed not to. And finally, packet filters only check the HEADER of a packet, and not the DATA part of the packet. This means that techniques such as tunneling a service within another service will easily bypass a packet filter (e.g., running Telnet on port 80 through a firewall where the standard Telnet port 23 is blocked, but HTTP port 80 is open. Because the packet filter only sees source/destination and port number, it will allow it to pass).

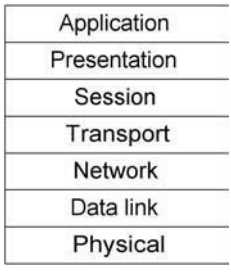


EXHIBIT 42.2 The OSI seven-layer model.

Why Use Packet Filter Firewalls?

Some security managers may not be aware of it, but most probably there are lots of devices already in their network that can do packet filtering. The best examples are various routers. Most (if not all) routers today can be equipped with access lists, controlling IP traffic flowing through the router with various degrees of security. In many networks, it will just be a matter of properly configuring them for the purpose of acting as a packet filter firewall. In fact, the author usually recommends that all routers be equipped with at least a minimum of access lists, in order to maintain security for the router itself and its surroundings at a minimal level. Using packet filtering usually has little or no impact on throughput, which is another plus over the other technologies. Finally, packet filter firewalls support most (if not all) TCP/IP-based services.

Why Not Use Packet Filter Firewalls?

Well, they only work at OSI layer 3, or the network layer as it is usually called. Packet filter firewalls only check single IP packets; they do not care whether or not the packet is part of a session. Furthermore, they do not do any checking of the actual contents of the packet, as long as the basic header information is okay (such as source and destination IP address). It can be frustrating and difficult to create rules for packet filter firewalls, and maintaining consistent rules among many different packet filter firewalls is usually considered very difficult. As previously mentioned, the typical fail-open defaults should be considered dangerous in most cases.

Stateful Inspection Firewalls

Basically, stateful inspection firewalls are the same thing as packet filter firewalls, but with the ability to keep track of the state of connections in addition to the packet filtering abilities. By dynamically keeping track of whether a session is being initiated, currently transmitting data (in either direction), or being closed, the firewall can apply stronger security to the transmission of data. In addition, stateful inspection firewalls have various ways of handling popular services such as HTTP, FTP, and SMTP. These last options (of which there are many variants of from product to product) enable the firewall to actually check whether or not it is HTTP traffic going to TCP port 80 on a host in a network by "analyzing" the traffic. A packet filter will only assume that it is HTTP traffic because it is going to TCP port 80 on a host system; it has no way of actually checking the DATA part of the packet, while stateful inspection can partially do this.

A stateful inspection firewall is capable of understanding the opening, communication, and closing of sessions. Stateful inspection firewalls usually have a fail-close default configuration, meaning that they will not allow a packet to pass if they do not know how to handle the packet. In addition to this, they can also provide an extra level of security by "understanding" the actual contents (the data itself) within packets and sessions, compared to packet filters. This last part only applies to specific services, which may be different from product to product.

Why Use Stateful Inspection Firewalls?

Stateful inspection firewalls give high performance and provide more security features than packet filtering. Such features can provide extra control of common and popular services. Stateful inspection firewalls support most (if not all) services transparently, just like packet filters, and there is no need to modify client configurations or add any extra software for them to work.

Why Not Use Stateful Inspection Firewalls?

Stateful inspection firewalls may not provide the same level of security as application-level firewalls. They let the server and the client talk "directly" to each other, just like packet filters. This may be a security risk if the firewall does not know how to interpret the DATA contents of the packets flowing through the firewall. Even more disturbing is the fact that many people consider stateful inspection firewalls to be easier to configure wrongly, compared to application-level firewalls. This is due to the fact that packet filters and stateful inspection firewalls support most, if not all, services transparently, while application-level firewalls usually support only a very limited number of services and require modification to client software in order to work with non-supported services.

In a white paper from Network Associates,[3] the Computer Security Institute (CSI) was quoted as saying, "It is quite possible, in fact trivial, to configure stateful inspection firewalls to permit dangerous services through the firewall…. Application proxy firewalls, by design, make it far more difficult to make mistakes during configuration."

Of course, it should be unnecessary to say that no system is secure if it is not configured correctly. And human faults and errors are the number one, two, and three reasons for security problems, right?

Application-Level Firewalls

Application-level firewalls (or just proxies) work as a "man-in-the-middle," where the client asks the proxy to perform a task on behalf of the client. This could include tasks such as fetching Web pages, sending mail, retrieving files using FTP, etc. Proxies are application specific, meaning that they need to support the specific application (or, more exactly, the application-level protocol) that will be used. There are also standards for generic proxy functionality, with the most popular being SOCKS. SOCKS was originally authored by David Koblas and further developed by NEC. Applications that support SOCKS will be able to communicate through firewalls that also support the SOCKS standard.[4]

Similar to a stateful inspection firewall, the usual default of an application-level firewall is fail-close, meaning that it will block packets/sessions that it does not understand how to handle.

Why Use Application-Level Firewalls?

First of all, they provide a high level of security, primarily based on the simple fact that they only support a very limited number of services; however, they do support most, if not all, of the usual services that are needed on a day-to-day basis. They understand the protocols at the application layer and, as such, they may block parts of a protocol (allow receiving files using FTP, but denying sending files using FTP as an example). They can also detect and block vulnerabilities, depending on the firewall vendor and version.

Furthermore, there is no direct contact being made between the client and the server; the firewall will handle all requests and responses for the client and the server. With a proxy server, it is also easy to perform user authentication, and many security practitioners will appreciate the extensive level of logging available in application-level firewalls.

For performance reasons, many application-level firewalls can also cache data, providing faster response times and higher throughput for access to commonly accessed Web pages, for example. The author usually does not recommend that a firewall do this because a firewall should handle the inspection of traffic and provide a high level of security. Instead, security practitioners should consider using a stand-alone caching proxy server for increasing performance while accessing common Web sites. Such a stand-alone caching proxy server may, of course, also be equipped with additional content security, thus controlling access to Web sites based on content and other issues.

Why Not Use Application-Level Firewalls?

By design, application-level firewalls only support a limited number of services. If support for other applications/services/protocols is desired, applications may have to be changed in order to work through an application-level firewall. Given the high level of security such a firewall may provide (depending on its configuration, of course), it may have a very negative impact on performance compared to packet filtering and stateful inspection firewalls.

What the Market Wants versus What the Market Really Needs

Many firewalls today seem to mix these technologies together into a simple and easy-to-use product. Firewalls try to be a "turnkey" or "all-in-one" solution. Security in a firewall that can be configured by more or less plugging it in and turning it on is something in which this author has little faith. And, the all-in-one solution that integrates VPN, anti-virus, content security/filtering, traffic shaping, and similar functionality is also something in which this author has little trust. In fact, firewalls seem to get increasingly complex in order to make them easier to configure, use, and understand for the end users. This seems a little bit wrong; by increasing the amount of code in a product, the chances of security vulnerabilities in the product increase, and most probably exponentially.

In the author's opinion, a firewall is a "black box" in a network, which most regular users will not see or notice. Users should not even know that it is there.

The market decides what it wants, and the vendors provide exactly that. But does the market always know what is good for it? This is a problem that security professionals should always give priority to — teaching security understanding and security awareness.

Firewall Technologies: Quick Summary

As a rule of thumb, packet filters provide the lowest level of security, but the highest throughput. They have limited security options and features and can be difficult to administrate, especially if there is a large number of them in a network.

Stateful inspection firewalls provide a higher level of security, but may not give the same throughput as packet filters. The leading firewalls on the market today are stateful inspection firewalls, often considered the best mix of security, manageability, throughput, and transparent integration into most environments.

Application-level firewalls are considered by many to give the highest level of security, but will usually give less throughput compared to the two other firewall technologies.

In any case, security professionals should never trust a firewall by itself to provide good security. And no matter what firewall a company deploys, it will not provide much security if it is not configured correctly. And that usually requires quite a lot of work.

Perimeter Defense and How Firewalls Fit In

Many people seem to believe that all the bad hackers are "out there" on the Internet, while none of their colleagues in a firm would ever even think of doing anything illegal, internally or externally. Sadly, however, there are statistics showing that internal employees carry out maybe 50 percent of all computer-related crime.

This is why it is necessary to explain that security in a firewall and its surrounding environment works two ways. Hackers on the Internet are not allowed access to the internal network, and people (or hostile code such as viruses and Trojans) on the internal network should be prevented from sending sensitive data to the external network. The former is much easier to configure than the latter. As a practical example of this, here is what happened during an Internet penetration test performed by the author some time ago.

Practical Example of Missing Egress (Outbound) Filtering

The client was an industrial client with a rather simple firewall environment connecting them to the Internet. They wanted a high level of security and had used external resources to help configure their Internet router act as a packet filter firewall, in addition to a stateful inspection firewall on the inside of the Internet router, with a connection to the internal network. They had configured their router and firewall to only allow e-mail (SMTP, TCP port 25) back and forth between the Internet and their anti-virus (AV) e-mail gateway placed in a demilitarized zone (DMZ) on the stateful inspection firewall. The anti-virus e-mail gateway would check all in- and outgoing e-mail before sending it to the final recipient, be it on the internal network or on the Internet. The router was incredibly well configured; inbound access lists were extremely strict, only allowing inbound SMTP to TCP port 25. The same thing was the case for the stateful inspection firewall.

While testing the anti-virus e-mail gateway for SMTP vulnerabilities, the author suddenly noticed that each time he connected to the SMTP connector of the anti-virus e-mail gateway, it also sent a Windows NetBIOS request in return, in addition to the SMTP login banner.

This simple fact reveals a lot of information to an unauthorized person (see Exhibit 42.3). First of all, there is an obvious lack of egress (outbound) filtering in both the Internet router and the firewall. This tells us that internal systems (at least this one in the DMZ) can probably do NetBIOS communication over TCP/IP with external systems. This is highly dangerous for many reasons. Second, the anti-virus e-mail gateway in the DMZ is installed with NetBIOS, which may indicate that recommended good practices have not been followed for installing a Windows server in a high-security environment. Third, it may be possible to use this system to access other systems in the DMZ or on other networks (including the internal network) because NetBIOS is being used for communication among windows computers in a workgroup or domain. At least this is the author's usual experience when doing Internet penetration testing. Of course, an unauthorized person must break into the server in the DMZ first, but that also proves to be easier than most people want to believe.

How Can One Prevent Such Information Leakage?

Security managers should check that all firewalls and routers connecting them to external networks have been properly configured to block services that are considered "dangerous," as well as all services that are never supposed to be used against hosts on external networks, especially the Internet.

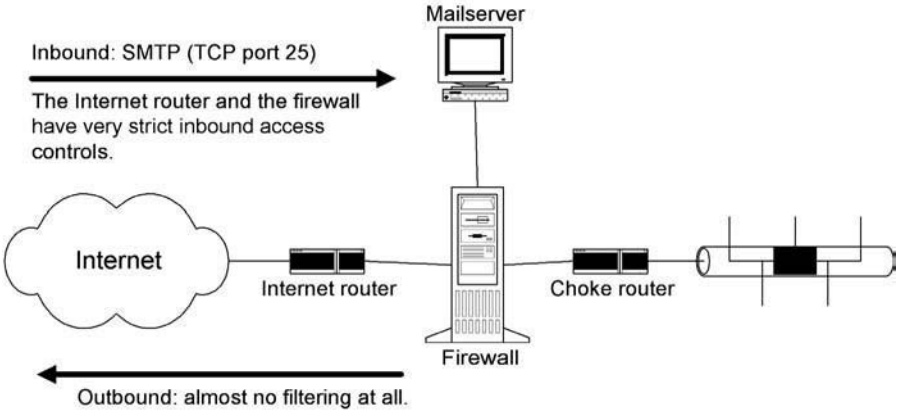


EXHIBIT 42.3 Missing egress filtering in the router and the firewall may disclose useful information to unauthorized people.

As a general rule, security managers should never allow servers and systems that are not being used at the local console to access the Internet in any way whatsoever. This will greatly enhance security, in such a way that hostile code such as viruses and Trojans will not be able to directly establish contact with and turn over control of the system to unauthorized persons on any external network.

This also applies to systems placed in a firewall DMZ, where there are systems that can be accessed by external people, even without any kind of user authentication. The important thing to remember here is: who makes the initial request to connect to a system?

If it is an external system making a connection to a mail server in a DMZ on TCP port 25 (SMTP), it is okay because it is (probably) incoming e-mail. If the mail server in the DMZ makes a connection to an external system on TCP port 25, that is also okay because it does this to send outgoing e-mail. However, if the only purpose of the mail server is to send and receive mail to and from the Internet, the firewalls and even the routers should be configured in accordance with this.

For the sake of easy administration, many people choose to update their servers directly from the Internet; some even have a tendency to sit directly on production servers and surf the World Wide Web without any restrictions or boundaries whatsoever. This poses a high security risk for the server, and also the rest of the surrounding environment, given the fact that (1) Trojans may get into the system, and (2) servers tend to have the same usernames and passwords even if they do not have anything in common except for being in the same physical/logical network.

To quote Anthony C. Zboralski Gaius[5] and his article "Things to Do in Cisco Land when You're Dead" in *Phrack Magazine*[6]:

> It's been a long time since I stopped believing in security. The core of the security problem is really because we are trusting trust (read Ken Thomson's article, Reflections on Trusting Trust). If I did believe in security then I wouldn't be selling penetration tests.

It can never be said that there is a logical link between high security and easy administration, nor will there ever be. Security is difficult, and it will always be difficult.

Common Mistakes that Lead to System and Network Compromises

Many security professionals say that "networks are hard on the outside, and soft on the inside," a phrase this author fully agrees with. The listing that follows shows some of the common weaknesses encountered over and over again.

- Remote access servers (RAS) are connected to the internal network, allowing intruders access to the network just like internal users, as soon as they have a username and password.
- Access lists and other security measures are not implemented in WAN routers and networks. Because small regional offices usually have a lower level of physical security, it may be easier to get access to the office, representing a serious risk to the entire network.

- Many services have default installations, making them vulnerable. They have known weaknesses, such as standard installation paths; default file and directory permissions that give all uses full control of the system, etc.
- Employees do not follow written password policies, and password policies are usually written with users (real people) in mind, and not generic system accounts.
- Many unnecessary services are running on various systems without being used. Many of these services can easily be used for denial-of-service (DoS) attacks against the system and across the network.
- Service applications run with administrator privileges, and their passwords are rarely changed from the default value. As an example, there are backup programs in which the program's username and password are the same as the name of the program, and the account has administrative privileges by default. Take a look at some of the default usernames/passwords lists that exist on the Internet; they list hundreds of default usernames and passwords for many, many different systems.[7]
- Companies have trust in authentication mechanisms and use them as their only defense against unauthorized people trying to get access to the various systems in the network. Many companies and people do not seem to understand that hackers do not need a username or password to get access to different systems; there are many vulnerabilities that give them full control within seconds.

Most, if not all, security professionals will recognize many of these as problems that will never go away. At the same time, it is very important to understand these problems, and professionals should work continuously to reduce or remove these problems.

When performing penetration testing, common questions and comments include: "How are you going to break into our firewall?" and "You are not allowed to do this and this and that." First of all, penetration testing does not involve breaking into firewalls, just trying to bypass them. Breaking into a firewall by itself may show good technical skills, but it does not really do much harm to the company that owns it. Second, hackers do not have to follow any rules, either given by the company they attack or the laws of the country. (Or the laws of the many countries they are passing through in order to do the attack over the Internet, which opens up lots more problems for tracking down and punishing the hackers, a problem that many security professionals are trying to deal with already.)

What about Security at the Management Workstations?

Many companies are deploying extremely tight security into their Internet connection environment and their internal servers. What many of them do wrong is that they forget to secure the workstations that are being used to administrate those highly secured systems. During a recent security audit of an Internet bank, the author was given an impressive presentation with firewalls, intrusion detection systems, proxies, and lots of other stuff thrown in. When checking a bit deeper, it was discovered that all the high-security systems were managed from specific workstations located on their internal network. All those workstations ("owned" by network administrators) were running various operating systems (network administrators tend to do this…) with more or less default configurations, including default usernames and passwords, SNMP,[8] and various services. All those workstations were in a network mixed with normal users; there were no access restrictions deployed except username/password to get access to those management stations. They even used a naming convention for their internal computers that immediately revealed which ones were being used for "critical system administration." By breaking into those workstations first (Trojans, physical access, other methods), it did not take long to get access to the critical systems.

Intrusion Detection Systems and Firewalls

Lately, more and more companies have been deploying intrusion detection systems (IDSs) in their networks. Here is another area in which it is easy to make mistakes. First of all, an IDS does not really help a company improve its security against hackers. An IDS will help a company to better detect and document an attack, but in most cases it will not be able to stop the attack. It is tempting to say that an IDS is just a new term for extensive logging and automated/manual analysis, which have been around for quite some time now.

Some time ago, someone came up with the bright idea of creating an IDS that could automatically block various attacks, or reconfigure other systems like firewalls to block the attacks. By doing a spoofing attack (very easy these days), hackers could create a false attack that originated from a trusted source (third party), making the IDS block all communications between the company and the trusted source. And suddenly everybody understood that the idea of such automated systems was probably a bad idea.

Some IDSs are signature based, while others are anomaly based. Some IDSs have both options, and maybe host and network based agents as well. And, of course, there are central consoles for logging and administrating the IDS agents deployed in the network. (How good is the security at those central consoles?)

- *Problem 1.* Signature-based detection more or less depends on specific data patterns to detect an attack. Circumventing this is becoming easier every day as hackers learn how to circumvent the patterns known by the IDS, while still making patterns that work against the target systems.
- *Problem 2.* Most IDSs do not understand how the receiving system reacts to the data sent to it, meaning that the IDSs can see an attack, but it does not know whether or not the attack was successful. So, how should the IDS classify the attack and assess the probability of the attack being successful?
- *Problem 3.* IDSs tend to create incredible amounts of false alerts, so who will check them all to see if they are legitimate or not? Some companies receive so many alerts that they just "tune" the system so that it does not create that many alerts. Sometimes this means that they do not check properly to see if there is something misconfigured in their network, but instead just turn off some of the detection signatures, thus crippling the IDS of its functions.
- *Problem 4.* Anomaly-based detection relies on a pattern of "normal" traffic and then generates alerts based on unusual activity that does not match the "normal" pattern. What is a "normal" pattern? The author has seen IDS deployments in which an IDS was placed into a network that was configured with all sorts of protocols, unnecessary services, and cleartext authentication flying over the wire. The "normal" template became a template for which almost everything was allowed, more or less disabling the anomaly detection capability of the IDS. (This is also very typical for "personal firewalls," which people are installing on their home systems these days.)

An IDS can be a very effective addition to a firewall because it is usually better at logging the contents of the attack compared to a firewall, which only logs information such as source/destination, date/time, and other information from the various IP/TCP/UDP headers. Using an IDS, it is also easier to create statistics over longer periods of time of hacker activity compared to just having a firewall and its logs. Such statistics may also aid in showing management what the reality is when it comes to hacking attempts and illegal access against the company's systems, as well as raising general security awareness among its users.

On the other hand, an IDS requires even more human attention than a firewall, and a company should have very clearly defined goals with such a system before buying and deploying it. Just for keeping hackers out of your network is not a good enough reason.

General Recommendations and Conclusions

A firewall should be configured to protect itself, in addition to the various networks and systems that it moves data to and from. In fact, a firewall should also "protect" the Internet, meaning that it should prevent internal "hackers" from attacking other parties connected to the Internet, wherever and whoever they are. Surrounding network equipment such as routers, switches, and servers should also be configured to protect the firewall environment in addition to the system itself.

Security professionals should consider using user authentication before allowing access to the Internet. This will, in many situations, block viruses and Trojans from establishing contact with hosts on the Internet using protocols such as HTTP, FTP, and Telnet, for example.

It may be unnecessary to say, but personal use of the Internet from a company network should, in general, be forbidden. Of course, the level of control here can be discussed, but the point is to prevent users from downloading dangerous content (viruses, Trojans) and sending out files from the internal network using protocols such as POP3, SMTP, FTP, HTTP, and other protocols that allow sending files in ASCII or binary formats.

Finally, other tools should be deployed as well to bring the security to a level that actually matches the level required (or wanted) in the company security policy. In the author's experience, probably less than 50 percent of all firewall installations are doing extensive logging, and less than 5 percent of the firewall owners are actually doing anything that even resembles useful log analysis, reporting, and statistics. To some, it seems like the attitude is "we've got a firewall, so we're safe." Such an attitude is both stupid and wrong.

Firewalls and firewall technologies by themselves cannot be trusted, at least not in our present Internet age of communications with hackers hiding in every corner. Hackers tunneling data through allowed protocols and ports can easily bypass today's firewalls, using encryption schemes to hide their tracks. Security professionals should, nonetheless, understand that a firewall, as part of a consistent overall security architecture, is still an important part of the network security in a company.

The best security tool available is still the human brain. Use it wisely and security will improve.

Notes

1. www.netscan.org.
2. http://www.interhack.net/pubs/fwfaq/, Copyright © Marcus J. Ranum and Matt Curtin.
3. Network Associates, "Adaptive Proxy Firewalls — The Next Generation Firewall Architecture."
4. Note that there are two major versions of SOCKS: SOCKS V4 and SOCKS V5. Version 4 does not support authentication or UDP proxying, while version 5 does.
5. www.hert.org, quoted with permission.
6. www.phrack.com.
7. http://packetstorm.securify.com/ is a good place to search for such lists, and much more useful information as well.
8. Simple Network Management Protocol, one of the author's favorite ways of mapping large networks fast and easy. Also mentioned as number 10 on the SANS' Institute "Top Ten Vulnerabilities" list at http://www.sans.org/topten.htm.

43

The (In)Security of Virtual Private Networks

James S. Tiller, CISA, CISSP

It is no surprise that virtual private networks (VPN) have become tremendously popular among many dissimilar business disciplines. Regardless of the vertical market or trade, VPNs can play a crucial role in communication requirements, providing flexibility and prompt return on investment when implemented and utilized properly. The adoption of VPNs has been vast and swift; and as technology advances, this trend will only increase. Some of the popularity of VPNs is due to the perceived relative ease of implementing the technology. This perceived simplicity and the promise of cheap, limitless access has created a mad rush to leverage this newfound communication type. Unfortunately, these predominant characteristics of VPNs have overshadowed fundamental security flaws that seem to remain obscure and hidden from the sales glossies and product presentations. This chapter is dedicated to shedding light on the security risks associated with VPNs and the misunderstanding that VPNs are synonymous with security.

It is crucial that the reader understands the security limitations detailed herein have almost nothing to do with VPN technology itself. There are several types of VPN technologies available — for example, IPSec, SSL, and PPTP, to mention a few — and each has advantages and disadvantages depending on the requirements and implementation. In addition, each has various levels of security that can be leveraged to accommodate a mixture of conditions. The insecurity of VPNs as a medium and a process is being discussed, and not the technical aspects or standards.

What is being addressed is the evaluation of VPNs by the general consumer arrived at from the sales paraphernalia flooding the market and the industry's products claiming to fill consumers' needs. Unfortunately, the demand is overwhelming, and the development of sufficient controls that could be integrated to increase the security lags behind what is being currently experienced. The word "security" appears frequently when VPNs are being discussed, which typically applies when defining the VPN itself — the protection of data in transit. Unfortunately, the communication's security stops at the termination point of the VPN, a point where security is paramount.

The goal of this chapter is to introduce VPNs, and explain their recent surge in popularity as well as the link to current advances in Internet connectivity, such as broadband. Then, the security experienced with legacy remote access solutions is compared with the realized security the industry has more recently adopted. This is an opportunity to look beyond the obvious and discuss the huge impact this technology is having on the total security posture of organizations. The problem is so enormous that it is difficult to comprehend — a "can't see the forest for the trees" syndrome.

One Thing Leads to Another

The popularity of VPNs seems to have blossomed overnight. The ability to remove the responsibility of maintaining a dedicated line for each contiguous remote user at the corporate site and leverage the existing

Internet connection to multiplex a greater number of connections previously unobtainable has catapulted VPN technology.

As with many technological combinations, one type may tend to feed from another and reap the benefits of its companion's advances. These can materialize as improvements or options in the technologies and the merger of implementation concepts — a marriage of symbiotic utilities that culminate to equal more than attainable alone. Cell phones are an example of this phenomenon. Cell phones support digital certificates, encryption, e-mail, and browsing, among other combinations and improvements. The wireless community has leveraged technologies normally seen in networking that are now gaining attention from their use in another environment. Cell phone use is more robust and the technology used is employed in ways not originally considered. It is typically a win–win situation.

The recent universal embracement of VPNs can be attributed to two primary changes in the communication industry: global adoption of Internet connectivity, and inexpensive broadband Internet access. These contemporary transformations and the ever-present need to support an increasing roaming user community have propelled VPN technologies to the forefront of popularity.

Roaming Users

Roaming is characterized by the natural progression from early networks providing services to a captive population and allowing those same services to be accessible from outside the normal boundaries of the network. Seemingly overnight, providing remote access to users was paramount and enormous resources were allocated to providing it.

Initially, as shown in Exhibit 43.1, modems were collected and connected into a common device that provided access to the internal network, and, of course, the modems were connected to phone lines that ultimately provided the access. As application requirements grew exponentially, the transmission speed of modems increased modestly and change was on the horizon. The first wave of change came in the form of remote desktops, or in some cases, entire systems. As detailed in Exhibit 43.2, a user would dial in and connect

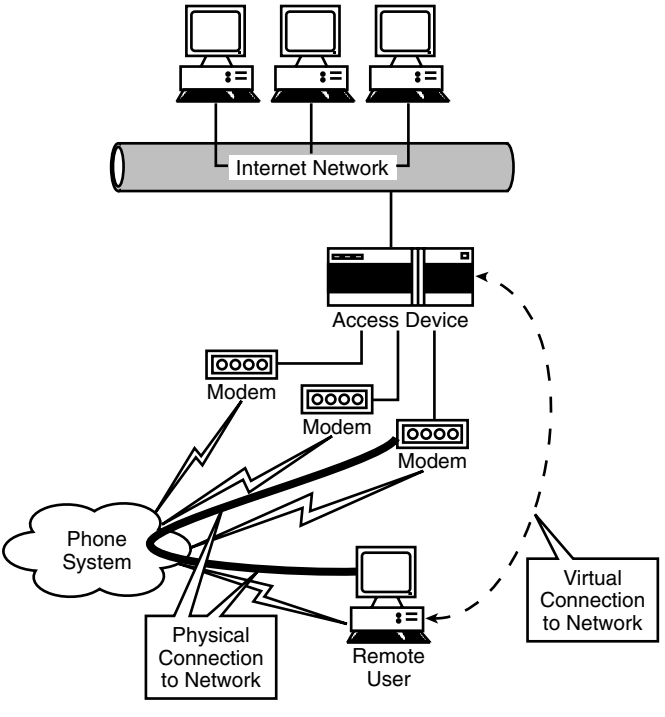


EXHIBIT 43.1 Standard remote access via modems.

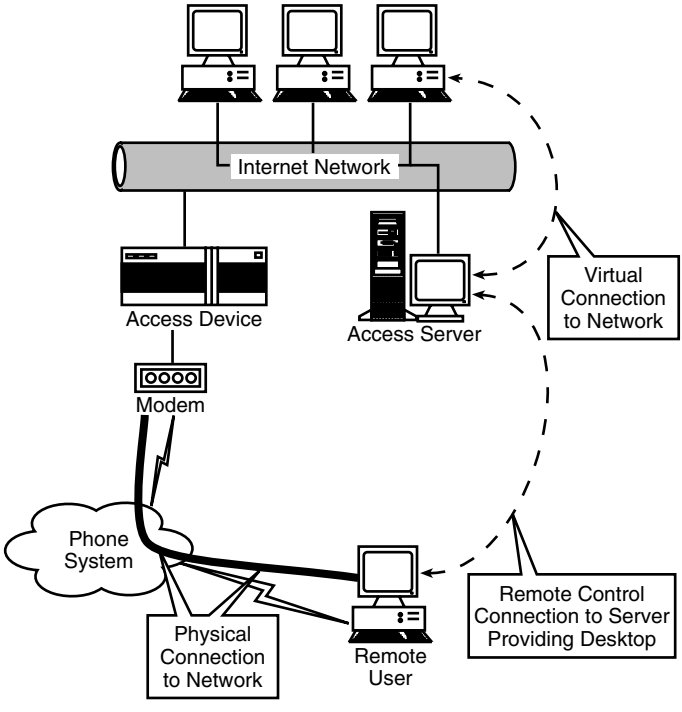


EXHIBIT 43.2 Standard remote access via modems using remote control or remote desktop.

to a system that could be either remotely controlled or export the desktop environment to the remote user. In both cases, the bandwidth required between the remote user and the core system was actually reduced and the functionality was amplified. Cubix, Citrix, and PC Anywhere became the dominant players in providing the increased capabilities, each with its own requirements, advantages, and cost.

Performance was realized by the fact that the remote access server on the internal network had very high access speeds to the other network resources. By using a lightweight protocol to control the access server, or to obtain desktop imagery, the modem connection had virtually the same feel as if on the actual network. It was at this point in the progression of remote access that having the same look and feel of the internal network had become the gauge to which all remote access solutions would be measured. From this point forward, any differences or added inconveniences would diminish the acceptance of a remote access solution.

Internet Adoption

The Internet's growth has been phenomenal. From the number of people taking their first steps on the Net, to the leaps in communication technologies, Internet utilization has become increasingly dense and more populated. The Internet has become a requirement for business and personal communications rather than a novelty or for simple amusement. Businesses that were not associated in some way with the Internet are now attempting to leverage it for expansion and increase client satisfaction while reducing costs. It is not uncommon for an organization to include an Internet connection for a new or existing office as a default install.

In contrast, early adopters of dedicated Internet connections, as a rule, had a single access point for the entire organization. As depicted in Exhibit 43.3, remote offices could get access by traversing the wide area network (WAN) to the central location at which the Internet was accessible. This very common design scenario was satisfactory when Internet traffic and requirements were limited in scope and frequency. As the requirements for Internet access grew, the number of connections grew in direct proportion, until the WAN began to suffer. Shortly thereafter, as the costs for direct connectivity declined and the Internet became more and more a part of business life, it became an essential tool and greater access was needed.

Presently, the Internet has become an indispensable utility for successful businesses, and the volume of Internet traffic coursing through internal networks is astounding. The need for information now greatly

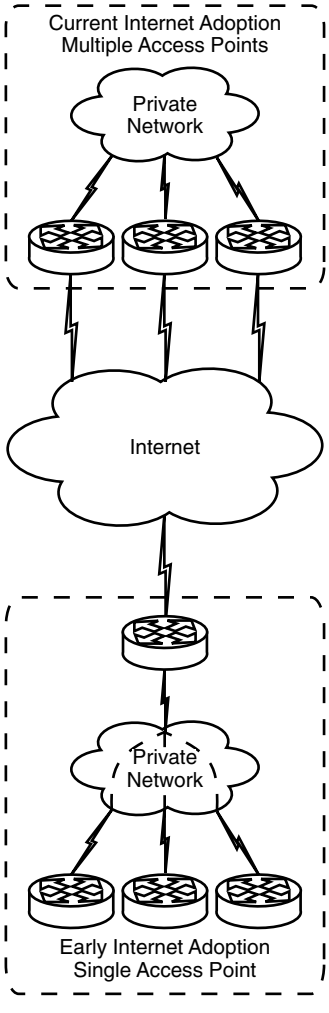


EXHIBIT 43.3 Internet access through one central point compared to the several typically seen now.

outweighs the cost of Internet connectivity. In the past, Internet connections had to be validated and carefully considered prior to implementation. Today, the first question is typically, "How big a pipe do we need?" not "Where should we put it?"

The vast adoption of the Internet and acceptance of it as a fundamental requirement has resulted in the increased density and diversity of the Internet. Today, organizations have several access points and leverage them to reduce load on other internal networks and provide increased performance for internal users as well as providing service redundancy. By leveraging the numerous existing connections, an organization can implement VPN technology to enhance communication, while using a service that was cost-justified long before the inclusion of VPNs.

Broadband

Before the existence of high-speed access to the Internet that is standard today, there were typically only modems and phone lines that provided painfully slow access. There were, of course, the few privileged users who had ISDN available to them that provided some relief. However, access was still based on modems and could be a nightmare to get to work properly. The early adopters of remote access used modems to obtain data or services. As the Internet became popular, modems were used to connect to an Internet service provider (ISP) that

provided the means for accessing the Internet. In either case, the limited speed capabilities were a troubling constant.

Today's personal and home access to the Internet can reach speeds historically realized only with expensive lines that only the largest companies could afford or obtain. At present, a simple device can be installed that provides a connection to the ISP and leverages Ethernet to connect to the host PC in the home or small office. Today, access is provided and controlled separately from the PC and rarely requires user intervention. The physical connection and communication medium are transparent to the user environment. Typically, the user turns on the computer and the Internet is immediately available. This is in stark contrast to the physical connection associated with the user's system and the modem, each working together to become the signal termination point and assuming all the responsibilities that are associated with providing the connection.

As with many communication technologies (especially with regard to modem-based remote access), a termination point must be supplied to provide the connection to the remote devices or modems. With dial-up solutions, a modem (virtual or physical) is supplied for the remote system to dial into and establish communications. A similar requirement exists for broadband, whether for cable modems or xDSL technologies: a termination point must be supplied to create a connection for the remote devices at the home or office.

The termination point at the core — with regard to the adoption of VPNs — has become one of the differentiating factors between broadband and modems. To provide remote dial-up access for employees, a single modem could be installed in a server — or workstation for that matter — and a phone line attached. The remote user could be supplied with a modem, the phone number, and the use of some basic software; a connection could be established to provide ample access to the system and services.

In contrast, broadband implementations are more complicated and considerably more expensive; thus, today, only service providers implement this type of technology. An example is Internet cable service; not too many companies have access to the cable infrastructure to build their own internal remote access solution. Currently, broadband is not being used for point-to-point remote access solutions. Therein lies the fundamental appeal of VPNs: a way to leverage this advanced communication technology to access company resources.

Not only is the huge increase in speed attractive because some of the application requirements may be too great for the limited bandwidth provided by modems, but the separation of the technology from the computer allows for a simplified and scalable integration. Under these circumstances, broadband is extremely attractive for accessing corporate resources. It is one thing to have broadband for high-speed Internet browsing and personal excursions, but it is another to have those same capabilities for business purposes. Unfortunately, as described earlier, broadband technologies are complex and infeasible for a nonservice provider organization to implement for internal use. The result is a high-speed communication solution that currently only provides Internet access — that is, until that advent of VPNs.

Extended Access

As communication capabilities increased and companies continued to integrate Internet activities into everyday procedures, the creation of VPN technology to merge the two was critical. Dial-up access to the Internet and broadband provide access to the Internet from nearly anywhere and with high speeds. Both allow global access to the Internet, but there is no feasible or cost-effective way to terminate the connection to the company headquarters. Since broadband access was intimately associated with the Internet and direct-dial solutions were ineffective and expensive, the only foreseeable solution was to leverage the Internet to provide private communications. This ultimately allowed organizations to utilize their existing investment in Internet connectivity to multiplex remote connections. The final hurdle was to afford security to the communication in the form of confidentiality, information integrity, access control, authentication, auditing, and, in some cases, nonrepudiation.

The global adoption of the Internet, its availability, and the increased speeds available have exceeded the limitless access enjoyed with dial-up. With dial-up, the telephone system was used for establishing communications — and telephones are everywhere. The serial communication itself was carried over a dedicated circuit that would be difficult to intercept for the everyday hacker and therefore relatively secure. Now that the Internet is everywhere, it can be used to duplicate the availability that exists with the telephone network while taking advantage of the increased speeds. Granted, if a modem is used to connect to the Internet, the speed is not realized and the phone system is being used to connect, but locally; the Internet is still being used for the common connection medium. Even with dial-up remote access, this was a huge leap in service because

many corporate-provided remote access solutions could be difficult to connect to from overseas. If not restricted by policy, cost became an issue because phone equipment and systems were not of the quality they are today, and long-distance transmissions would hinder the connection. In contrast, there are tens of thousands of ISPs worldwide that can provide access to the Internet, not including the very large ISPs that provide phone numbers globally. Finally, in addition to the seemingly endless supply of access points, there are companies that act as a central point for billing and management for hundreds of ISPs worldwide. From the point of view of the user, there is one large ISP everywhere on the globe.

The final hurdle was to provide the communication protection from in-transit influence or exposure as had occurred with old remote access over the phone network. VPN technology was immediately used to fill this gap. With the advent of expanded communication capabilities and the availability of the Internet, the ever-expanding corporate existence could be easily supported and protected during transit.

Connected All the Time

In the past, a remote user could dial into a modem bank at headquarters and access services remotely with little concern for eavesdropping, transmission interception, or impersonation. From the perspective of the hosting site, layers of security could be implemented to reduce exposure. Authentication, dial-back, time limitations, and access restrictions were employed to increase control over the communication and decrease exposure to threats. These protection suites were made possible primarily because of the one-on-one aspect of the communication; once the connection was established, it could be easily identified and controlled. As far as the communication itself, it was relatively protected while traversing the public phone system over dedicated circuits.

Because broadband technology can utilize Ethernet to allow connectivity to the access device, the computer simply has to be "on" for Internet communications (see Exhibit 43.4). This represents a huge change from traditional modem access, where the computer was responsible for establishing and maintaining the connection. Currently, with typical broadband the connection is sustained at the access device, allowing Internet connectivity, regardless of the state of other systems on the Ethernet interface. The Ethernet interface on the computer does not require a user to initialize it, know a phone number, or be concerned about the connection. All these options are controlled by the operating system; even the IP address is automatically assigned by the ISP, reducing the interaction with the user even further. Now the responsibility for Internet connectivity rests

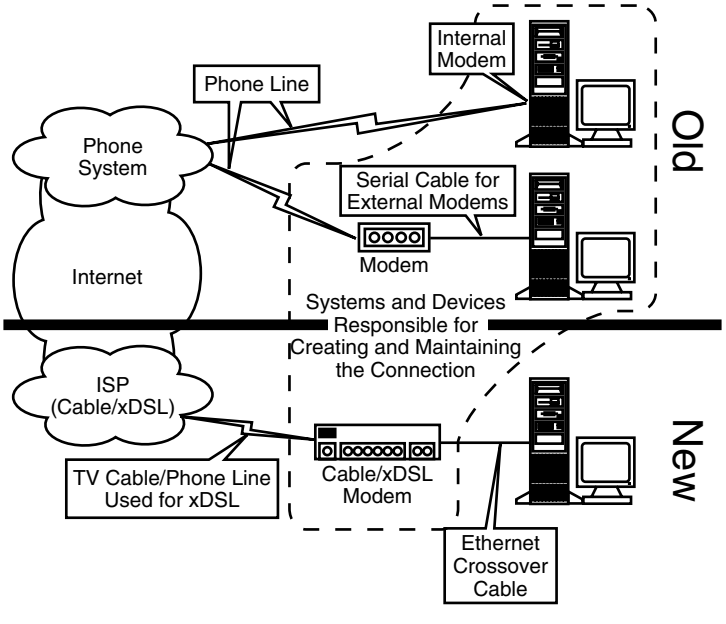


EXHIBIT 43.4 Broadband removed the user and system from the establishment of the connection.

solely on the access device, freeing the user and the user's computer from the need to maintain the connection. The end system is simply a node on a network.

Computers that are connected to the access device are connected to the Internet with little or no protection. It is very common for a broadband provider to install the cable or line and an Ethernet interface in the computer and directly connect the system with no security modifications. This results in basic end-systems with no security control being connected directly to the Internet for extended periods of time. The difference is tremendous. Instead of a fleeting instance of a roaming user on the Internet dialing up an ISP, the IP address, type of traffic, and even the location of the computer are exposed to the Internet for extended periods of time. When compared with the direct remote user dial-up support for corporations, the exposure is staggering. The obvious difference is that the user is connected to the Internet whereas the dial-up service provided by the company was point-to-point.

It is widely accepted that when a system is connected to the Internet, regardless of type, it is exposed to a colossal number of threats. It is also accepted that the greater the length of continuous time the connection is established, the greater the exposure or the risk of being found and targeted. Firewalls are usually placed on networks that have dedicated Internet connections, but they are not usually seen on hosts that have intermittent connections to the Internet. One of the reasons can be the nature of the connection — it is much more difficult to hit a moving target. But the reality is that this can be misleading, and roaming systems can be accosted in the same way as a system with a dedicated connection. In short, dial-up access to the Internet exposes the system to threats, and dedicated connections are exposed to the same threats as well, but with increased risk that can typically be attributed to duration. Whether connected all the time or some of the time, by broadband or modem, if you are on the Internet you are exposed to attack; it just so happens that when connected all the time, you are a sitting duck, not a flying one.

Accessing Corporate Networks

VPN technology is the final catalyst for allowing remote users to gain access to corporate resources by utilizing the Internet. This was a natural progression; the Internet is everywhere. Like the phone system, the higher bandwidth connections are becoming the norm, and VPN technology is securing the transmission with encryption techniques and authentication.

Much of VPN's success has been attributed to the advent and availability of broadband technologies, because high-speed access was great for browsing and getting bigger things off the Internet faster, but that is about all. Almost overnight the bandwidth typically associated with personal access, such as 32K or even 56K modems, to the Internet was increased 100 times. The greater access speeds attained by moving away from the public phone system and modems to dedicated broadband connectivity were quickly followed by rash of excitement; however, at the same time, many wanted the service to access corporate resources. As the excitement wore off from the huge leap in access speeds, many turned their eyes on ways to use this for remote access. It is at this point that VPN technology took off and absorbed the technical community.

Remote client software was the first on the scene. A product package included a device that was connected to the Internet at the corporate site and the client software that was loaded on the roaming system, resulting in remote access to corporate resources over the Internet. A great deal of time and money was invested in remote access solutions, and that continues today. In concert with remote client-based access, the rush to VPNs was joined by DSL and cable modem replacements that provided the VPN termination, once again relieving the client system from the responsibility of the communication. VPNs are now a wildfire being pushed across the technical landscape by a gale-force wind of broadband access.

Once unbridled access to the corporate network was available, it was not uncommon for remote sites or users to copy or open data normally maintained under the protection of elaborate firewalls and other protection suites provided at the corporate site. For many implementations, VPNs are used to run applications that would normally not be available on remote systems or require expensive resources and support to provide to employees at remote offices. In short, VPNs are being used for nearly everything that is typically available to a system residing on the internal network. This is to be expected, considering that vendors are selling the technology to do just that — operate as if on the internal network. Some solutions even incorporate Microsoft's Windows Internet Naming Service (WINS) and NetBIOS capabilities into their products to allow Domain browsing for systems and resources as if at the corporate site.

In essence, VPNs are being implemented as the panacea to integrate remote activities into internal operations as seamlessly as possible. The end product is data and applications being run from systems well outside the confines of a controlled environment.

Open Ended

Fundamentally, the service afforded by a VPN is quite simple: protect the information in transit, period. In doing so, various communications perks can be realized. A good example is *tunneling*. To accommodate protected communications as seamlessly as possible, the original data stream is encapsulated and then transmitted. The encapsulation procedure simplifies the protection process and transmittal of the datagram. The advantage that arises is that the systems in the VPN communicate as if there were no intermediary. An example, shown in Exhibit 43.5, is a remote system that creates a datagram that would operate normally on the internal network; instead, it is encapsulated and forwarded over the Internet to a system at the corporate office that de-encapsulates (and decrypts, if necessary) the original datagram and releases it onto the internal network. The applications and end-systems involved are typically never the wiser.

The goal for some VPN implementations is to provide communications for remote users over the Internet that emulates intranet services as closely as possible. Many VPN solutions are critiqued based on their capabilities to allow services to the client systems that are usually only available internally. With the adoption of broadband Internet access there is less stress on pure utilitarian aspects normally seen with dial-up solutions, where various limitations are assumed because of the limited bandwidth. To allow for the expanded communication requirements, many VPN solutions integrate into the environment in a manner that remains transparent not only to the user, but also to the applications that utilized the connection. Therefore, the protection realized by the VPN is extended only to the actual transport of data — exactly its purpose.

For the most part, prior to encapsulation or encryption, anything goes, and the VPN simply protects the transmission. The connection is protected but that does not equate to the communication being protected. To detail further, systems on internal networks are considered a community with common goals that are protected from the Internet by firewalls and other protection measures. Within the trusted community, data flows openly between systems, applications, and users; a VPN simply augments the process and protects it during transmission over the Internet. The process is seamless and transparent, and it accommodates the traffic and application needs. The result is that data is being shared and utilized by shadowy internal representations of the remote systems.

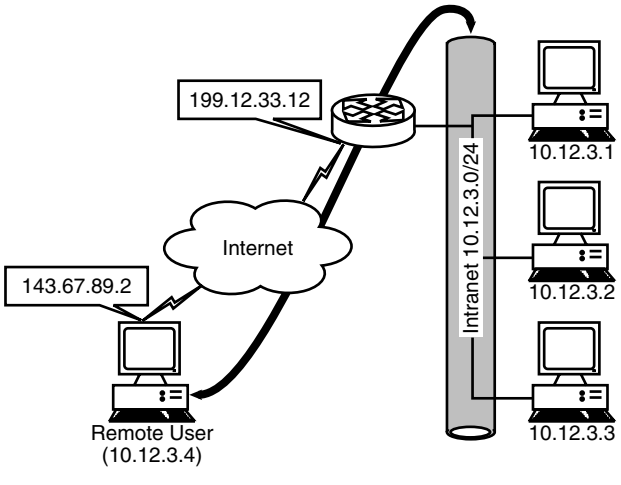


EXHIBIT 43.5 Attacker must attempt access to corporate data directly, the most difficult path.

Access Points

Having internal services wholly available to systems residing on internal networks is expected. The internal network is typically a controlled, protected, and monitored environment with security policies and procedures in place. As services and data are accessed internally, the exposure or threat to that communication is somewhat known and accepted at some level. Most organizations are aware of security threats on internal networks, but have assumed a level of risk directly proportional to the value or impact of loss if they were to be attacked. Much of this is attributed to simple population control; they assume greater risk to internal resources because there are fewer people internally than on the Internet, interaction is usually required (hence, a network), and each system can be monitored if desired. Basically, while some statistics tell us that internal networks are a growing source of attacks on corporate data, organizations feel confident that they can control what lies within their walls. Even organizations that do not have security policies and may consider themselves vulnerable will always assume that there is room to grow and implement security measures as they see fit. Nevertheless, the Internet represents a much greater threat in the eyes of many organizations, and this may be a reality for some organizations; each is different. The fundamental point is that the Internet is an unknown and will always be a threat, whereas certain measures can be taken — or the risk can be accepted — more readily on an internal network. In any case, internal networks are used to share information and collaborate to support or grow a business, and it is that open interaction people want from home over the Internet.

VPN technology is a total contradiction of the assumed posture and reach of control. The internal network, where applications, services, and data reside, is considered safe by virtue of firewalls, procedures, and processes overseen by administrators focused on maintaining security in some form or another. However, the nature of VPN negates the basic postulation of corporate security and the understood security attitude. Attackers who may have been thwarted by hardened corporate firewalls may find remote VPN clients much easier targets that may provide the same results.

On the whole, administrators are constantly applying security patches, updating processes, and performing general security maintenance on critical systems to protect them from vulnerabilities. Meanwhile, these vulnerabilities remain on end-user systems, whose users are much less likely to maintain their systems with the same integrity. In the event that an advanced user were to introduce a comprehensive protection plan, many remote systems do not run enterprise-class operating systems and are inherently insecure. Microsoft's Windows 95 and 98 platforms are currently installed on the majority of personal or end-user class systems and are well-known for limited security capabilities and overall robustness. Therefore, fundamental flaws weaken any applied security in the system.

The collision of the attributes that contribute to a common VPN implementation result in the cancellation of applied security infrastructure at the corporate site. Nearly every aspect of Internet-facing protection is invalidated the minute a user connects to corporate with a VPN. A single point of protection applies only if the protected network does not interact with the volatile environment being evaded.

Envelope of Security

To fully grasp this immense exposure, envision a corporate network segmented from the Internet by an arsenal of firewalls and intrusion detection systems, and even suppose that armed guards protect the building housing a private community of systems. Assume that the data on the network is shared and accessed in the open while on the internal network. Each system participating is protected and controlled equally by the establishment.

Now, take one of the systems to an uncontrolled remote location and build a point-to-point connection with modems. The remote computer is still isolated and not connected to any untrusted systems other than the phone system. The communication itself is relatively anonymous and its interception would be complicated, if discovered. However, as we see in VPNs, encryption can be applied to the protocol over the phone system for added protection.

Next, take the same system at the remote location and connect it to the Internet and establish a VPN to the corporate network. Now the system is exposed to influences well beyond the control realized when the computer was at the corporate office; still, the same access is being permitted.

In the three foregoing examples, degradation in security occurs as the computer is emoved from a controlled environment to a remote location and dial-up access is provided. The risks range from the system being stolen to the remote chance of the transmission being captured while communicating over the telephone network, but the overall security of the system and the information remain relatively protected. However, when the remote computer is placed on the Internet, the exposure to threats and the risk of operation are increased exponentially.

In the beginning of the example, the systems reside in an envelope of protection, isolated from unauthorized influences by layers of protection. Next, we stretch the envelope of protection out to the remote dial-in system; understandably, the envelope is weakened, but it certainly exists in nature to keep the information sheltered. The remote dial-in system loses some of the protection supplied by the fortified environment of corporate and is exposed to finite set of threats, but what is more important is that the envelope of security for the corporate site had not been dramatically affected.

In reality, the added risks of allowing remote systems to dial in directly are typically associated with unauthorized access, usually gained through the phone system. Corporate provides phone numbers to remote users to gain access and those same numbers are accessible from anywhere on the planet. Attackers can easily and quickly determine phone number ranges that have a high probability of including the target remote access numbers. Once the range is known, a phone-sweeping or "war-dialer" program can be employed to test each number with little or no intervention from the attacker. However, there are many factors that still manage to keep these risks in check. Dial-back, advanced and multi-layered authentication, extensive logging, time constraints, and access constraints can combine to make a formidable target for the attacker. With only a single point of access and the remote system in isolation, the security envelope remains intact and tangible. The degree of decay, of course, is directly related to the security of the single point of access at corporate and the level of isolation of the remote system.

In the last scenario, where the employment of a VPN provides corporate connectivity over the Internet, the security is perceived to be very high, if not greater than or equal to dial-up access solutions. Why not? They appear to have the same attributes and arguably the same security. In dial-up solutions, the communication is relatively protected, the system providing termination at corporate can be secured, and authentication measures can be put in place to reduce unauthorized access. VPNs, too, have these attributes and can be exercised to acquire an inclusive security envelope.

Unfortunately, the VPN offers a transparent envelope, a security façade that would not normally exist at such intensity if VPNs were not so accomplished as a protocol. The corporate-provided envelope is stretched to a breaking point with VPNs by the sheer fact that the remote system has gained control of the aspect of security and the employment of protection. It will become very clear that the envelope of security is no longer granted or managed by corporate but rather the remote system is now the overseer of all security — locally and into corporate.

A remote system connects to the Internet and obtains an IP address from the ISP to allow communication with the rest of the Internet community. Somewhere on the Internet is a VPN gateway on the corporate network that is providing access to the internal network. As the remote system establishes the VPN to share data, a host of vulnerabilities are introduced that can completely circumvent any security measures taken by corporate that would normally be providing the security envelope. It is at the point of connecting to the Internet where the dramatic tumbling of realized security takes place, and the remote system becomes the judge, jury, and possibly the executioner of corporate security.

The remote system may have employed a very robust VPN solution, one that does not allow the host system to act as a router or allow the forwarding of information from the Internet into the private network. To take it one step further, the VPN solution may employ limited firewalling capabilities or filtering concepts to limit access into the internal network. Nonetheless, the protection possibly supplied by the VPN client or firewall software can be turned off by users, ultimately opening them up to attack. In the event that a package can be implemented in which the user cannot turn off the protection suite, it can be assumed that a vulnerability will arise that requires a patch to remedy.

This scenario is extremely common and nearly an everyday occurrence for firewall and perimeter security administrators simply attempting to keep up with a limited number of firewalls. Given the lack of attention normally seen in many organizations toward their firewall maintenance, one can only imagine the disintegration of security when vulnerabilities are discovered in the remote system's firewall software.

Vulnerability Concepts

To fully understand the extremity of the destruction of perceived corporate security made available by ample amounts of technology and processes, it is necessary to know that the remote system is open and exposed to the Internet. In some cases, as with broadband, the exposure is constant and for long periods of time, making it predictable — an attacker's greatest asset.

The Internet is a sea of threats, if nothing else, simply because of the vast numbers of people and technologies available to them to anonymously wreak havoc on others, especially those unprepared. There are several different types of attacks that are for different uses and affect different layers in the communication. For example, denial-of-service (DoS) attacks are simply geared to eliminate the availability of a system or service — a purely destructive purpose. DoS attacks take advantage of weaknesses in low-level communication attributes, such as a protocol vulnerability, or higher-level weaknesses that may reside in the application itself. Some other attacks have very specific applications and are designed for particular situations to either gain access or obtain information. It is becoming more and more common to see these attacks taking advantage of application errors and quirks. The results are applications specifically engineered to obtain system information, or even to remotely control the host system.

Trojans have become very sophisticated and easy to use, mostly because of huge weaknesses in popular operating systems and very resourceful programmers. A typical system sitting on the Internet can have a Trojan installed that cannot only be used to gain access to the system, remotely control portions of the host system, obtain data stored locally, and collect keyboard input, but can notify the attacker when the host system is online and ready for access. In some cases, information can be collected offline and sent to the attacker when the Internet connection is reestablished by the victim. It is this vulnerability that represents the worst-case scenario and, unfortunately, it is commonplace for a typical home system to be affected.

In a case where the Trojan cannot be installed or implemented fully, an attacker could gain enough access, even if temporarily, to collect vital information about the targeted system or user, ultimately leading to more attacks with greater results. It can be argued that anti-virus programs and host-based firewall applications can assist the user in reducing the vulnerabilities and helping in discovering them — and possibly eradicating them. Unfortunately, the implementation, maintenance, and daily secure operation of such applications rests in the hands of the user. Nevertheless, it is complicated enough protecting refined, highly technical environments with dedicated personnel, much less remote systems spread all over the Internet.

A Step Back

Early in the adoption of the Internet, systems were attacked, sometimes resulting in unauthorized access and the loss of data or the disclosure of proprietary information. As the threats became greater, increasingly more sophisticated, and difficult to stop, firewalls were implemented to reduce the direct exposure to the attack. In combination, systems that were allowing certain services were hardened against known weaknesses to further the overall protection. Furthermore, these hardened specific systems were placed on isolated networks, referred to as DMZs, to protect the internal network from attacks launched from them or weaknesses in their implementation. With all these measures in place, hackers to this day continue to gain astounding access to internal systems.

Today, a firewall is a fundamental fixture in any Internet facing connection, and sometimes in huge amounts protecting vast numbers of systems and networks. It has become the norm, an accepted fact of Internet life, and an expensive one as well. Protecting the internal systems and resources from the Internet is paramount, and enormous work and finances are usually dedicated to supporting and maintaining the perimeter.

It is reasonable to state that much of the protection implemented is to protect proprietary data or information from dissemination, modification, or destruction. The data in question remains within the security envelope created by the security measures. Therefore, to get to the information, an attacker would have to penetrate, circumvent, or otherwise manipulate operational conditions to obtain the data or the means to access it more directly (see Exhibit 43.6).

With the advent of VPNs, the remote system is permitted a protected connection with the corporate data, inside the enclave of known risks and threats. It is assumed that the VPN protects the communication and

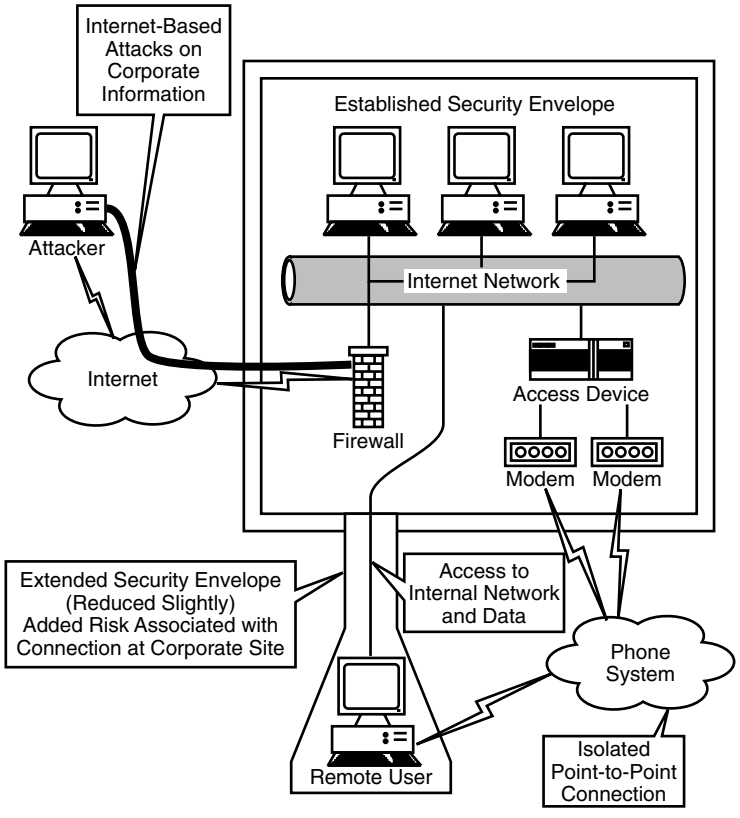


EXHIBIT 43.6 Attacker must attempt access to corporate data directly, the most difficult path.

stretches the security outward from the corporate to the remote location. Unfortunately, this assumption has overlooked an essential component of VPNs — the Internet. Now, as shown in Exhibit 43.7, an attacker can access corporate data on a system completely exposed and in control of a common user — not under the protection of technology or experience found at the corporate site.

From the point of view of the attacker, the information is simply on the Internet, as is the corporate connection; therefore, the access process and medium have not changed, just the level of security. The result is that the information is presented to the attacker, and direct access through a much more complicated path is not required. If it were not for the Internet connection, the remote hosts would have increased functionality, speed, and protection compared with legacy remote access with modems. Regrettably, the Internet is the link to the extended functionality as well as the link to ultimate insecurity.

Logically, this is a disaster for information security. We have invested monumental amounts of time, research, and money into the evolution of security and the mitigation of risk associated with connecting to a global, unrestricted network. We have built massive walls of security with bricks of technology ranging from basic router filtering, firewalls, and intrusion detection systems to system hardening, DMZs, and air-gaps. Now that we have a plethora of defense mechanisms pointed at the Internet, we are implementing an alternative route for attackers, leading them away from the traps and triggers and pointing them to our weakest points.

The concept of alternative forms and directions of attack when faced with considerable fortifications can be likened to medieval warfare. Castles were constructed with enormous walls to thwart intruders. Moats were filled, traps were laid, and deadly focal points were engineered to halt an attack. In some of these walls, typically under the surface of the moat, a secret gateway was placed that allowed scouts and spies out of the castle to collect information or even supplies to survive the siege. It is this reality that has repeated itself — a gateway placed facing the world to allow allies access into the stronghold. The differentiating factor between what is being seen now and ancient warfare is that long ago the kingdom would not permit a general, advisor, or any person outside the walls that could have information valuable to the enemy.

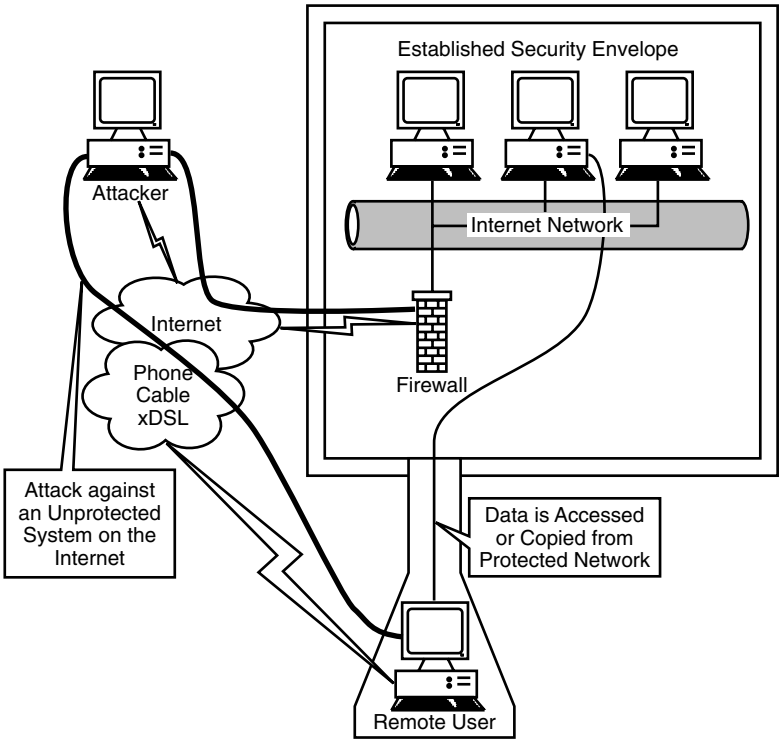


EXHIBIT 43.7 Attacker obtains data from a much less protected point on the Internet.

In stark contrast, today people from every level in the corporate chain access information outside the protected space. This is equivalent to sending a general with attack plans through the gateway, out of the castle, so he can work on the plan in his tent — presumably unprotected. It does not take much effort for an attacker to pounce on the general and collect the information that would normally require accessing the castle directly. In reality, a modern-day attacker would have so much control over the victim that data could be easily modified or collected in a manner that would render the owners oblivious to their activities. Exhibit 43.8 clearly depicts the evolution of the path of least resistance.

Disappointingly, the complicated labyrinthine safeguards we have constructed are squarely pointed at the enemy; meanwhile we are allowing the information out into the wild. The result is that the finely honed and tuned wall of protection is reduced to almost nothing. Where a small set of firewalls protected information on internal networks at a single entry point, there now exist thousands of access points with no firewalls. Not only have we taken a step back but also the problem reduced by firewalls has increased in scale. Early in Internet adoption a single Internet connection with a firewall would suffice. Today, organizations have several Internet connections with complicated protection measures. With the addition of VPNs for remote systems and small home offices, organizations have thousands of Internet connections beyond reasonable control.

Case in Point

Late one Friday, I received a phone call from a friend who worked for a large national construction company as a chief engineer. Calls from him were typical when his computer was acting up or a fishing trip was being planned for the weekend. However, this call started very unusually. He stated that he thought he had been hacked — his hard drive runs late into the night and the recently loaded BlackIce was logging a great deal of unknown traffic. I knew he used a cable modem and a VPN to work from home, either at night or during the day, to avoid traffic and general office interruptions. I was also aware that he used Windows 98 as an operating system and standard programs to complete his work. Additionally, he left his computer on all the time — why not?

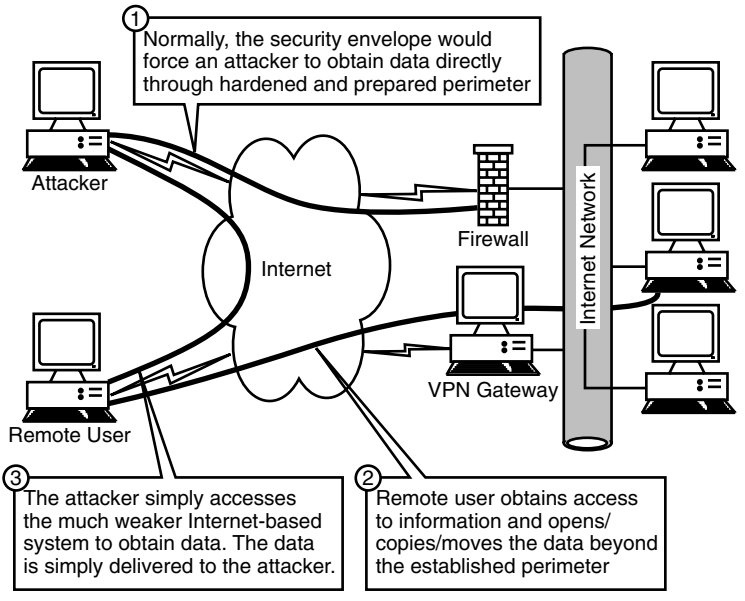


EXHIBIT 43.8 Data is accessed by a system exposed to vulnerabilities and various risks associated with the Internet.

Completely convinced that he had been attacked, I told him not to touch the computer and to start a sniffer using another computer on his home network to see what was going over the wire. In a few minutes, communications were started between his computer and an Internet-based host. It was clear, after looking at the traffic more clearly, that his system was being accessed. Between his experiences, log files from various software he had installed on the system, and previous experiences with other friends in his shoes, I assumed that his system was accessed. I had him unplug the Ethernet from the cable modem and asked how serious could the issue be — in other words, what was on the box that someone would want or appreciate getting.

After a short discussion, it appeared that the hacker was accessing all the bid packages for building projects all over the United States, each encrusted with logos, names, contact information, competition analysis, schedules, and cost projections. It was my friend's job to collect this information and review it for quality control and engineering issues. Further discussions proved that he knew when he last accessed the data based on work habits and general memory. It was at this point that he told me this had been going on for some time and he just got around to calling me. He wanted to try anti-virus programs and freeware first so that he would not bother me with a false alarm. Subsequently, we collectively decided to access the system to try to determine what was accessed and when.

The first thing we found was BackOrifice with basic plug-ins, which led me to believe that this may not have been intentionally directed at him, but rather someone wanting to play with a wide-open Windows system sitting on the Internet. We started checking files for access times; many were accessed in the middle of the night several weeks prior. More investigation turned up hidden directories and questionable e-mails he had received sometime before. At this point, I simply stopped and told him to assume the worst and try to think of anything else that may have been on his system. It turned out that a backup of his TurboTax database — not password protected — was on the system along with approved human resource documents for employees in his department who had recently received a raise.

The entire phone conversation lasted about three hours — that's all it took. I suspect that the call to his manager was much more painful and felt much longer. But was it his fault? His company provided him the Internet connection and the VPN software, and access from home was encouraged. It seemed logical to him and his manager. He needed access to the Internet for research, and he typically got more done at home than at the office. However, an unknown assailant on the Internet, who could be either a hired gun to get the information or a script-kiddie that stumbled into a pot of gold, accessed extremely sensitive information. In either case, it was out there and could have an impact on the business for years.

Solutions

There is, of course, no easy solution to the security dilemma that is presented by the implementation of VPNs. Even with sophisticated technology, organizations still cannot stop hackers. They continue to access systems in heavily protected networks with apparent ease. Much of this can be attributed to poor design, gaps in maintenance, improper configuration, or simple ignorance. In any case, with focused attention on the perimeter, unauthorized access is still happening at an alarming rate. Given this scenario of hundreds if not thousands of remote computers on the Internet, what can be done to protect them? Simply stated, if an internal network cannot be protected when the best efforts are thrown at the problem, there is little hope of protecting the masses at home and on the road.

As with any sound security practice, a security policy is crucial to the protection of information. Specifying data access limitations and operating parameters for information exchange can greatly reduce the exposure of information. In other words, if a certain type of information is not needed for remote work, then remote access systems should not provide access to that information or system. By simply reducing the breadth of access provided by the remote access solution, data can be inherently protected. The practice of limiting what is actually accessible by remote users has materialized in the form of firewalls behind VPN devices seemingly protecting the internal network from the VPN community. Unfortunately, this design has enormous limitations and can limit the scalability of the VPN in terms of flexibility of access. Another eventuality is the inclusion of filtering methods employed in the VPN access device. Filters can be created to control traffic that is injected into the internal network, and in some cases filters can be associated with actual authenticated users or groups.

No matter how access is restricted, at some point a remote user will require sensitive information and anyone implementing services for users has been faced with that "special case." Therefore, technology must take over to protect information. Just as we look to firewalls to protect our internal networks from the Internet, we must look to technology again to protect remote systems from relaying proprietary information into the unknown. The application of host-based protection software is not entirely new, but the growing number of attacks on personal systems has raised awareness of their existence. However, these applications are point solutions and not a solution that is scalable, flexible, or centrally controlled or managed to maintain security. In essence, each user is responsible for his or her realized security posture.

Conclusion

VPNs can be enormously valuable; they can save time, money, expand access, and allow organizations ultimate flexibility in communications. However, the private link supplied by a VPN can open a virtual backdoor to attackers. Organizations that permit sensitive data to traverse a VPN potentially expose that information to a plethora of threats that do not exist on the protected internal network.

There are many types of VPN products available, all with their own methods of establishing the connection, maintaining connectivity, and providing services usually found on the internal network. Unfortunately, if the remote system is not involved in dedicated communications with the central office via the VPN, the system can be considered extremely vulnerable.

The Internet has grown to permeate our lives and daily activities, but there has always been a line drawn in the sand by which separation from total assimilation can be measured. Firewalls, modems, routers, filters, and even software such as browsers can provide a visible point of access to the Internet. As technology becomes more prevalent, the demarcation between the Internet and private networks will begin to blur. Unfortunately, without proper foresight, the allocation of security measures and mitigation processes will not keep up with advances in information terrorism. If not properly planned and controlled, seemingly secure alternative routes into a fortification can negate all other protection; a castle's walls will be ineffective against an attack that does not come directly at them.

44

Cookies and Web Bugs: What They Are and How They Work Together

William T. Harding, Ph.D., Anita J. Reed, CPA, and Robert L. Gray, Ph.D.

What are cookies and what are Web bugs? Cookies are not the kind of cookies that we find in the grocery store and love to eat. Rather, cookies found on the World Wide Web are small unique text files created by a Web site and sent to your computer's hard drive. Cookie files record your mouse-clicking choices each time you get on the Internet. After you type in a Uniform Resource Locator (URL), your browser contacts that server and requests the specific Web site to be displayed on your monitor. The browser searches your hard drive to see if you already have a cookie file from the site. If you have previously visited this site, the unique identifier code, previously recorded in your cookie file, is identified and your browser will transfer the cookie file contents back to that site. Now the server has a history file of actually what you selected when you previously visited that site. You can readily see this because your previous selections are highlighted on your screen. If this is the first time you have visited this particular site, then an ID is assigned to you and this initial cookie file is saved on your hard drive.

A Web bug is a graphic on a Web page or in an e-mail message that is designed to monitor who is reading the Web page or e-mail message. A Web bug can provide the Internet Protocol (IP) address of the e-mail recipient, whether or not the recipient wishes that information disclosed. Web bugs can provide information relative to how often a message is being forwarded and read. Other uses of Web bugs are discussed in the details that follow. Additionally, Web bugs and cookies can be merged and even synchronized with a person's e-mail address. There are positive, negative, illegal, and unethical issues to explore relative to the use of Web bugs and cookies. These details also follow.

What Is a Cookie?

Only in the past few years have cookies become a controversial issue, but, as previously stated, not the kind of cookies that you find in the grocery store bearing the name "Oreos" or "Famous Amos." These cookies deal with information passed between a Web site and a computer's hard drive. Although cookies are becoming a more popular topic, there are still many users who are not aware of the cookies being stored on their hard drives. Those who are familiar with cookies are bringing up the issues of Internet privacy and ethics. Many companies such as DoubleClick, Inc. have also had lawsuits brought against them that ask the question: are Internet companies going too far?

To begin, the basics of cookies need to be explained. Lou Montulli for Netscape invented the cookie in 1994. The only reason, at the time, to invent a cookie was to enable online shopping baskets. Why the name "cookie"? According to an article entitled "Cookies … Good or Evil?," it is said that early hackers got their kicks from Andy Williams' TV variety show. A "cookie bear" sketch was often performed where a guy in a bear suit tried all kinds of tricks to get a cookie from Williams, and Williams would always end the sketch while screaming, "No cookies! Not now, not ever … NEVER!" A hacker took on the name "cookie bear" and annoyed mainframe computer operators by taking over their consoles and displaying a message "WANT COOKIE." It would not go away until the operator typed the word "cookie," and cookie bear would reply with a thank you. The "cookie" did nothing but damage the operator's nerves. Hence the name "cookie" emerged.

Cookie Contents

When cookies were first being discovered, rumors went around that these cookies could scan information off your hard drive and collect details about you, such as your passwords, credit card numbers, or a list of software on your computer. These rumors were rejected when it was explained that a cookie is not an executable program and can do nothing directly to your computer. In simple terms, cookies are small, unique text files created by a Web site and sent to a computer's hard drive. They contain a name, a value, an expiration date, and the originating site. The header contains this information and is removed from the document before the browser displays it. You will never be able to see this header, even if you execute the view or document source commands in your browser. The header is part of the cookie when it is created. When it is put on your hard drive, the header is left off. The only information left of the cookie is relevant to the server and no one else.

An example of a header is as follows:

```
Set-Cookie: NAME=VALUE; expires=DATE; path=PATH;
domain=DOMAIN_NAME; secure
```

The NAME=VALUE is required. NAME is the name of the cookie. VALUE has no relevance to the user; it is anything the origin server chooses to send. DATE determines how long the cookie will be on your hard drive. No expiration date indicates that the cookie will expire when you quit the Web browser. DOMAIN_NAME contains the address of the server that sent the cookie and that will receive a copy of this cookie when the browser requests a file from that server. It specifies the domain for which the cookie is valid. PATH is an attribute that is used to further define when a cookie is sent back to a server. Secure specifies that the cookie only be sent if a secure channel is being used.

Many different types of cookies are used. The most common type is named a visitor cookie. This keeps track of how many times you return to a site. It alerts the Webmaster of which pages are receiving multiple visits. A second type of cookie is a preference cookie that stores a user's chosen values on how to load the page. It is the basis of customized home pages and site personalization. It can remember which color schemes you prefer on the page or how many results you like from a search. The shopping basket cookie is a popular one with online ordering. It assigns an ID value to you through a cookie. As you select items, it includes that item in the ID file on the server. The most notorious and controversial is the tracking cookie. It resembles the shopping basket cookie, but instead of adding items to your ID file, it adds sites you have visited. Your buying habits are collected for targeted marketing. Potentially, companies can save e-mail addresses supplied by the user and spam you on products based on information they gathered about you.

Cookies are only used when data is moving around. After you type a URL in your browser, it contacts that server and requests that Web site. The browser looks on your machine to see if you already have a cookie file from the site. If a cookie file is found, your browser sends all the information in the cookie to that site with the URL. When the server receives the information, it can now use the cookie to discover your shopping or browsing behavior. If no cookie is received, an ID is assigned to you and sent to your machine in the form of a cookie file to be used the next time you visit.

Cookies are simply text files and can be edited or deleted from the computer system. For Netscape Navigator users, cookies can be found under (C:/Program Files/ Netscape/Users/default or user name/cookie.txt) directory, while Explorer users will find cookies stored in a folder called Cookies under (C:/windows/Cookies). Users cannot harm their computer when they delete the entire cookie folder or selected files. Web browsers have options that alert users before accepting cookies. Furthermore, there is software that allows users to block cookies, such as Zero-knowledge systems, Junkguard, and others that are found at www.download.com.

For advanced users, cookies can also be manipulated to improve their Web usage. Cookies are stored as a text string, and users can edit the expiration date, domain, and path of the cookie. For instance, JavaScript makes the cookies property of the documents object available for processing. As a string, a cookie can be manipulated like any other string literal or variable using the methods and properties of the string object.

Although the cookie is primarily a simple text file, it does require some kind of scripting to set the cookie and to allow the trouble-free flow of information back and forth between the server and client. Probably the most common language used is Perl CGI script. However, cookies can also be created using JavaScript, Livewire, Active Server Pages, or VBScript.

Here is an example of a JavaScript cookie:

```
<SCRIPT language=JavaScript>
  function setCookie (name, value, expires, path, domain,
secure) {
  document.cookie = name + "=" +  escape(value) +
  ((expires) ? "; expires=" +  expires : "") +
  ((path) ? "; path=" + path :  "") +
  ((domain) ? "; domain=" +  domain : "") +
  ((secure) ? "; secure" : "");
  }
  </SCRIPT>.
```

Although the design of the cookie is written in a different language than the more common Perl CGI script that we first observed, the content includes the same name-value pairs. Each one of these scripts is used to set and retrieve only their unique cookie and they are very similar in content. The choice of which one to use is up to the creators' personal preference and knowledge.

When it comes to being able to actually view what the cookie looks like on your system, what you get to see from the file is very limited and not easily readable. The fact is that all of the information on the cookie is only readable in its entirety by the server that set the cookie. Furthermore, in most cases, when you access the files directly from your cookies.txt file or from the windows/cookies directory with a text editor, what you see looks mostly like indecipherable numbers or computer noise. However, Karen Kenworthy of Winmag.com (one super-sleuth programmer) has created a free program that will locate and display all of the cookies on your Windows computer. Her cookie viewer program will display all the information within a cookie that is available except for any personal information that is generally hidden behind the encoded ID value. Exhibit 44.1 shows Karen's Cookie Viewer in action.

As you can see, the Cookie Viewer shows that we have 109 cookies currently inside our Windows\Cookie directory. Notice that she has added a Delete feature to the viewer to make it very easy for the user to get rid of all unwanted cookies. When we highlight the cookie named anyuser@napster[2].txt, we can see that it indeed came from napster.com and is available only to this server. If we are not sure of the Web site a cookie came from, we can go to the domain or IP address shown in this box to decide if we really need that particular cookie. If not, we can delete it! Next we see that the Data Value is set at 02b07, which is our own unique ID. This series of numbers and letters interacts with a Napster server database holding any pertinent information we have previously entered into a Napster form. Next we see the creation date, the expiration date, and a computation of the time between the two dates. We can also see that this cookie should last for ten years. The cookie viewer takes expiration dates that Netscape stores as a 32-bit binary number and makes it easily readable. Finally, we see a small window in regard to the security issue, which is set at the No default.

Positive Things about Cookies

First of all, the purpose of cookies is to keep track of information on your browsing history. When a user accesses a site that uses cookies, up to 255 bytes of information are passed to the user's browser. The next time the user visits that site, the cookie is passed back to the server. The cookie might include a list of the pages that the user has viewed or the user's viewing patterns based on prior visits. With cookies, a site can track usage patterns and customize the information displayed to individuals as they log on to the site.

Second, cookies can provide a wealth of information to marketers. By using Internet cookies, online businesses can target ads that are relevant to specific consumers' needs and interests. Both consumers and marketers can benefit from using cookies. The marketers can get a higher rate of Click-Through viewers, while

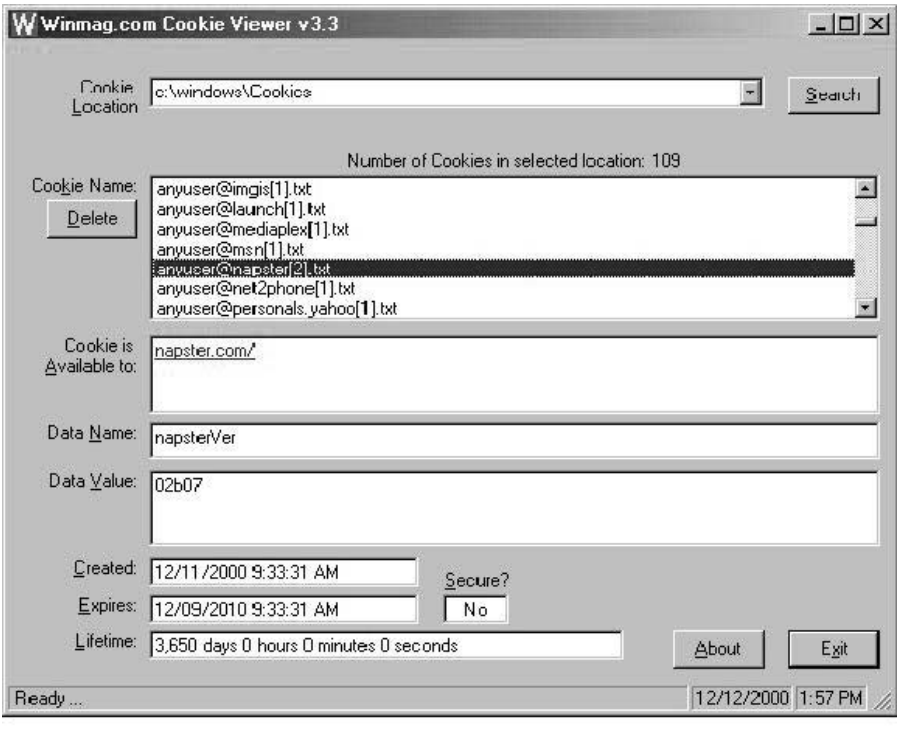


EXHIBIT 44.1 Karen's cookie viewer.

customers can view only the ads that interest them. In addition, cookies can prevent repetitive ads. Internet marketing companies such as Focalink and DoubleClick implement cookies to make sure an Internet user does not have to see the same ads over and over again. Moreover, cookies provide marketers with a better understanding of consumer behavior by examining the Web surfing habits of the users on the Internet. Advanced data mining companies like NCR, Inc. and Sift, Inc. can analyze the information about customers in the cookie files and better meet the needs of all consumers.

An online ordering system can use cookies to remember what a person wants to buy. For example, if a customer spends hours of shopping looking for a book at a site, and then suddenly has to get offline, the customer can return to the site later and the item will still be in his shopping basket.

Site personalization is another beneficial use of cookies. Let's say a person comes to the CNN.com site but does not want to see any sports news; CNN.com allows that person to select this as an option. From then on (until the cookie expires), the person will not have to see sports news at CNN.com.

Internet users can use cookies to store their passwords and user IDs, so the next time they want to log on to the Web site, they do not have to type in the password or user ID. However, this function of cookies can be a security risk if the computer is shared among other users. Hotmail and Yahoo are some of the common sites that use this type of cookie to provide quicker access for their e-mail users.

Cookies have their advantages, described in "Destroying E-Commerce's 'Cookie Monster' Image." Cookies can target ads that are relevant to specific consumers needs and interests. This benefits a user by keeping hundreds of inconvenient and unwanted ads away. The cookies prevent repetitive banner ads. Also, through the use of cookies, companies can better understand the habits of consumer behavior. This enables marketers to meet the needs of most consumers. Cookies are stored at the user's site on that specific computer. It is easy to disable cookies. In Internet Explorer 4.0, choose the View, Internet Options command, click the Advanced tab, and click the Disable All Cookies option.

Negative Issues Regarding Cookies

The main concerns about using cookie technology are the security and privacy issues. Some believe that cookies are a security risk, an invasion of privacy, and dangerous to the Internet. Whether or not cookies are ethical is based on how the information about users is collected, what information is collected, and how this information is used. Every time a user logs on to a Web site, he or she will give away information such as service provider, operating system, browser type, monitor specifications, CPU type, IP address, and what server last logged on.

A good example of the misuse of cookies is the case when a user shares a computer with other users. For example, at an Internet café, people can snoop into the last user's cookie file stored in the computer's hard disk and potentially uncover sensitive information about the earlier user. That is one reason why it is critical that Web developers do not misuse cookies and do not store information that might be deemed sensitive in a user's cookie file. Storing information such as someone's Social Security number, mother's maiden name, or credit card information in a cookie is a threat to Internet users.

There are disadvantages and limitations to what cookies can do for online businesses and Web users. Some Internet consumers have several myths about what cookies can do, so it is crucial to point out things that cookies cannot do:

- Steal or damage information from a user's hard drive
- Plant viruses that would destroy the hard drive
- Track movements from one site to another site
- Take credit card numbers without permission
- Travel with the user to another computer
- Track down names, addresses, and other information unless consumers have provided such information voluntarily

On January 27, 2000, a California woman filed suit against DoubleClick, accusing the Web advertising firm of unlawfully obtaining and selling consumers' private information. The lawsuit alleges that DoubleClick employs sophisticated computer tracking technology, known as cookies, to identify Internet users and collect personal information without their consent as they travel around the Web. In June 2000, DoubleClick purchased Abacus Direct Corporation, a direct marketing service that maintains a database of names, addresses, and the retail purchasing habits of 90 percent of American households. DoubleClick's new privacy policy states that the company plans to use the information collected by cookies to build a database profiling consumers. DoubleClick defends the practice of profiling, insisting that it allows better targeting of online ads which in turn makes the customer's online experiences more relevant and advertising more profitable. The company calls it "personalization."

According to the Electronic Privacy Information Center, "DoubleClick has compiled approximately 100 million Internet profiles to date." Consumers felt this provided DoubleClick with too much access to unsuspecting users' personal information. Consumers did not realize that most of the time they were receiving an unauthorized DoubleClick cookie. There were alleged violations of federal statutes, such as the Electronic Communication Privacy Act and the Stored Wire and Electronic Communications and Transactional Records Access Act. In March 2000, DoubleClick admitted to making a mistake in merging names with anonymous user activity.

Many people say that the best privacy policies would let consumers "opt in," having a say in whether they want to accept or reject specific information. In an article titled "Keeping Web Data Private," Electronic Data Systems (EDS) Corp. in Plano, Texas, was said to have the best practices. Bill Poulous, EDS's director of E-commerce policy stated, "Companies must tell consumers they're collecting personal information, let them know what will be done with it and give them an opportunity to opt out, or block collection of their data." Poulous also comments that policies should be posted where the average citizen can read and understand them and be able to follow them.

What Is a Web Bug?

A Web bug is a graphic on a Web page or in an e-mail message that is designed to monitor who is reading the Web page or an e-mail message. Like cookies, Web bugs are electronic tags that help Web sites and advertisers track visitors' whereabouts in cyberspace. However, Web bugs are essentially invisible on the page and are much smaller — about the size of the period at the end of a sentence. Known for tracking down the creator of the Melissa virus, Richard Smith, Chief Technology Officer of www.privacyfoundation.org, is credited with uncovering the Web bug technique. According to Smith, "Typically set as a transparent image, and only 1×1 pixel in size, a Web bug is a graphic on a Web page or in an e-mail message that is designed to monitor who is reading the Web page or e-mail message." According to Craig Nathan, Chief Technology Officer for Meconomy.com, the 1×1 pixel Web bug "is like a beacon, so that every time you hit a Web page it sends a ping or call-back to the server saying 'Hi, this is who I am and this is where I am.'"

Most computers have cookies, which are placed on a person's hard drive when a banner ad is displayed or a person signs up for an online service. Savvy Web surfers know they are being tracked when they see a banner ad. However, people cannot see Web bugs, and anti-cookie filters will not catch them. So the Web bugs can wind up tracking surfers in areas online where banner ads are not present or on sites where people may not expect to be trailed.

An example of a Web bug can be found at http:www.investorplace.com. There is a Web bug located at the top of the page. By choosing View, Source in Internet Explorer or View, Page Source in Netscape you can see the code at work. The code, as seen below, provides information about an "Investor Place" visitor to the advertising agency DoubleClick:

```
<IMG  SRC="http:ad.doubleclick.net/activity;src=328142;
      type=mmti;  cat=invstr;ord=<Time>?"WIDTH=1
      HEIGHT=1  BORDER=0>
```

It is also possible to check for bugs on a Web page. Once the page has loaded, view the page's source code. Search the page for an IMG tag that contains the attributes WIDTH=1 HEIGHT=1 BORDER=0 (or WIDTH="1" HEIGHT ="1" BORDER="0"). This indicates the presence of a small, transparent image. If the image that this tag points to is on a server other than the current server (i.e., the IMG tag contains the text SRC="http://"), it is quite likely a Web bug.

Privacy and Other Web Bug Issues

Advertising networks, such as DoubleClick or Match Point, use Web bugs (also called "Internet tags") to develop an "independent accounting" of the number of people in various regions of the world, as well as various regions of the Internet, who have accessed a particular Web site. Advertisers also account for the statistical page views within the Web sites. This is very helpful in planning and managing the effectiveness of the content because it provides a survey of target market information (i.e., the number of visits by users to the site). In this same spirit, the ad networks can use Web bugs to build a personal profile of sites a person has visited. This information can be warehoused on a database server and mined to determine what types of ads are to be shown to that user. This is referred to as "directed advertising."

Web bugs used in e-mail messages can be even more invasive. In Web-based e-mail, Web bugs can be used to determine if and when an e-mail message has been read. A Web bug can provide the IP address of the recipient, whether or not the recipient wishes that information disclosed. Within an organization, a Web bug can give an idea of how often a message is being forwarded and read. This can prove helpful in direct marketing to return statistics on the effectiveness of an ad campaign. Web bugs can be used to detect if someone has viewed a junk e-mail message or not. People who do not view a message can be removed from the list for future mailings.

With the help of a cookie, the Web bug can identify a machine, the Web page it opened, the time the visit began, and other details. That information, sent to a company that provides advertising services, can then be used to determine if someone subsequently visits another company page in the same ad network to buy something or to read other material. "It's a way of collecting consumer activity at their online store," says David Rosenblatt, senior vice president for global technology at DoubleClick. However, for consumer watchdogs,

Web bugs and other tracking tools represent a growing threat to the privacy and autonomy of online computer users.

It is also possible to add Web bugs to Microsoft Word documents. A Web bug could allow an author to track where a document is being read and how often. In addition, the author can watch how a "bugged" document is passed from one person to another or from one organization to another.

Some possible uses of Web bugs in Word documents include:

- Detecting and tracking leaks of confidential documents from a company
- Tracking possible copyright infringement of newsletters and reports
- Monitoring the distribution of a press release
- Tracking the quoting of text when it is copied from one Word document to a new document

Web bugs are made possible by the ability in Microsoft Word for a document to link to an image file that is located on a remote Web server. Because only the URL of the Web bug is stored in a document and not the actual image, Microsoft Word must fetch the image from a Web server each and every time the document is opened. This image-linking feature then puts a remote server in the position to monitor when and where a document file is being opened. The server knows the IP address and host name of the computer that is opening the document. A host name will typically include the company name of a business. The host name of a home computer usually has the name of a user's Internet service provider. Short of removing the feature that allows linking to Web images in Microsoft Word, there does not appear to be a good preventative solution. In addition to Word documents, Web bugs can also be used in Excel 2000 and PowerPoint 2000 documents.

Synchronization of Web Bugs and Cookies

Additionally, Web bugs and browser cookies can be synchronized to a particular e-mail address. This trick allows a Web site to know the identity of people (plus other personal information about them) who come to the site at a later date. To further explain this, when a cookie is placed on your computer, the server that originally placed the cookie is the only one that can read it. In theory, if two separate sites place a separate unique cookie on your computer, they cannot read the data stored in each other's cookies. This usually means, for example, that one site cannot tell that you have recently visited the other site. However, the situation is very different if the cookie placed on your computer contains information that is sent by that site to an advertising agency's server and that agency is used by both Web sites. If each of these sites places a Web bug on its page to report information back to the advertising agency's computer, every time you visit either site, details about you will be sent back to the advertising agency utilizing information stored on your computer relative to both sets of cookie files. This allows your computer to be identified as a computer that visited each of the sites.

An example will further explain this. When Bob, the Web surfer, loads a page or opens an e-mail that contains a Web bug, information is sent to the server housing the "transparent GIF." Common information being sent includes the IP address of Bob's computer, his type of browser, the URL of the Web page being viewed, the URL of the image, and the time the file was accessed. Also potentially being sent to the server, the thing that could be most threatening to Bob's privacy, is a previously set cookie value, found on his computer.

Depending on the nature of the preexisting cookie, it could contain a whole host of information from usernames and passwords to e-mail addresses and credit card information. To continue with our example, Bob may receive a cookie upon visiting Web Site #1 that contains a transparent GIF that is hosted on a specific advertising agency's server. Bob could also receive another cookie when he goes to Web Site #2 that contains a transparent GIF which is hosted on the same advertising agency's server. Then the two Web sites would be able to cross-reference Bob's activity through the cookies that are reporting to the advertiser. As this activity continues, the advertiser is able to stockpile what is considered to be non-personal information on Bob's preferences and habits, and, at the same time, there is the potential for the aggregation of Bob's personal information as well.

It is certainly technically possible, through standardized cookie codes, that different servers could synchronize their cookies and Web bugs, enabling this information to be shared across the World Wide Web. If this were to happen, just the fact that a person visited a certain Web site could be spread throughout many Internet servers, and the invasion of one's privacy could be endless.

Conclusion

The basics of cookies and Web bugs have been presented to include definitions, contents, usefulness, privacy concerns, and synchronization. Several examples of the actual code of cookies and Web bugs were illustrated to help the reader learn how to identify them. Many positive uses of cookies and Web bugs in business were discussed. Additionally, privacy and other issues regarding cookies and Web bugs were examined. Finally, the synchronization of Web bugs and cookies (even in Word documents) was discussed.

However, our discussions have primarily been limited to cookies and Web bugs as they are identified, stored, and used today only. Through cookie and Web bug metadata (stored data about data), a great deal of information could be tracked about individual user behavior across many platforms of computer systems. Someday we may see cookie and Web bug mining software filtering out all kinds of different anomalies and consumer trends from cookie and Web bug warehouses! What we have seen thus far may only be the tip of the iceberg. (Special thanks go to the following MIS students at Texas A&M University–Corpus Christi for their contributions to this research: Erik Ballenger, Cynthia Crenshaw, Robert Gaza, Jason Janacek, Russell Laya, Brandon Manrow, Tuan Nguyen, Sergio Rios, Marco Rodriquez, Daniel Shelton, and Lynn Thornton.)

Further Reading

1. Bradley, Helen. "Beware of Web Bugs & Clear GIFs: Learn How These Innocuous Tools Invade Your Privacy," *PC Privacy,* 8(4), April 2000.
2. Cattapan, Tom. "Destroying E-Commerce's 'Cookie Monster' Image," *Direct Marketing,* 62(12), 20–24+, April 2000.
3. Hancock, Bill. "Web Bugs — The New Threat!," *Computers & Security,* 18(8), 646–647, 1999.
4. Harrison, Ann. "Keeping Web Data Private," *Computerworld,* 34(19), 57, May 8, 2000.
5. Junnarkar, S. "DoubleClick Accused of Unlawful Consumer Data Use," *Cnet News,* January 28, 2000.
6. Kearns, Dave. "Explorer Patch Causes Cookie Chaos," *Network World,* 17(31), 24, July 31, 2000.
7. Kokoszka, Kevin. "Web Bugs on the Web," Available: http://writings142.tripod.com/kokoszka/paper.html
8. Kyle, Jim. "Cookies … Good or Evil?," *Developer News,* November 30, 1999.
9. Mayer-Schonberger, Viktor. "The Internet and Privacy Legislation: Cookies for a Treat?" Available: http://wvjolt.wvu.edu/wvjolt/current/issue1.
10. Olsen, Stefanie. "Nearly Undetectable Tracking Device Raises Concern," *CNET News.com,* July 12, 2000, 2:05 p.m. PT.
11. Rodger, W. "Activists Charge DoubleClick Double Cross," *USA Today,* July 6, 2000.
12. Samborn, Hope Viner. "Nibbling Away at Privacy," *ABA Journal, The Lawyer's Magazine,* 86, 26–27, June 2000.
13. Sherman, Erik. "Don't Neglect Desktop When It Comes to Security," *Computerworld,* 25, 36-37, September 2000.
14. Smith, Richard. "Microsoft Word Documents that 'Phone Home,'" *Privacy Foundation.* Available: http://www.privacyfoundation.org/ advisories/advWordBugs.html, August 2000.
15. Turban, Efraim, Lee, Jae, King, David, and Chung, H. *Electronic Commerce: A Managerial Perspective,* Prentice-Hall, 2000.
16. Williams, Jason. "Personalization vs. Privacy: The Great Online Cookie Debate," *Editor & Publisher,* 133(9), 26–27, February 28, 2000.
17. Wright, Matt. "HTTP Cookie Library," Available: http://www.worldwidemart.com/scripts/.

Web Site Sources

1. http://www.webparanoia.com/cookies.html
2. http://theblindalley.com/webbugsinfo.html
3. http://www.privacyfoundation.org/education/ webbug.html

4. http://ciac.llnl.gov/ciac/bulletins/i-034.shtml
5. http://ecommerce.ncsu.edu/csc513/ student_work/tech_cookie.html
6. http://www.rbaworld.com/security/computers/ cookies/cookies.shtml
7. http://www.howstuffworks.com/cookie2.htm

45

Leveraging Virtual Private Networks

James S. Tiller, CISA, CISSP

Increasingly, virtual private networks (VPNs) are being adopted for many uses, which range from remote access and small office/home office (SOHO) support to Business-to-Business (B2B) communications. Almost as soon as the technology became available, organizations of nearly all business verticals began implementing VPNs in some form or another. Regardless of the business type or market, VPNs seem to permeate all walks of life in the communications environment. They meet several needs for expanded communications and typically can be implemented in a manner that provides a quick return on investment.

Given the availability and scope of different products, implementing VPNs has never been easier. In many cases, VPNs are relatively easy to install and support. Many solutions are shrink-wrapped, in that products are aligned to provide what many companies wish to employ. This is not to imply that VPNs are simplistic, especially in large environments where they can become convoluted with the integration of routing protocols, access controls, and other Internetworking technologies. However, VPNs are, in essence, another form of communication platform and should be leveraged as such.

In addition to the assortment of products and generally known applications of VPNs, the excitement for the technology and the promise of secure communications are only matched by the confusion of which protocol to employ. There are several standards and types of VPNs available for the choosing, each with its own attributes that can accommodate various requirements of the solution differently than the next technology in line. Of course, each vendor has a rendition of that standard, and the method for employing it may be different from others supposedly building on the same foundations. Nevertheless, VPNs are very popular and are being deployed at an amazing rate. One can expect more of the same as time and technology advance.

VPNs are capable of providing a communication architecture that mimics traditional wide area networks. Mostly, these applications utilize the Internet to leverage a single connection to exchange data with multiple remote sites and users. Several virtual networks can be established by employing authentication, encryption, and policy, ultimately building a Web of virtual channels through the Internet.

Early in VPN interest, the Internet was considered unreliable and inconsistent. Until recently, the Internet's capabilities were questionable. Internet connections would randomly fail, data rates would greatly fluctuate, and it was generally viewed as a luxury and considered unmanageably insecure by many. In the light of limited Internet assurance, the concern of successfully transferring mission-critical, time-sensitive communications over the Internet greatly overshadowed security-related concerns. Who cared if one could secure it if the communication was too slow to be useful? As the general needs of the Internet grew, so did the infrastructure. The Internet is generally much more reliable and greater data rates are becoming more affordable. The greater number of Internet access points, increased speeds, better reliability, and advanced perimeter technology have all combined to entice the reluctant to entertain Internet-based VPNs for wide area communications.

In light of the inevitable expansion of adoption, this chapter addresses some concepts of using VPNs in a method that is not typically assumed or sold. Most certainly considered by VPN evangelists, the ideas described here are not new, but rather not common among most implementations. This chapter simply explores some ideas that can allow organizations to take advantage of environmental and technological conditions to amplify the functionality of their networks.

Key Advantages of VPNs

There are several reasons for an organization to deploy a VPN. These can include directives as simple as costs savings and increased functionality or access. Also, the reasoning may be more driven by controlling the access of extranets and the information they can obtain.

In any case, VPNs offer the quick establishment of communications utilizing existing Internet connections and provide flexibility of security-related services. Neither of these attributes are as clear-cut with conventional communications — specifically, Frame Relay (FR). It is difficult to compare the two technologies because the similarities diverge once one gets past virtual circuits; however, time and security can be discussed.

The allocation of FR circuits, especially new locations that do not have a connection to the provider, can be very time-consuming. In the event the network is managed by a third party, it may take excessive work to have a new permanent virtual circuit (PVC) added to the mix, assigned address space, and included in the routing scheme. In addition, every PVC costs money.

As far as security is concerned, the confidentiality of the data traversing the network is directly related to the provider of the communication. If no precautions are employed by the owner of the data prior to being injected onto the wide area network (WAN), the protection of the information is provided by the carrier and its interconnection relationship with other providers.

Time Is Money

In contrast to FR, in this example, VPNs can be quickly established and eliminated with very little administration. Take, for example, a company with an Internet connection and VPN equipment that wishes to establish a temporary link to another organization to obtain services. There could be currently thousands of other VPNs operating over the same connection to various remote sites and users. Even so, no physical changes need to be made and no equipment needs to be purchased — only the configuration needs to be modified to include another site. In recent history, this required a detailed configuration of each terminating point of the proposed VPN. Now, many products have extensive management capabilities that allow remote management of the VPNs. Some operate within the VPN, while others leverage management standards such as SNMPv3 for secured management over the Internet.

Given the ability to sever or create communications almost instantly, the advantages to an ever-changing communication landscape are obvious. It is not uncommon for a business to necessitate a temporary connection to another for the exchange of information. Advertising firms, consulting firms, information brokers, logistics organizations, and manufacturing companies all typically require or could take advantage of communications with their clients or partners. If the capability were there to quickly establish those communications to a controlled location, communications could be allowed to flow within a very short time frame. The same holds true once the relationship or requirement for connectivity has expired. The VPN can be removed without any concern for communication contracts, investment management, or prolonged continuance.

Security Is Money Too

The security a VPN provides may seem evident. The connection is established over the Internet, usually, and data is provided authentication and encrypted for protection while it traverses an open sea of vulnerabilities. However, some advantages are not as obvious. A good example is when multiple connections are required to various external organizations that may integrate at different locations throughout the enterprise.

A geographically large organization may have several connections to other organizations at several of its sites. There may be sites that have several different extranet connections, and in some cases, each connection may have its own router and FR service provider.

There are many security challenges in such environments. Access must be tightly controlled to eliminate attacks from either network into another, or even worse, an attack between extranets using the connectivity provided by the organization's network. Security is sometimes applied by access control lists (ACLs) on the router(s) that limit the activities available to the communication. For some organizations, security is provided by allocating a dedicated network behind a firewall. In many cases, this can suffice with centrally managed firewalls. The only problem is that many firewalls are expensive and it can be difficult to cost-justify their addition to networks with limited requirements or longevity.

In contrast, there are several cost-effective products, in many forms and sizes, that can be effectively deployed to provide secure, flexible VPNs. Now that IPSec services are available in routers, many can provide extensive VPN services along with basic communication, routing protocols, firewall services, authentication, and other attributes that enhance the final solution. Ultimately, a VPN policy can be quickly established and introduced into the router and can be used to control access through a single point. A policy uses specifics within the connection to identify certain communications and apply the appropriate security protection suite as well as limiting access into the network.

Merged Networks

VPNs were initiated into the industry for remote access solutions. Roaming users dialing into vast modem pools were usually provided toll-free numbers, or simply access numbers that they used to connect to the home office. The cost was time sensitive, as was building the remote access solution itself. VPNs allowed the remote user to connect to the Internet and establish a private channel to the home office. The Internet connection was not time sensitive and usually proved to be cost-effective. The cost savings were further realized at the home office in that a single device could support thousands of simultaneous users. This was a quantum leap from the traditional dial-in solution.

During this time, many organizations were considering using the same concept for network-to-network communication. This started in the form of supporting small remote offices or virtual offices for employees working from home. The advent of broadband Internet access for the private community catapulted the use of VPNs to capture the cost-efficient, high-bandwidth access available for homes and remote offices.

It soon became evident that the same concepts could be used to enhance the traditional WAN. The concept of supporting remote offices expanded to support larger and larger sites of the organization. Usually, VPNs were employed at sites that had limited communication requirements or bandwidth. The practice of migrating portions of an organization that had few communication requirements was because the thought is if the VPN fails, there will be negligible impact on business operations. Much of this is because of the unknowns of the Internet and VPN technology itself.

Many organizations today have Internet access points at several sites. These can be leveraged to create VPNs between other offices and partners. The advantages of a mixed WAN, built from traditional WAN technologies and VPNs, become evident under certain conditions.

Logical Independence

Companies usually have one or more corporate offices or data hubs that supply various services, such as e-mail and data management, to other branch offices, small remote offices, virtual home offices, and remote users. Communications between non-hub sites, such as branch offices, can be intensive for large organizations, especially when business units are spread across various sites.

Exhibit 45.1 illustrates a traditional network connecting sites to one another. An FR cloud provides connections through the use of PVCs. To accomplish this, the remote site must have access to the FR cloud. If the

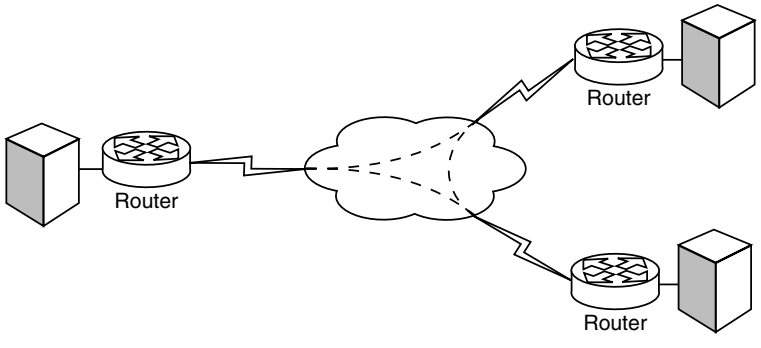


EXHIBIT 45.1 Traditional WAN environment.

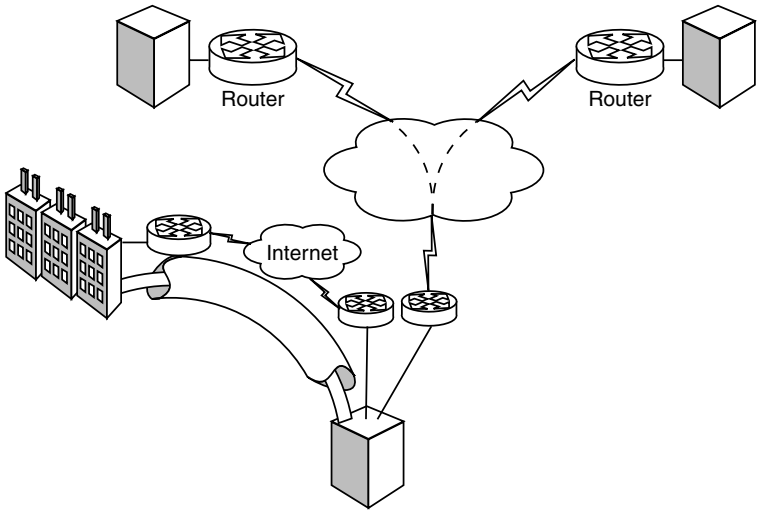


EXHIBIT 45.2 Basic VPN use.

FR service provider does not offer the service in a certain region, the organization may be forced to use another company and rely on the providers to interconnect the FR circuit. To avoid this, some organizations employ VPNs, usually because Internet connections are readily available.

As shown in Exhibit 45.2, a VPN can be quickly integrated into an FR-based WAN to provide communications. The site providing the primary Internet connection for the WAN can allow access to the centralized data. It is feasible to add many remote sites using a VPN. Once the initial investment is made at the corporate site, adding another site only incurs costs at the remote location.

It is worth noting that the VPN can now provide access to remote users from the corporate office, or access for managers to the remote office from home.

As depicted in Exhibit 45.3, the corporate site can be used as a gateway to the other locations across the WAN. In this example, it is obvious how closely the VPN mimics a traditional network. It is not uncommon for a central site to provide communications throughout the WAN. This configuration is usually referred to as "hub & spoke" and many companies employ a version of this architecture.

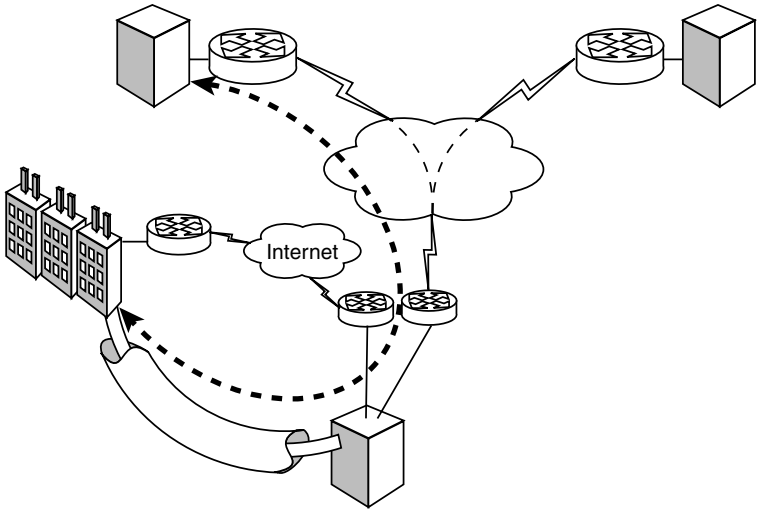


EXHIBIT 45.3 VPN integration.

As remote sites are added, the VPN can be leveraged as a separate WAN and the hub site treated as a simple gateway as with normal hub & spoke WAN operations. The VPN provides ample flexibility, cost savings, and some added advantages, including remote access support, while operating similarly to the customary WAN.

As companies grow, each site is usually provided an Internet connection to reduce the Internet traffic over the WAN. The reality is that more connections to the Internet simply equate to more points for the realization of threats. Nevertheless, organizations that have several connections to the Internet in their environment can leverage the VPN in ways not feasible in the FR world.

As seen in Exhibit 45.4, a VPN can be created to bypass the original corporate site and get to the final destination directly. What is even more interesting is that the process is automatic. For example, if the remote WAN site is addressed 10.10.10.0 and the corporate site providing the VPN termination is 20.20.20.0, the remote warehouse will make a request for 10.10.10.0. If there is only one VPN option, the request will be forwarded across the VPN to the 20.20.20.0 network where the WAN will provide the communication to the 10.10.10.0 network. However, if the 10.10.10.0 network has VPN capabilities, the remote warehouse can be easily configured to forward traffic to 10.10.10.0 to the site's VPN device. The same holds true for the 20.20.20.0 network.

Integration of Routing

Routing protocols are used in complex networks to make determinations in communication management. These protocols traverse the same communication channel as the user data and learn from the path taken. For example, distant-vector routing protocols base their metrics on the distance between sites, while link-state routing protocols ensure that the link is established. In either case, routing decisions can be made based on these basic fundamentals, along with administrative limitations such as cost and bandwidth utilization. These definitions are excessively simplistic; however, the goal is to convey that data is directed through networks based on information collected from the network itself.

Therefore, as traditional networks integrate VPNs, routing decisions take on new meaning. For example, for a five-site WAN that migrated three of them to VPNs, there are few decisions to make between the Internet-based sites. Because the communication conduit is virtual, the routing protocol only "sees" the impression of a circuit. As a routing protocol packet is injected into the data stream that is ultimately tunneled in a VPN, it is passed through a labyrinth of networks that interact with the packet envelope, while the original routing protocol packet is passed quietly in its cocoon. From the routing protocol's perspective, the network is perfect.

Putting aside the fact that the routing protocol is virtually oblivious to the vast networks traversed by the VPN, in the event of a system failure there will not be too many options on the Internet side of the router. If a remote system fails, an alternate route can be instantly constructed, rather than monitored for availability

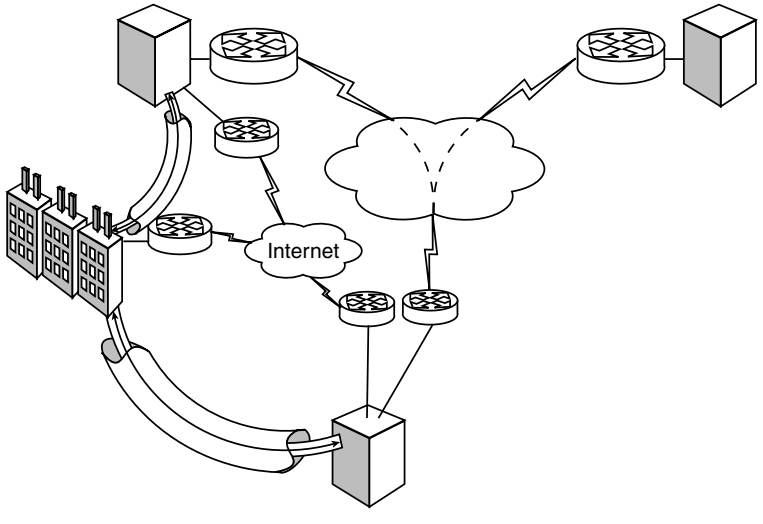


EXHIBIT 45.4 VPN providing logical freedom.

as with routing protocols. A connection can be created to another site that may have a subordinate route to the ultimate destination. This can include a traditional WAN link.

Policy-Based Routing

Some interesting changes are taking place to accommodate the integration of VPNs into conventional networks. The routing decisions are getting segmented and treated much differently. First, routing protocols are being used in simple networks where they are usually not in a traditional WAN, and routing decisions have moved to the edge devices providing the VPN. Meanwhile, the VPN cloud over the Internet is becoming a routing black hole.

To illustrate, OSPF (Open Shortest Path First) is a routing protocol that provides a hierarchical structure by the employment of Areas. Areas provide administrative domains and assist in summarizing routing information that ultimately interacts with Area 0. In this example network, there are three routers in Area 0 — Area Border Routers (ABRs) A, B, and C — each communicating by VPN. In addition to sharing information in Area 0, each has other routers in remote sites that make up supporting Areas.

As demonstrated in Exhibit 45.5, the Internet-based VPN is Area 0 and the remote site clusters represent three sub-areas (also referred to as stub, subordinate, and autonomous areas). Routing information regarding the network is shared between the sub-areas and their respective ABRs. The ABRs, in turn, share that information between them and ultimately with their supported sub-areas. In this scenario, the entire network is aware of communication states throughout and can make determinations based on that data. It is necessary that the ABRs share link information within Area 0 to ensure that sub-area routing data is provided to the other ABRs and sub-areas, and to ensure that the best, or configured, route is being used for that communication. For example, in a traditional WAN, it may be less expensive or simply easier to route from A to C through B. To accomplish this, or any other variation, the ABRs must share Area 0-specific information learned from the network.

Once a VPN is introduced into Area 0, the OSPF running between the ABRs is encapsulated. Therefore, information between Areas is being shared, but little is being learned from Area 0 between the ABRs. In reality, there is little to actually be gained. If the OSPF running between the ABRs were to learn from the network, it would be the Internet and little could be done to alter a route.

The result is that the routing protocol becomes a messenger of information between remote sites but has little impact on the virtual communications. To accommodate complicated VPNs and the fact that there is little to learn from the Internet — and what one can learn might be too complex to be utilized — policies can be created to provide alternates in communications. Because a virtual channel in a VPN architecture is, for the most part, free, one only needs to create a VPN when applicable.

VPN-Determined Routing

As described, in a conventional WAN, it may be applicable to route from A to C through B, especially if C's connection to A fails. Routing protocols observe the breakdown and can agree that re-routing through B is a

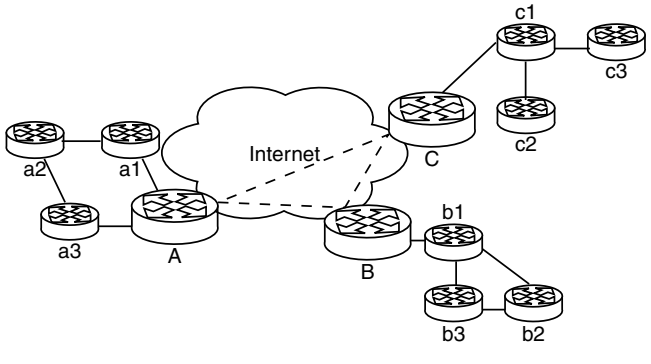


EXHIBIT 45.5 VPN effects on routed networks.

viable alternative. In a VPN, this can get more intense, depending on the configuration. For example, assume the Internet connection goes down at site C but there is a backup to site B, possibly through the supported Areas, such as a connection from c1 to b2. The ABRs know about the alternate route, but the cost is too great for normal use. When the Internet connection goes down, the VPN fails and the routing protocol cannot make a decision because there are literally no options to get across the VPN it is aware of. In short, the exit point for the data and routing protocols is simply an Internet interface. At that point, the VPN policy takes over and routes information through a particular VPN based on destination.

To accommodate this, VPN systems can learn from the routing protocols and include what is learned in the policy. Because the routing protocol is simply injected into an interface and the ultimate VPN it traverses is determined by a policy, the policy will conclude that a VPN between A and B can be leveraged because there is an alternate route between the A and C Areas between c1 and b2.

It is not necessary for the VPN to advertise its VPNs as routes to the routing protocol because they can be created or dropped instantly. The creation and deletion of routes can wreak havoc on a routing protocol. Many routing protocols, such as OSPF, do not converge immediately when a new route is discovered or an existing one is deleted, but it can certainly have an impact, depending on the frequency and duration with which the route appears and disappears.

The final hurdle in this complicated marriage between policy and routing protocol occurs when there are several connections to the Internet at one location. It is at this point that the two-pronged approach to routing requires a third influence. Typically, the Border Gateway Protocol (BGP) is used by ISPs to manage multiple connections to a single entity. The organization interfaces with the ISP's BGP routing tables to learn routes from the ISP to the Internet, as well as the ISP learning changes to the customer's premise equipment. The VPN systems must take the multiple routes into consideration; however, as long as the logical link between sites is not disrupted (such as with IP address changes), the VPN will survive route modifications.

Ultimately, it is necessary to understand that VPNs are typically destination-based routed and the termination point is identified by policy to forward the data to the appropriate VPN termination point. As VPN devices learn from routing protocols, they can become a surrogate for the routing protocol they learn from and provide a seemingly perfect conduit for the sharing of routing information.

Off-Loading the WAN

One of the most obvious uses for VPNs, yet not commonly seen in the field, is WAN off-loading. The premise is to leverage the VPN infrastructure as an augmentation to the WAN, rather than a replacement. VPNs can be implemented with little additional investment or complexity, and collaboration with a WAN will promote some interesting effects.

It is worth stating that when a VPN is implemented as a WAN replacement, the virtual nature of the new infrastructure lends itself to being leveraged easily. This is a prime example of leveraging VPNs. Take an existing infrastructure that may be originally put in place for remote access, mold it into a remote office support structure, and leverage that to provide WAN off-loading. Most of these concepts can be realized from the initial investment if the preliminary planning was comprehensive.

Time-Insensitive Communications

The title of this chapter section surely requires a second look. In today's technical environment, it seems that everything is time sensitive. However, there are applications that are heavily used by the populous of computer users that are not time sensitive.

E-mail is an example of an application that is not time sensitive, when compared to other applications such as chat. Interestingly enough, e-mail is a critical part of business operations for many companies, yet instant delivery is not expected, nor required. A few-minute wait for a message is nearly unnoticeable. In addition to being the lifeblood for organizations, in some cases, e-mail is used as a data-sharing platform. Everyone has witnessed the 5-MB attachment to 1334 recipients and the flood of flames that reflect back to the poor soul who started the whole thing. Of course, there are a few people who reply to all and inadvertently include the original attachment. The concept is made clear: e-mail can create a serious load on the network. It would not be out of line to state that some networks were engineered simply to enhance performance for enlarging e-mail requirements.

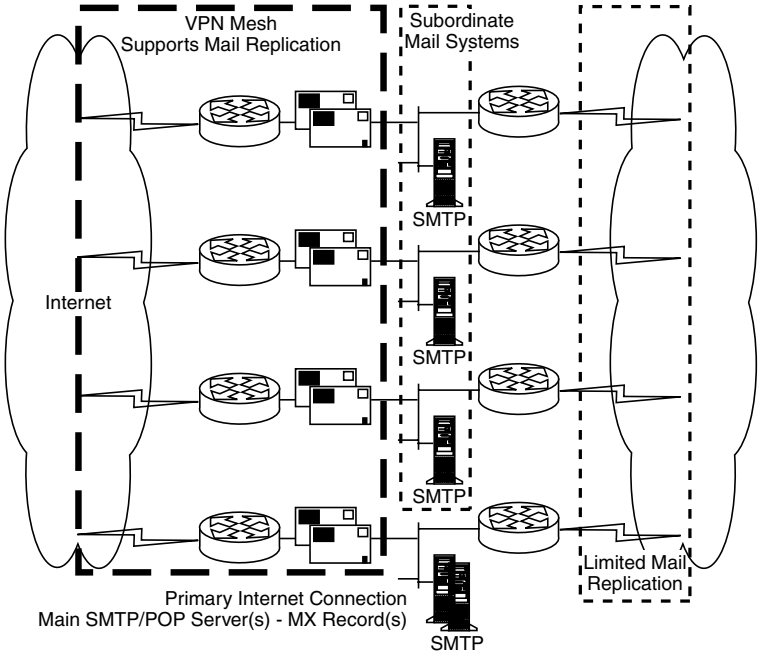


EXHIBIT 45.6 VPN providing alternate communication or specific applications.

A VPN network can be created to mirror the WAN and leveraged for the specific application. For example, in Exhibit 45.6, a VPN can provide the communication platform for the replication of e-mail throughout a domain.

In many e-mail infrastructures, collaboration between mail services is created to get e-mail to its final destination. The public mail service may be connected to the Internet at a single point at the corporate headquarters. As e-mail is received for a user in a remote site, the primary server will forward it to the service that maintains the user's mailbox for later delivery. The mail servers are connected logically and relationships are established, such as sites to collect servers into manageable groups.

The relationships between servers can be configured in such a manner as to direct the flow of data in a direction away from the normal communication channels and toward the VPN. The advantages should become clear immediately. Large attachments, large distributions lists, newsletters, and general communications are shunted onto the Internet where bandwidth restrictions may slow the progress, but the WAN is released from the burden.

Depending on the volume of e-mail relative to other data flows across the WAN, substantial cost savings can be realized above and beyond the original savings accrued during the initial VPN implementation. For example, if bandwidth requirements are reduced on the WAN, the cost can be reduced as well.

The concept of leveraging VPNs is especially significant for international companies that may use only a handful of applications across the expensive WAN links. Some large organizations use posting processes to reduce the load and align efforts around the globe by bulk processing. These packages can be delivered easily over a VPN, reducing the load on the less cost-effective WAN that is used for more time-sensitive applications.

Another example of posting is observed in the retail industry. Many companies are engineered to collect point-of-sale (POS) information and provide limited local processes such as credit card verification and local merchandise management. There comes a point when the information needs to be sent back to a home office for total processes to manage the business from a national or global scale. On one occasion, the communication was provided to the stores by satellite — nearly 120 stores nationwide. A router existed at each location to provide IP connectivity for the POS system, e-mail, and in some cases, phone lines. Between the cost of the VSAT service and the ground-station equipment, an Internet connection and VPN saved nearly 40 percent in costs and the bandwidth was increased by 50 percent. The result was that the posting took much less time, ultimately freeing up cycles on the mainframe for processing, which at the time was becoming crucial.

In addition to fulfilling several needs with a single solution — increased bandwidth and greater efficiency in processing — each store now had VPN capabilities. As the POS application capabilities increased, store-to-store determination could be made directly for regional product supply. That is, the register scanner can locate the nearest store to that location that has the product a customer desires without contacting the corporate office. This is not revolutionary, but the creation of a dynamic VPN for that transaction is.

Security is the final off-loading advantage. Of course, security is a huge selling point for VPNs and the words rarely appear separate from each other. However, this chapter addresses leveraging of the communication technology rather than the implied security. But security is a tangible asset. For example, the e-mail off-load illustration can be configured to protect e-mail in transit. A VPN can be created between each mail server at the operating system level, resulting in the encryption of all inter-domain exchanges. Although mail encryption programs are widely available, many users simply do not use them. When an organization finally gets users to encrypt messages, an administrative key should be included to avoid data loss in the face of a disgruntled employee.

Somewhere in between exists the security advantage of VPNs. Inter-domain traffic is encrypted and users are none the wiser. Of course, this cannot be directly compared to PGP (Pretty Good Privacy) or Certificates, but it keeps the general observer from accessing stray e-mail. For every e-mail system that does not employ encryption for inter-domain delivery, the e-mail is exposed at all points — on the Internet and intranet.

Fail-over

One of the interesting aspects of VPNs is that once they are in place, a world of options begins to open. By multiplexing several virtual connections through a single point — which can also be considered a single cost — the original investment can be leveraged for several other opportunities.

An example is WAN fail-over. WAN fail-over is much like the merger of VPNs and WANs; however, the VPN can provide an alternate route for some or all of the original traffic that would normally have been blocked due to a failure somewhere in the WAN infrastructure.

Consider the following example. A service provider (SP), called Phoenix, that provides not only application services but also FR services to various clients nationwide has a plethora of client and service combinations. Some clients purchase simple FR services, while others use the SP for Internet access. Of these clients, many are end users of applications, such as ERP systems, human resource systems, e-mail, kiosks, off-site storage, and collaboration tools. To maintain the level of service, Phoenix maintains a network operations center (NOC) that provides network management and support for the clients, applications, and communications.

For the FR customers that use the provided communication to access the applications, a VPN can be implemented to support the application in the event the FR were to fail. Many organizations have Internet connections as well as dedicated communications for vital application requirements. Therefore, leveraging one against the other can present great fault tolerance opportunities. This is relatively easy to configure and is an example of how to use the Internet with VPN technology to maintain connectivity.

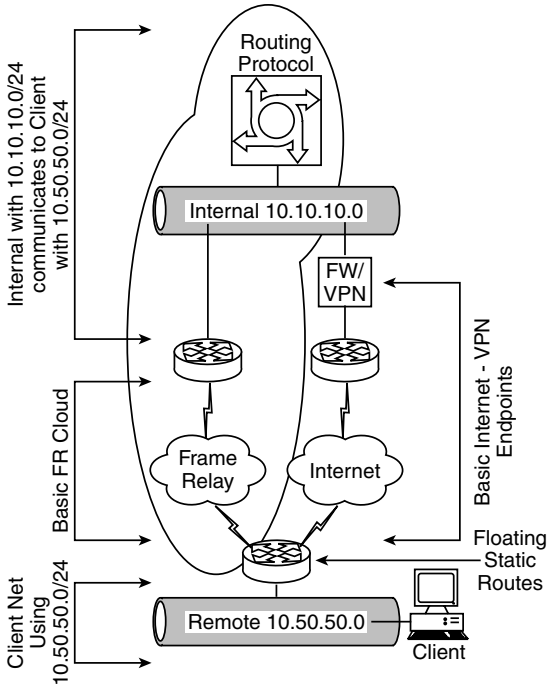
As shown in Exhibit 45.7, a dedicated connection can be created in concert with an Internet connection.

At Phoenix, there exists the standard FR cloud to support clients. This network is connected to the NOC via the core switch. Connected to the switch is the VPN provisioning device. In this example, it is a firewall as well, which provides Internet connectivity. On the client network, there is a router that has two interfaces: one for the dedicated circuit connection and the other to the Internet.

Based on the earlier discussion of routing protocols and VPNs, it does not help to operate routing protocols over the VPN. In reality, it is impossible in this case for two very basic reasons. The routing protocol used is multicast based and many firewalls are multicast unfriendly. Also, it is not a very secure practice to permit routing protocols through a firewall, even if it is for a VPN.

To accommodate the routing protocol restrictions, two design aspects are employed. First is a routing protocol employed as normal through the FR cloud to maintain the large number of customers and their networks. Second, floating static routes are employed on the customer's router. Essentially, a floating static route moves up the routing table when an automated route entry is deleted. For example, if a route is learned by OSPF, it will move to the top of the routing table. If the route is no longer valid and OSPF deletes the route from the routing table, the static route will take precedence.

The solution operates normally, with OSPF seeing the FR as the primary communication (based on administrative cost) back to Phoenix. In the event that a circuit fails, OSPF will delete the route in the client's routing table that directs traffic toward the FR cloud and the Internet route will take over. As a



Note: Here, one router on the client's network provides communication to the Internet as well as to Phoenix. This is shown for simplicity. It is possible that several routers can be used in the configuration with no functional impact.

EXHIBIT 45.7 VPN providing alternate communication or specific applications.

packet, which is destined for the SP, is injected into the interface, the VPN policy will identify the traffic and create a VPN with the SP. Once the VPN is created, data flows freely across the VPN onto the SP's network. As the data returns toward the client, it knows to go to the VPN device because the FR router on their side has performed the same routing deductions and forwards the packet to the VPN device.

The fail-over to the VPN can take some time, depending on how fast the VPN can be established. In contrast, the fail-back is instant. As the FR circuit comes back online, OSPF monitors the link and, once the link is determined to be sound, the routing protocol places the new route back into the routing table. From this point, the traffic simply starts going through the FR cloud. Interestingly, if the FR circuit were to fail prior to the VPN lifetime expiration, the fail-over would be nearly instant as well. Because the VPN is idle, the first packet is sent immediately.

There are some issues with this design; two, in fact, are glaring. If the FR cloud were to completely fail, all the FR customers with VPN backup would request a VPN at the same time, surely overloading the VPN system. There are a couple of options to accommodate such a problem. A load management solution can be implemented that redirects traffic to a cluster of VPN devices, distributing the load across all systems. A cheaper method is to simply modify the VPN policy on the client router to go to a different VPN device than the next. In short, distribute the load manually.

The other issues come into play when the SP wants to implement an FR connection in a network that uses Internet routable IP addresses, or some other scheme. This normally would not be a problem, but there is customer premise equipment (CPE) that needs to be managed by the SP to provide the services. An example is an application gateway server. The NOC would have a set of IP addresses to manage devices over the FR, but after the fail-over, those IP addresses may change.

Similar to Exhibit 45.7, Exhibit 45.8 employs NAT (network address translation) to ensure that no matter the source IP address of the managed elements on the client's network, the SP's NOC will always observe the same IP address.

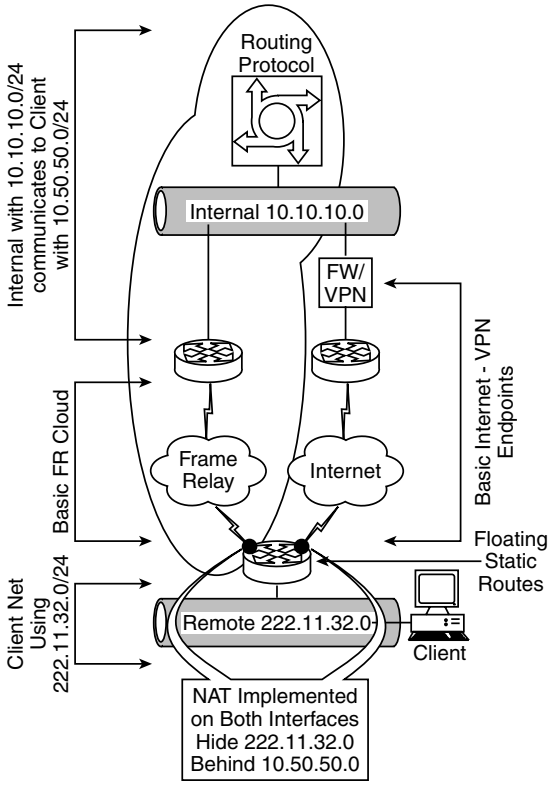


EXHIBIT 45.8 VPN fail-over using network address translation.

Conclusion

As VPNs were introduced to the technical community, network-to-network VPNs were the primary structure. Some vendors pushed the roaming user aspect of VPN technology. Remote user support became the character attached to VPN technology. This was because of the vendor focus, and the Internet was simply not seen as a safe medium.

Today, remote access VPNs are the standard, and the ability to support 5000 simultaneous connections is typical among the popular products. However, the use of VPN communications for conventional network infrastructures has not experienced the same voracious acceptance.

VPNs have been tagged as the "secure remote access" solution, and the value of a virtual connection through a public network has yet to be fully discovered. VPNs are a network and can be treated as such. As long as the fundamentals are understood, accepted, and worked with in the final design, the ability to salvage as much functionality will become apparent.

46

Wireless LAN Security

Mandy Andress, CISSP, SSCP, CPA, CISA

Wireless LANs provide mobility. Who does not want to be able to carry his laptop to the conference room down the hall and still have complete network access without worrying about network cables? Manufacturing companies are even using wireless LANs (WLANs) to monitor shop-floor machinery that is not traditionally accessible by network cabling. Increased mobility and accessibility improves communication, productivity, and efficiency. How much more productive could a team meeting be if all participants meeting in the conference room still had access to the network and the files relating to the project being discussed?

Wireless LANs can also provide a cost benefit. Installing and configuring wired communications can be costly, especially in those hard-to-reach areas. Ladders, drop ceilings, heavy furniture, knee pads, and a lot of time are often necessary to get all components installed and connected properly. By comparison, wireless LAN installations are a breeze. Plug in the access point, install a wireless NIC, and one is all set. An access point is the device that acts as a gateway for wireless devices. Through this gateway, wireless devices access the network. See Exhibit 4646.1 for an illustration.

The increased mobility and cost-effectiveness make wireless LANs a popular alternative. The Gartner Group has predicted that wireless LAN revenue would total $487 million in 2001, and the value of installed wireless LANs will grow to $35.8 billion by 2004. The Cahners In-Stat Group has predicted that the wireless LAN market will grow 25 percent annually over the next few years, from $771 million in 2000 to $2.2 billion in 2004. While these estimates are quite different, they share one common theme: a significant number of new wireless LANs will be deployed and existing installations will be expanded. This growth will occur because increases in speed, decreases in price, and the adoption of a formal standard with broad industry support have all occurred in the past few years.

Standards

Before discussing security issues with wireless LANs, a discussion of the standards that are the basis for communication is in order. In June 1997, the IEEE (Institute of Electrical and Electronic Engineers) finalized the initial standard for wireless LANs, IEEE 802.11. This standard specifies a 2.4-GHz operating frequency with data rates of 1 and 2 Mbps and the ability to choose between using frequency hopping or direct sequence, two noncompatible forms of spread spectrum modulation. In late 1999, the IEEE published two supplements to the initial 802.11 standard: 802.11a and 802.11b.

Like the initial standard, 802.11b operates in the 2.4-GHz band, but data rates can be as high as 11 Mbps and only direct sequence modulation is specified. The 802.11a standard specifies operation in the 5-GHz band using OFDM (orthogonal frequency division multiplexing) with data rates up to 54 Mbps. Advantages of this standard include higher capacity and less RF interference with other types of devices.

Standards 802.11a and 802.11b operate in different frequencies; thus, there is little chance they will be interoperable. They can coexist on one network, however, because there is no signal overlap. Some vendors claim they will provide a dual-radio system with 802.11a and 802.11b in the future.

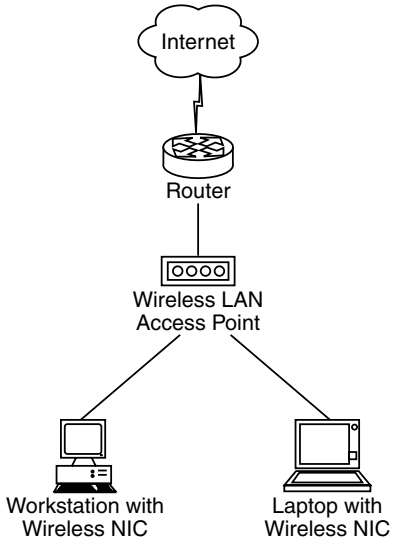


EXHIBIT 46.1 Wireless access points server as the network gateway.

To complicate issues, Europe has developed the HiperLAN/2 standard, led by the European Telecommunications Standards Institute (ETSI). HiperLAN/2 and 802.11a share some similarities: both use OFDM technology to achieve their data rates in the 5-GHz range, but they are not interoperable.

For the remainder of this chapter, discussions will focus on 802.11b wireless LANs because they comprise the current installed base.

Security Issues

Wireless LANs have major security issues. Default configurations, network architecture, encryption weaknesses, and physical security are all areas that cause problems for wireless LAN installations.

Default Installations

Default installations of most wireless networks allow any wireless NIC to access the network without any form of authentication. One can easily drive around with laptop in hand and pick up many network connections. Because this vulnerability is so prevalent, "war driving" is quickly replacing "war dialing" as the method of finding backdoors into a network. Wireless LAN administrators may realize that radio waves are easier to tap passively than cable, but they may not realize just how vulnerable they really are.

Wireless ISPs must be very conscious of their wireless network configurations. If someone is able to access their networks without authentication, they are essentially stealing service. The wireless ISP is losing revenue and the illegal user is taking up valuable bandwidth.

Once a user gains access to the wireless network, whether authorized or unauthorized, the only things preventing him from accessing unauthorized servers or applications are internal security controls. If these are weak or nonexistent, an unauthorized user could easily gain access to one's network through the wireless LAN and then gain complete control of one's network by exploiting internal weaknesses.

Denial-of-service attacks are also a very real threat to wireless networks. If running a mission-critical system on a wireless network, attackers do not need to gain access to any system to cause damage and financial harm to an organization; they just need to flood the network with bogus radio transmissions.

Mitigating Risk

To use wireless LANs in an enterprise or production environment, one must mitigate the inherent risk in current products and standards. Enterprise-level wireless LAN security focuses on two issues: network access

must be limited to authorized users, and wireless traffic should be protected from sniffing. The 802.11b standard does include some security mechanisms, but their scalability is questionable.

MAC Address

One way to secure access to a wireless network is to instruct access points to pass only those packets originating from a list of known addresses. Of course, MAC (Media Access Control) addresses can be spoofed, but an attacker would have to learn the address of an authorized user's Ethernet card before this is successful. Unfortunately, many wireless cards have the MAC address printed right on the face of the card.

Even if the user and administrator can secure the card address, they still have to compile, maintain, and distribute a list of valid MAC addresses to each access point. This method of security is not feasible in a lot of public WLAN applications, such as those found in airports, hotels, and conferences, because they do not know their user community in advance. Additionally, each brand of access point has some limit on the number of addresses allowed.

Service Set ID

Another setting on the access point that can be used to restrict access is the network name, also known as the SSID (Service Set ID). An access point can be configured to allow any client to connect to it or to require that a client specifically request the access point by name. Although this was not meant primarily as a security feature, setting the access point to require the SSID can let the ID act as a shared group password.

As with any password scheme, however, the more people who know the password, the higher the probability that an unauthorized user will misuse it. The SSID can be changed periodically, but each user must be notified of the new ID and reconfigure his wireless NIC.

Wired Equivalent Privacy (WEP)

The 802.11b standard provides encrypted communication between clients and access points via WEP (Wired Equivalent Privacy). Under WEP, users of a given access point often share the same encryption key. To achieve mobility within a campus, all access points must be set to use the same key and all clients have the same encryption key as well. Additionally, data headers remain unencrypted so that anyone can see the source and destination of data transmission.

WEP is a weak protocol that uses 40- and 128-bit RC4. It was designed to be computationally efficient, self-synchronizing, and exportable. These are the characteristics that ultimately crippled it. The following are just a few of the attacks that could easily be launched against WEP:

- Passive attacks to decrypt traffic based on statistical analysis
- Active attacks to inject new traffic from unauthorized mobile stations, based on known plaintext
- Dictionary-building attack that, after analysis of about a day's worth of traffic, allows real-time automated decryption of all traffic

With these limitations, some vendors do not implement WEP, although most provide models with and without it. An access point can be configured to never use WEP or to always require the use of WEP. In the latter case, an encrypted challenge is sent to the client. If the client cannot respond correctly, it will not be allowed to use the access point, making the WEP key another password. As with using the SSID as a password, the administrator could routinely change the WEP key, but would have the same client notification and configuration issues.

Of course, an attacker possessing the WEP key could sniff packets off the airwaves and decrypt them. Nonetheless, requiring WEP substantially raises the minimum skillset needed to intercept and read wireless data.

Authentication Solutions

Some vendors offer proprietary solutions to the authentication/scalability problem. The wireless client requests authorization from the access point, which forwards the request to a RADIUS server. Upon authorization, the RADIUS server sends a unique encryption key for the current session to the access point, which transmits it to the client. While this standard offers a solution to the shared key problem, it currently requires an organization to buy all the equipment from one vendor. Other vendors use public key cryptography to generate per-session keys.

This authentication solution resembles pre-standard implementations of the pending IEEE 802.1x standard that will eventually solve this problem in a vendor-interoperable manner. The 802.1x standard is being developed as a general-purpose access-control mechanism for the entire range of 802 technologies. The authentication mechanism is based on the Extensible Authentication Protocol (EAP) in RADIUS. EAP lets a client negotiate authentication protocols with the authentication server. Additionally, the 802.1x standard allows encryption keys for the connection to be exchanged. This standard could appear in wireless products as early as 2002.

While waiting for 802.1x, there are a few other approaches the administrator can take to increase the security of a wireless LAN.

Third-Party Products

Several products exist to secure wireless LANs. For example, WRQ's NetMotion (www.netmotionwireless.com) requires a user login that is authenticated through Windows NT. It uses better encryption (3DES and Twofish) than WEP and offers management features such as the ability to remotely disable a wireless network card's connection. One of the main issues with this solution is that the server currently must run on Windows NT and client support is only provided for Windows 95, 98, ME, and CE. Support for Windows 2000 server and client is currently under development.

Gateway Control

Gateway solutions create special sub-nets for wireless traffic. Instead of using normal routers, these sub-nets have gateways that require authentication before packets can be routed. The sub-nets can be created with VLAN technology using the IEEE 802.1Q standard. With this standard, administrators can combine selected ports from different switches into a single sub-net. This is possible even if the switches are geographically separated as long as VLAN trunking is supported on the intervening switches. Nodes that use VLAN ports cannot access addresses on other sub-nets without going through a router or gateway, even if those other sub-nets are located on the same physical switch as the VLAN ports.

Once the VLAN is established, administrators need to create a gateway that will pass traffic only from authorized users. A VPN gateway can be used because the function of a VPN server is to require an endpoint. Using a VPN server as the gateway not only requires authentication of the tunnel endpoint, but it also encrypts the wireless stream with a key unique to the tunnel, eliminating the need to use the shared key of WEP.

The VPN approach is hardly ideal, however. Understanding VPN technology, selecting a VPN gateway, configuring the server, and supporting clients are complex tasks that are not easy for the average LAN administrator to accomplish.

Another solution, currently used by Georgia Tech, uses a special firewall gateway. This approach still uses the VLAN approach to aggregate wireless traffic to one gateway; but instead of being a VPN, this gateway is a dual-homed UNIX server running specialized code. The IT staff at Georgia Tech uses the IP Tables firewall function in the latest Linux kernel to provide packet filtering. When a system joins the wireless network, the firewall/router gives it a DHCP address. To authorize access, the client must open a Web browser. The HTTP request from the client triggers an automatic redirect authentication page from the gateway and the authentication request is passed to a Kerberos server. If authentication is successful, a Perl script adds the IP address to the rules file, making it a "known" address to the IP Tables firewall process.

From the user's perspective, he must launch a browser and enter a userid and password to gain access to the network. No client installation or configuration is required. Of course, this method only provides authentication — not encryption — and will not scale over a few hundred simultaneous users. This solution

is unique and elegant in the fact it allows complete on-the-fly network access without making any changes to the client, and it supports network cards from multiple vendors. This configuration is very useful in public WLAN applications (airports, hotels, conferences, etc.).

Conclusion

Wireless LANs have several security issues that preclude them from being used for highly sensitive networks. Poor infrastructure design, unauthorized usage, eavesdropping, interception, DoS attacks, and client system theft are all areas that one needs to analyze and consider. One can mitigate these risks by wrapping the communication in a VPN or developing one's own creative solution, but this can be complicated. New advancements in wireless technology, along with changes in the WEP standard, may improve security as well as usability.

EXPANDING INTERNET SUPPORT WITH IPV6

Gilbert Held

INSIDE

New and Renamed IPv6 Fields; Header Chains; Addressing; IPv6 Address Notation; Address Assignments; Migration Issues

OVERVIEW

The ability to obtain an appreciation for the functionality of IPv6 is best obtained by comparing its header to the IPv4 header. Exhibit 1 provides this comparison, showing the IPv4 header at the top of the illustration, with the IPv6 header below.

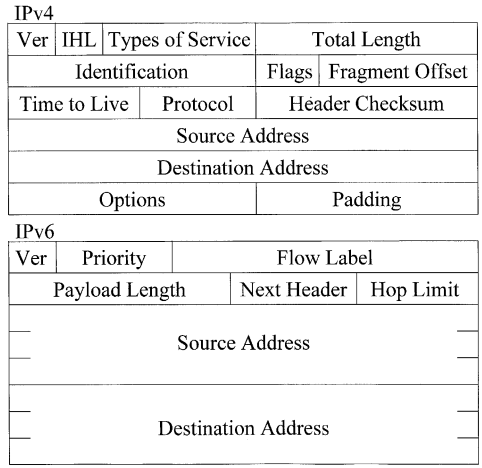
In comparing the two headers shown in Exhibit 1, one notes that IPv6 includes six less fields than the current version of the Internet Protocol. Although at first glance this appears to make an IPv6 header simpler, in actuality the IPv6 header includes a Next Header field that enables one header to point to a following header, in effect resulting in a daisy chain of headers. While the daisy chain adds complexity, only certain routers need to examine the contents of different headers, facilitating router processing. Thus, an IPv6 header, which can consist of a sequence of headers in a daisy chain, enables routers to process information directly applicable to their routing requirements. This makes IPv6 packet processing much more efficient for intermediate routers when data flows between two Internet locations, enabling those routers to process more packets per second than when the data flow consists of IPv4 headers.

A close examination of the two IP headers reveals that only one field kept the same meaning and position. That field is the Version field, which

PAYOFF IDEA

The next-generation Internet Protocol will significantly enhance the ability of the Internet in terms of device addressing, router efficiency, and security. Although the actual implementation of IPv6 is still a few years away, most network managers and administrators will eventually be tasked with planning migration strategies that will enable their organizations to move from the current version of the Internet Protocol to the next-generation Internet Protocol, IPv6. Due to this, it is important to obtain an appreciation for the major characteristics of IPv6, which will then serve as a foundation for discussing migration methods that can be considered to take advantage of the enhanced functionality of the next-generation Internet Protocol.

EXHIBIT 1 — Comparing IPv4 and IPv6



is encoded in the first four bits of each header as a binary value, with 0100 used for IPv4 and 0110 for IPv6.

Continuing the comparison of the two headers, note that IPv6 does away with seven IPv4 fields. Those fields include the Type of Service, Identification, Flags, Fragment Offset, Checksum, Options, and Padding. Because headers can be daisy chained and separate headers now identify specific services, the Type of Service field is no longer necessary. Another significant change between IPv4 and IPv6 concerns fragmentation, which enables senders to transmit large packets without worrying about the capabilities of intermediate routers. Under IPv4, fragmentation required the use of Identification, Flags, and Fragment Offset fields. Under IPv6, hosts learn the maximum acceptable segment size through a process referred to as path MTU (maximum transmission unit) discovery. Thus, this enabled the IPv6 designers to remove those three fields from the new header.

Another difference between IPv4 and IPv6 headers involves the removal of the Header Checksum. In an era of fiber backbones it was thought that the advantage obtained from eliminating the processing associated with performing the header checksum at each router was considerably more than the possibility that transmission errors would go undetected. In addition, since the higher layer (transport layer) and lower layer (IEEE 802 networks) perform checksum operations, the risk of undetected error at the network layer adversely affecting operations is minimal. Two more omissions from the IPv4 header are the Options and Padding fields. Both fields are not necessary in IPv6 because the use of

optional headers enables additional functions to be specified as separate entities. Since each header follows a fixed format, there is also no need for a variable Padding field, as was the case under IPv4.

Perhaps the change that obtains the most publicity is the increase in source and destination addresses from 32 bit fields to 128 bit fields. Through the use of 128-bit addressing fields, IPv6 provides the potential to supply unique addresses for every two- and four-footed creature on Earth and still have enough addresses left over to assign a unique address to every past, present, and future appliance. Thus, the extra 96 bit positions virtually ensures that one will not experience another IP address crunch such as the one now being experienced with IPv4.

NEW AND RENAMED IPV6 FIELDS

IPv6 adds three new fields while relabeling and slightly modifying the use of Total Length and Time to Live fields in Ipv4. Concerning the renamed and revised fields, the Total Length field in IPv4 was changed to a Payload Length. This subtle difference is important, as the use of a payload length now specifies the length of the data carried after the header instead of the length of the sum of both the header and data. The second revision represents the recognition of the fact that the Time to Live field under IPv4, which could be specified in seconds, was difficult — if not impossible — to use due to a lack of time-stamping on packets. Instead, the value used in that field was decremented at each router hop as a mechanism to ensure packets did not endlessly flow over the Internet, since they are discarded when the value of that field reaches zero. In recognition of the actual manner by which that field is used, it was renamed the Hop Limit field under IPv6.

The Priority field is four bits wide, enabling 16 possible values. This field enables packets to be distinguished from one another based on their need for processing precedence. Thus, file transfers would be assigned a low priority, while realtime audio or video would be assigned a higher priority.

Under IPv6, priority field values of 0 through 7 are used for traffic that is not adversely affected by backing off in response to network congestion. In comparison, values 8 to 15 are used for traffic that would be adversely affected by backing off when congestion occurs, such as realtime audio packets being transmitted at a constant rate. Exhibit 2 lists the priority values recommended for different types of congestion-controlled traffic.

Priorities 8 through 15 are used for traffic that would be adversely affected by backing off when network congestion occurs. The lowest priority value in this group, 8, should be used for packets one is most willing to discard under congestion conditions. In comparison, the highest priority, 15, should be used for packets one is least willing to have discarded.

EXHIBIT 2 — Recommended Congestion-Controlled Priorities

Priority	Type of Traffic
0	Uncharacterized traffic
1	Filter traffic, such as Netnews
2	Unattended data transfer (i.e., e-mail)
3	Reserved
4	Attended bulk transfer (i.e., FTP, HTTP)
5	Reserved
6	Interactive traffic (i.e., telnet)
7	Internet controlled traffic (i.e., SNMP)

The Flow Label field, also new to IPv6, allows packets that require the same treatment to be identified. For example, a realtime video transmission that consists of a long sequence of packets would more than likely use a Flow Label identifier as well as a high priority value so that all packets that make up the video are treated the same, even if other packets with the same priority arrive at the same time at intermediate routers.

HEADER CHAINS

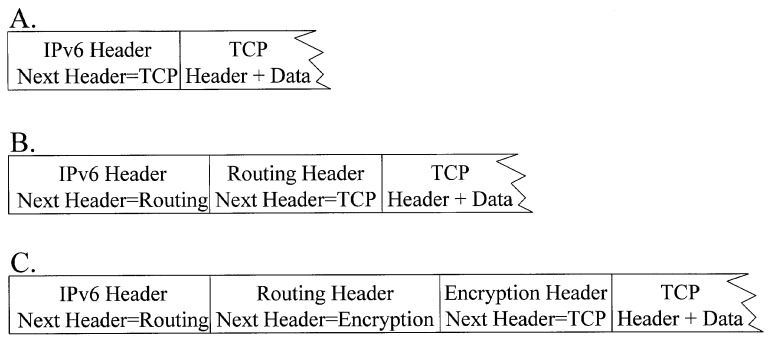
The ability to chain headers is obtained through the use of the IPv6 Next Header field. Currently, the IPv6 specification designates six extension headers. Those headers and a brief description of the functions they perform are listed in Exhibit 3.

To illustrate how the Next Header field in IPv6 is actually used, one can use a few of the headers listed in Exhibit 4 to create a few examples. First, assume that an IPv6 header is followed directly by a TCP header and data, with no optional extension headers. Then, the Next Header field in the IPv6 header would indicate that the TCP header follows as indicated in Exhibit 4A.

EXHIBIT 3 — IPv6 Extension Headers

Extension Header	Description
Hop by hop options	Passes information to all routers in a path
Routing	Defines the route through which a packet flows
Fragment	Provides information that enables destination address to concatenate fragments
Authentication	Verifies the originator
Encrypted security payload	Defines the algorithm and keys necessary to decrypt a previously encrypted payload
Destination options	Defines a generic header that can obtain one or more options identified by options type that can define new extensions on an as-required basis

EXHIBIT 4 — Creating a Daisy Chain of Headers



For a second example, assume that one wants to specify a path or route the packet will follow. To do so, one would add a Routing Header, with the IPv6's Next Header field containing a value that specifies that the Routing Header follows. Then, the Routing Header's Next Header field would contain an appropriate value that specifies that the TCP header follows. This header chain is illustrated in Exhibit 4B.

For a third example, assume one wants to both specify a route for each packet as well as encrypt the payload. To accomplish this, one would change the TCP Header's Next Header field value from the previous example where it indicates that there are no additional headers in the header chain, to a value that serves to identify the Encryption Header as the next header.

Exhibit 4C illustrates the daisy chain of IPv6 headers that would specify that a specific route is to be followed and the information required to decrypt an encrypted payload. Now that one has an appreciation for the general format of the IPv6 header, the use of its header fields, and how headers can be chained to obtain additional functionality, one can focus attention on addressing under IPv6.

ADDRESSING

Under IPv6, there are three types of addresses supported: unicast, multicast, and anycast. The key difference between IPv6 and IPv4 with respect to addressing involves the addition of an anycast type address and the use of 128 bit source and destination addresses.

An anycast address represents a special type of multicast address. Like a multicast address, an anycast address identifies a group of stations that can receive a packet. However, under an anycast addres, only the nearest member of a group receives the packet instead of all members. It is ex-

pected that the use of anycast addressing will facilitate passing packets from network to network as it allows packets to be forwarded to a group of routers without having to know which is the one nearest to the source. Concerning the actual 128 bit address used under IPv6, its expansion by a factor of four over IPv4 resulted in the necessity to introduce methods to facilitate the notation of this expanded address. Thus, the methods by which IPv6 addresses can be noted can be examined.

IPV6 ADDRESS NOTATION

Under IPv4, a 32-bit IP address can be encoded as eight hexadecimal digits. The expansion of the IP address fields to 128 bits results in a requirement to use 32 hexadecimal digits. However, because it is fairly easy to make a mistake that can go undetected by simply entering a long sequence of 32 digits, IPv6 allows each 128 bit address to be represented as eight 16-bit integers separated by colons (:). Thus, under IPv6 notation, one can represent each integer as four hexadecimal digits, enabling a 128 bit address to be encoded or noted as a sequence of eight groups of four hexadecimal digits separated from one another by a colon. An example of a IPv6 address follows:

AB01:0000:OO1A:000C:0000:0000:3A1C:1B1F

Two methods are supported by IPv6 addressing that can be expected to be frequently used by network managers and administrators when configuring network devices. The first method is zero suppression, which allows leading zeros in each of the eight hexadecimal groups to be suppressed. Thus, the application of zero suppression would reduce the previous IPv6 address as follows:

AB01:0:1A:C:0:0:3A1C:1B1E

A second method supported by IPv6 to facilitate the use of 128 bit addresses recognizes that during a migration process, many IPv4 addresses carried within an IPv6 address field will result in a considerable sequence of zero bit positions that cross colon boundaries. This zero density situation can be simplified by the use of a double colon (::), which can replace a single run of consecutive zeros. Thus, one can further simplify the previously zero suppressed IPv6 address as follows:

AB01:0:1A:C::3A1C:1B1E

Note that the use of the double colon can only occur once in an IPv6 address. Otherwise, its use would produce an ambiguous result because there would be no way to tell how many groups of four hexadecimal zeros a double colon represents.

EXHIBIT 5 — Initial IPv6 Address Space Allocation

Address Space Allocation	(binary)	Prefix Fraction of Address Space
Reserved	0000 0000	1/256
Unassigned	0000 0001	1/256
Reserved for NSAP allocation	0000 001	1/128
Reserved for IPX allocation	0000 010	1/128
Unassigned	0000 011	1/128
Unassigned	0000 1	1/32
Unassigned	0001	1/16
Unassigned	001	1/8
Provider-based unicast address	010	1/8
Unassigned	011	1/8
Reserved for geographic-based unicast addresses	100	1/8
Unassigned	101	1/8
Unassigned	110	1/8
Unassigned	1110	1/16
Unassigned	1111 0	1/32
Unassigned	1111 10	1/64
Unassigned	1111 110	1/128
Unassigned	1111 1110 0	1/512
Link local use addresses	1111 1110 10	1/1024
Site local use addresses	1111 1110 11	1/1024
Multicast addresses	1111 1111	1/256

ADDRESS ASSIGNMENTS

With 2^{128} addresses available for assignment, IPv6 designers broke the address space into an initial sequence of 21 address blocks, based on the use of binary address prefixes. As one might surmise, most of the address blocks are either reserved for future use or unassigned because even a small fraction of IPv6 address space is significantly larger than all of the IPv4 address space. Exhibit 5 provides a list of the initial IPv6 address space allocation. Of the initial allocation of IPv6 address space, probably the most important will be the provider-based unicast address. As noted in Exhibit 5, the prefix for this allocated address block is binary 010 and it represents one eighth ($^1/_8$) of the total IPv6 address space. The provider-based unicast address space enables the registry that allocates the address, the Internet service provider (ISP), and the subscriber to be identified. In addition, a subscriber can subdivide his address into a subnetwork and interface or host identifiers similar to the manner by which IPv4 class A through class C addresses can be subdivided into host and network identifiers. The key difference between the two is the fact that an extension to 128 bits enables an IPv6 address to identify organizations that assigned the address to include the registry and ISP. Concerning the registry, in North America, the Internet Network Information Center (Internet NIC) is tasked with distributing IPv4 addresses and can be expected to distribute IPv6 addresses. The European registry is the Network

Coordination Center (NCC) of RIPE, while the APNIC is responsible for distributing addresses for networks in Asian and Pacific countries.

MIGRATION ISSUES

After a considerable amount of deliberation by the Internet community, it was decided that the installed base of approximately 20 million computers using IPv4 would require a dual-stack migration strategy. Instead of one giant cutover sometime in the future, it was recognized that a considerable amount of existing equipment would be incapable of migrating to IPv6. Thus, an IPv6 Internet will be deployed in parallel to IPv4, and all IPv6 hosts will be capable of supporting IPv4. This means that network managers can decide both if and when they should consider upgrading to IPv6. Perhaps the best strategy is, that when in doubt, to obtain equipment capable of operating a dual stack, such as the one shown in Exhibit 5. In addition to operating dual stacks, one must consider one's network's relationship with other networks with respect to the version of IP supported. For example, if an organization migrates to IPv6, but its ISP does not, one will have to encapsulate IPv6 through IPv4 to use the transmission services of the ISP to reach other IPv6 networks. Fortunately, two types of tunneling — configured and automatic — have been proposed to allow IPv6 hosts to reach other IPv6 hosts via IPv4-based networks. Thus, between the use of a dual-stack architecture and configured and automatic tunneling, one will be able to continue to use IPv4 as the commercial use of IPv6 begins, as well as plan for an orderly migration.

RECOMMENDED COURSE OF ACTION

Although the first commercial use of IPv6 is still a few years away, an organization can prepare itself for Ipv6 use by ensuring that acquired hosts, workstations, and routers can be upgraded to support IPv6. In addition, one must consider the fact that the existing Domain Name Server (DNS) will need to be upgraded to support IPv6 addresses, and one must contact the DNS software vendor to determine how and when to implement IPv6 addressing support. By carefully determining the software and possible hardware upgrades, and by keeping abreast of Internet IPv6-related RFCs, one can plan a migration strategy that will allow an organization to benefit from the enhanced router performance afforded by IPv6 addressing.

Gilbert Held is an award-winning author and lecturer. Gil is the author of over 40 books and 300 technical articles. Some of Gil's recent titles include *Data Communications Networking Devices, 4th ed.; Ethernet Networks, 3rd ed.; LAN Performance, 2nd ed.;* and *Working with Network Based Images,* all published by John Wiley & Sons of New York and Chichester, England.

VIRTUAL PRIVATE NETWORKS: SECURE REMOTE ACCESS OVER THE INTERNET

John R. Vacca

INSIDE

Remote User Access over the Internet; Connecting Networks over the Internet; Connecting Computers over the Intranet; User Authentication; Address Management; Data Encryption; Key Management; Multiprotocol Support; Point-to-Point Tunneling Protocol (PPTP); Layer 2 Tunneling Protocol (L2TP); IP Security Protocol (IPSec); Integrated RAS-VPN Clients; Proxy Servers; Information Technology Groups (ITGs); Secure Internet Access; High-Speed Internet Access; RAS Reporting; Internet Usage Chargeback

INTRODUCTION

The components and resources of one network over another network are connected via a Virtual Private Network (VPN). As shown in Exhibit 1, VPNs accomplish this by allowing the user to tunnel through the Internet or another public network in a manner that lets the tunnel participants enjoy the same security and features formerly available only in private networks.

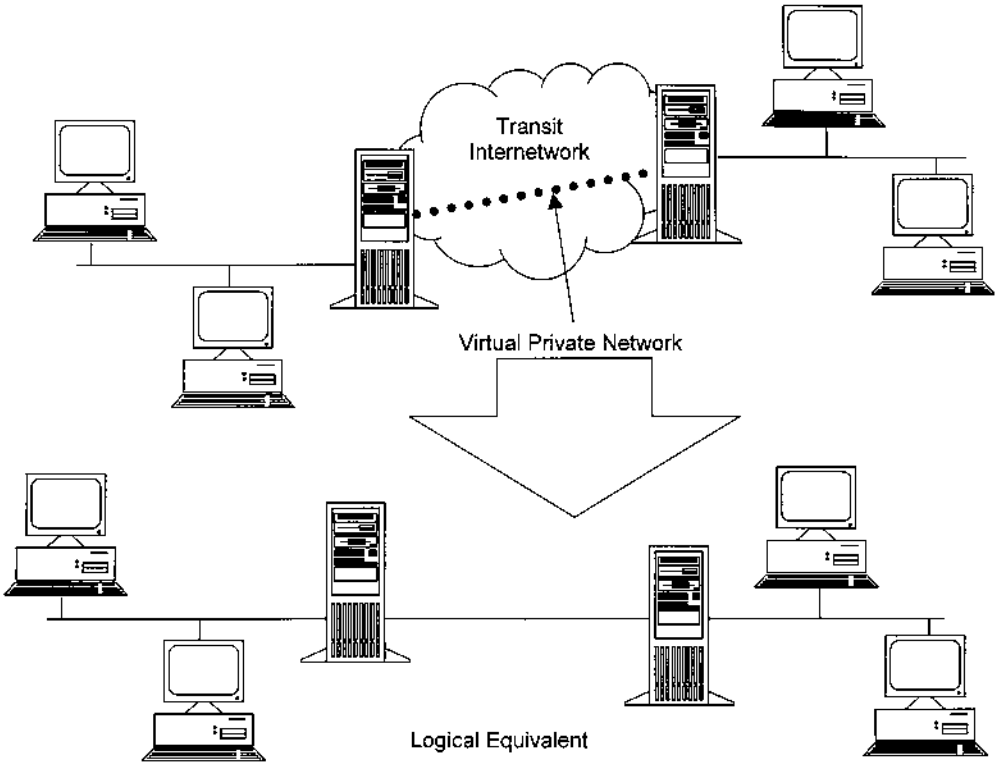
Using the routing infrastructure provided by a public internetwork (such as the Internet), VPNs allow telecommuters, remote employees like salespeople, or even branch offices to connect in a secure fashion to an enterprise server located at the edge of the enterprise local area network (LAN). The VPN is a point-to-point connection between the user's computer and an enterprise server

PAYOFF IDEA

There is no doubt about it: Virtual Private Networks (VPNs) are hot. Secure remote access over the Internet and telecommuting needs are escalating. Distributed enterprise models like extranets are also increasing. The use of VPN technologies by enterprises or corporations require pragmatic, secure Internet remote access solutions that must be easy to use, economical, and flexible enough to meet all of their changing needs. In this article, the reader will learn how enterprises or corporations like Microsoft; UUnet Technologies, Inc., Telco Research, and ATCOM, Inc. are saving more than $28 million every year by using VPNs to do secure remote access over the Internet by their traveling employees and sales reps. The reader will also learn how to make secure Internet remote access information technology (IT) solutions easy to use and easy to manage by telecommunications managers (TMs).

Auerbach Publications

EXHIBIT 1 — Virtual Private Network



from the user's perspective. It also appears as if the data is being sent over a dedicated private link because the nature of the intermediate internetwork is irrelevant to the user. As previously mentioned, while maintaining secure communications, VPN technology also allows an enterprise to connect to branch offices or to other enterprises (extranets) over a public internetwork (such as the Internet). The VPN connection across the Internet logically operates as a wide area network (WAN) link between the sites. In both cases, the secure connection across the internetwork appears to the user as a private network communication (despite the fact that this communication occurs over a public internetwork); hence the name Virtual Private Network.

VPN technology is designed to address issues surrounding the current enterprise trend toward increased telecommuting, widely distributed global operations, and highly interdependent partner operations. Here, workers must be able to connect to central resources and communicate with each other. And, enterprises need to efficiently manage inventories for just-in-time production.

An enterprise must deploy a reliable and scalable remote access solution to provide employees with the ability to connect to enterprise computing resources regardless of their location. Enterprises typically choose one of the following:

- an IT department-driven solution, where an internal information systems department is charged with buying, installing, and maintaining enterprise modem pools and a private network infrastructure
- value-added network (VAN) solutions, where an enterprise pays an outsourced enterprise to buy, install, and maintain modem pools and a telco infrastructure

The optimum solution in terms of cost, reliability, scalability, flexible administration and management, and demand for connections is provided by neither of these traditional solutions. Therefore, it makes sense to find a middle ground where the enterprise either supplements or replaces its current investments in modem pools and its private network infrastructure with a less-expensive solution based on Internet technology. In this manner, the enterprise can focus on its core competencies with the assurance that accessibility will never be compromised, and that the most economical solution will be deployed. The availability of an Internet solution enables a few Internet connections (via Internet service providers, or ISPs) and deployment of several edge-of-network VPN server computers to serve the remote networking needs of thousands or even tens of thousands of remote clients and branch offices, as described next.

VPN Common Uses

The next few subsections of this article describe in more detail common VPN situations.

Secure Remote User Access over the Internet. While maintaining privacy of information, VPNs provide remote access to enterprise resources over the public Internet. A VPN that is used to connect a remote user to an enterprise intranet is shown in Exhibit 2. The user first calls a local ISP Network Access Server (NAS) phone number, rather than making a leased-line, long-distance (or 1-800) call to an enterprise or outsourced NAS. The VPN software creates a virtual private network between the dial-up user and the enterprise VPN server across the Internet using the local connection to the ISP.

Connecting Networks over the Internet. To connect local area networks at remote sites, there exist two methods for using VPNs: using dedicated lines to connect a branch office to an enterprise LAN, or a dial-up line to connect a branch office to an enterprise LAN.

Using Dedicated Lines to Connect a Branch Office to an Enterprise LAN. Both the branch office and the enterprise hub routers can use a local dedicated circuit and local ISP to connect to the Internet, rather than using an expensive long-haul dedicated circuit between the branch office and the enterprise hub. The local ISP connections and the public Internet are used by the VPN software to create a virtual private network between the branch office router and the enterprise hub router.

Using a Dial-Up Line to Connect a Branch Office to an Enterprise LAN. The router at the branch office can call the local ISP, rather than having a router at the branch office make a leased-line, long-distance or (1-800) call to an enterprise or outsourced NAS. Also, in order to create a VPN between the branch office router and the enterprise hub router across the Internet, the VPN software uses the connection to the local ISP as shown in Exhibit 3.

The facilities that connect the branch office and enterprise offices to the Internet are local in both cases. To make a connection, both client/server, and server/server VPN cost savings are largely predicated on the use of a local access phone number. It is recommended that the enterprise hub router that acts as a VPN server be connected to a local ISP with a dedicated line. This VPN server must be listening 24 hours per day for incoming VPN traffic.

Connecting Computers over an Intranet

The departmental data is so sensitive that the department's LAN is physically disconnected from the rest of the enterprise internetwork in some

EXHIBIT 2 — Using a VPN to Connect a Remote Client to a Private LAN

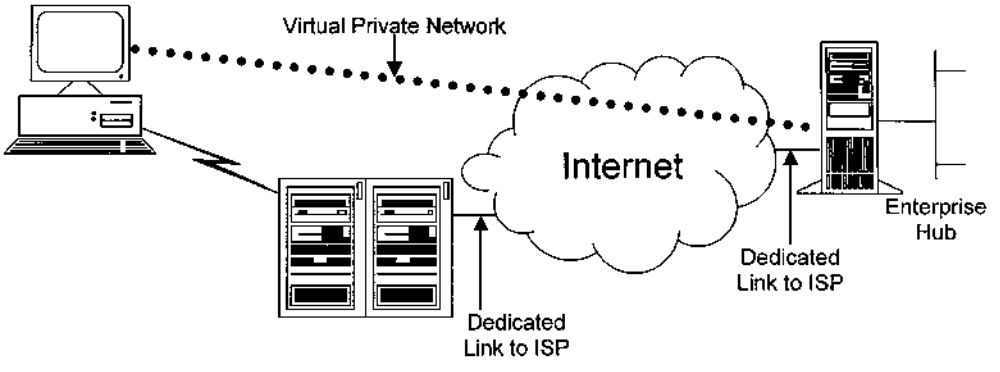
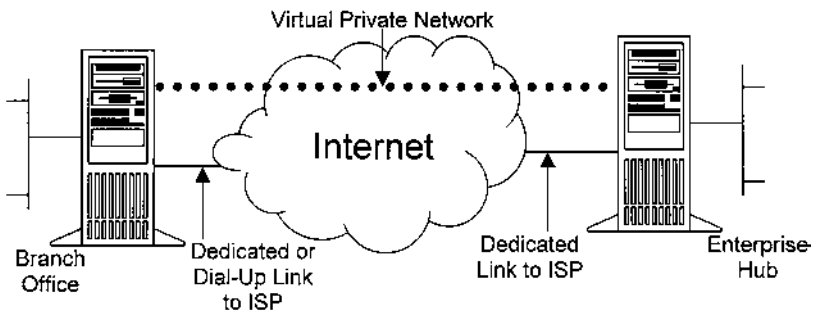


EXHIBIT 3 — Using a VPN to Connect Two Remote Sites



enterprise internetworks. All of this creates information accessibility problems for those users not physically connected to the separate LAN, although the department's confidential information is protected.

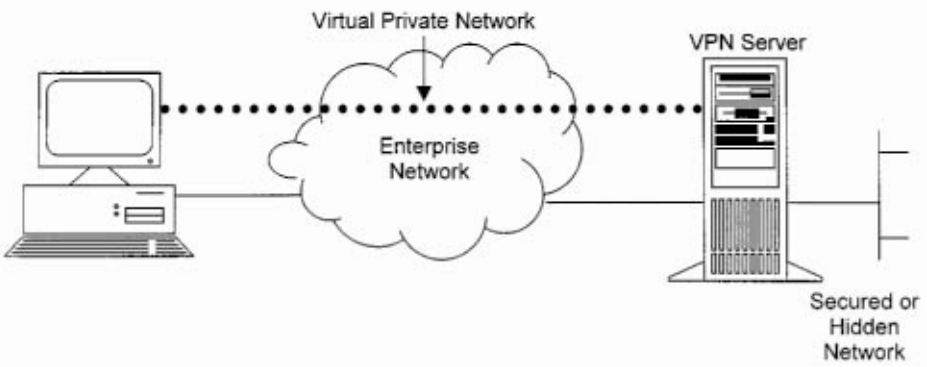
VPNs allow the department's LAN to be separated by a VPN server (see Exhibit 4), but physically connected to the enterprise internetwork. One should note that the VPN server is not acting as a router between the enterprise internetwork and the department LAN. A router would interconnect the two networks, thus allowing everyone access to the sensitive LAN. The network administrator can ensure that only those users on the enterprise internetwork who have appropriate credentials (based on a need-to-know policy within the enterprise) can establish a VPN with the VPN server and gain access to the protected resources of the department by using a VPN. Additionally, all communication across the VPN can be encrypted for data confidentiality. Thus, the department LAN cannot be viewed by those users who do not have the proper credentials.

BASIC VPN REQUIREMENTS

Normally, an enterprise desires to facilitate controlled access to enterprise resources and information when deploying a remote networking solution. In order to easily connect to enterprise local area network (LAN) resources, the solution must allow freedom for authorized remote clients. And, in order to share resources and information (LAN-to-LAN connections), the solution must also allow remote offices to connect to each other. Finally, as the data traverses the public Internet, the solution must ensure the privacy and integrity of data. Also, in the case of sensitive data traversing an enterprise internetwork, the same concerns apply. A VPN solution should therefore provide all of the following at a minimum:

- *Address management:* the solution must assign a client's address on the private net, and must ensure that private addresses are kept private

EXHIBIT 4 — Using a VPN to Connect to Two Computers on the Same LAN



- *Data encryption:* data carried on the public network must be rendered unreadable to unauthorized clients on the network
- *Key management:* the solution must generate and refresh encryption keys for the client and server
- *Multiprotocol support:* the solution must be able to handle common protocols used in the public network; these include Internet Protocol (IP), Internet Packet Exchange (IPX), etc.
- *User authentication:* the solution must verify a user's identity and restrict VPN access to authorized users; in addition, the solution must provide audit and accounting records to show who accessed what information and when

Furthermore, all of these basic requirements are met by an Internet VPN solution based on the Point-to-Point Tunneling Protocol (PPTP) or Layer 2 Tunneling Protocol (L2TP). The solution also takes advantage of the broad availability of the worldwide Internet. Other solutions meet some of these requirements, but remain useful for specific situations, including the new IP Security Protocol (IPSec).

Point-to-Point Tunneling Protocol (PPTP)

PPTP is a Layer 2 protocol that encapsulates PPP frames in IP datagrams for transmission over an IP internetwork, such as the Internet. PPTP can also be used in private LAN-to-LAN networking.

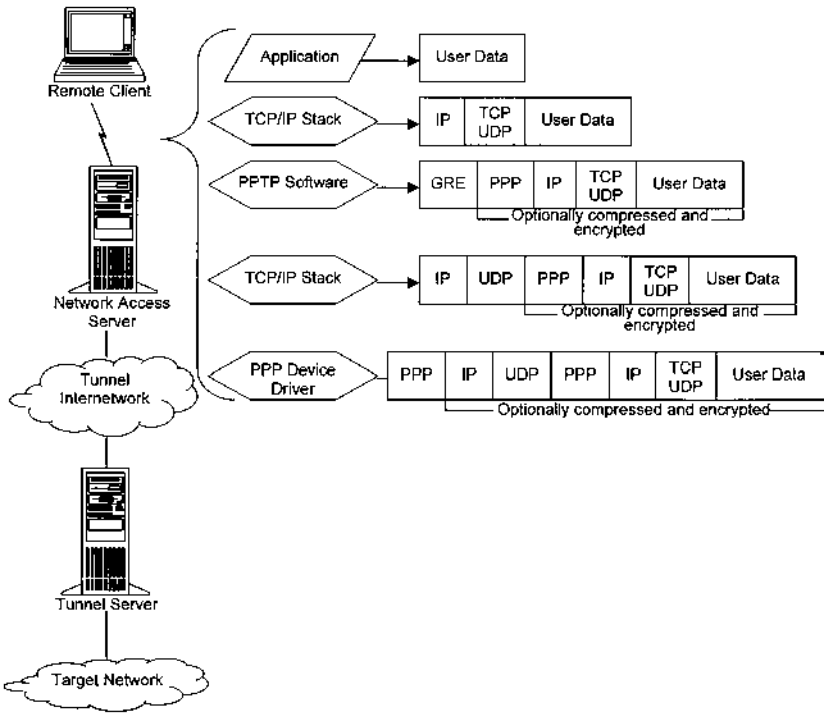
PPTP is documented in the draft RFC, "Point-to-Point Tunneling Protocol."[1] This draft was submitted to the IETF in June 1996 by the member enterprises of the PPTP Forum, including Microsoft Corporation, Ascend Communications, 3Com/Primary Access, ECI Telematics, and U.S. Robotics (now 3Com).

The Point-to-Point Tunneling Protocol (PPTP) uses Generic Routing Encapsulation (GRE) encapsulated Point-to-Point Protocol (PPP) frames for tunneled data and a TCP connection for tunnel maintenance. The payloads of the encapsulated PPP frames can be compressed as well as encrypted. How a PPTP packet is assembled prior to transmission is shown in Exhibit 5. The illustration shows a dial-up client creating a tunnel across an internetwork. The encapsulation for a dial-up client (PPP device driver) is shown in the final frame layout.

Layer 2 Forwarding (L2F)

L2F (a technology proposed by Cisco Systems, Inc.) is a transmission protocol that allows dial-up access servers to frame dial-up traffic in PPP and transmit it over WAN links to an L2F server (a router). The L2F server then unwraps the packets and injects them into the network. Unlike PPTP and L2TP, L2F has no defined client.[2]

EXHIBIT 5 — Construction of a PPTP Packet



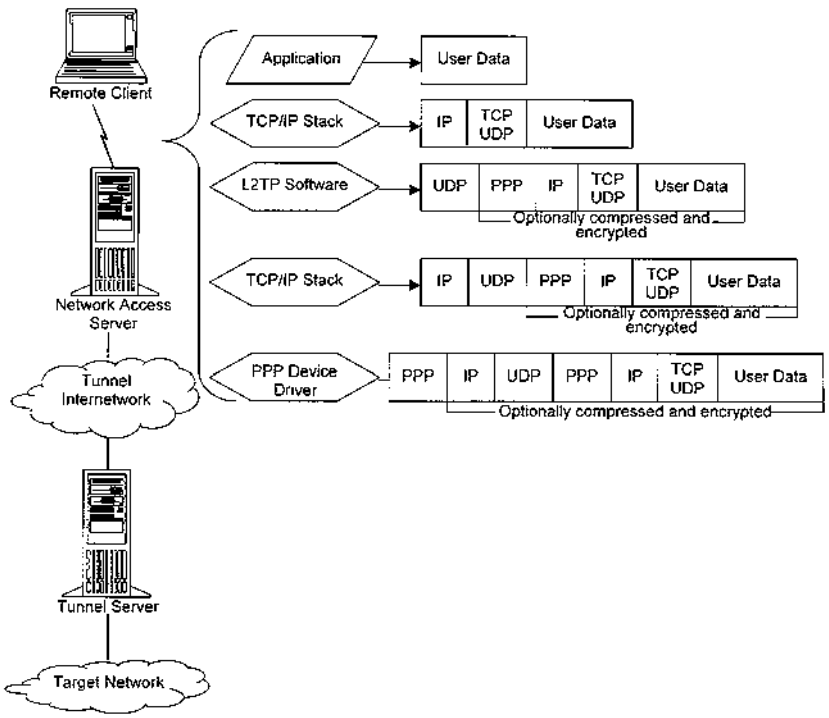
Layer 2 Tunneling Protocol (L2TP)

A combination of PPTP and L2F makes up L2TP. In other words, the best features of PPTP and L2F are incorporated into L2TP.

L2TP is a network protocol that encapsulates PPP frames to be sent over Asynchronous Transfer Mode (ATM), IP, X.25, or Frame Relay networks. L2TP can be used as a tunneling protocol over the Internet when configured to use IP as its datagram transport. Without an IP transport layer, L2TP can also be used directly over various WAN media (such as Frame Relay). L2TP is documented in the draft RFC, Layer 2 Tunneling Protocol "L2TP" (draft-ietf-pppext-l2tp-09.txt). This document was submitted to the IETF in January 1998.

For tunnel maintenance, L2TP over IP internetworks uses UDP and a series of L2TP messages. As the tunneled data, L2TP also uses UDP to send L2TP-encapsulated PPP frames. The payloads of encapsulated PPP frames can be compressed as well as encrypted. How an L2TP packet is assembled prior to transmission is shown in Exhibit 6. A dial-up client

EXHIBIT 6 — Construction of an L2TP Packet



creating a tunnel across an internetwork is shown in the exhibit. The encapsulation for a dial-up client (PPP device driver) is shown in the final frame layout. L2TP over IP is assumed in the encapsulation.

L2TP Compared to PPTP. PPP is used to provide an initial envelope for the data for both PPTP and L2TP. Then, it appends additional headers for transport through the internetwork. The two protocols are very similar. There are differences between PPTP and L2TP, however. For example,

- L2TP provides for header compression. When header compression is enabled, L2TP operates with four bytes of overhead, as compared to six bytes for PPTP.
- L2TP provides for tunnel authentication, while PPTP does not. However, when either protocol is used over IPSec, tunnel authentication is provided by IPSec so that Layer 2 tunnel authentication is not necessary.

- PPTP can only support a single tunnel between endpoints. L2TP allows for the use of multiple tunnels between endpoints. With L2TP, one can create different tunnels for different qualities of service.
- PPTP requires that the internetwork be an IP internetwork. L2TP requires only that the tunnel media provide packet-oriented point-to-point connectivity. L2TP can be used over IP (using UDP), Frame Relay permanent virtual circuits (PVCs), X.25 virtual circuits (VCs), or ATM VCs.

Internet Protocol Security (IPSec) Tunnel Mode

The secured transfer of information across an IP internetwork is supported by IPSec (a Layer 3 protocol standard). Nevertheless, in the context of tunneling protocols, one aspect of IPSec is discussed here. IPSec defines the packet format for an IP over an IP tunnel mode (generally referred to as IPSec Tunnel Mode), in addition to its definition of encryption mechanisms for IP traffic. An IPSec tunnel consists of a tunnel server and tunnel client. These are both configured to use a negotiated encryption mechanism and IPSec tunneling.

For secure transfer across a private or public IP internetwork, IPSec Tunnel Mode uses the negotiated security method (if any) to encapsulate and encrypt entire IP packets. The encrypted payload is then encapsulated again with a plaintext IP header. It is then sent on the internetwork for delivery to the tunnel server. The tunnel server processes and discards the plaintext IP header and then decrypts its contents to retrieve the original payload IP packet. Upon receipt of this datagram, the payload IP packet is then processed normally and routed to its destination on the target network. The following features and limitations are contained within the IPSec Tunnel Mode:

- It is controlled by a security policy: a set of filter-matching rules. This security policy establishes the encryption and tunneling mechanisms available in order of preference and the authentication methods available, also in order of preference. As soon as there is traffic, the two machines perform mutual authentication, and then negotiate the encryption methods to be used. Thereafter, all traffic is encrypted using the negotiated encryption mechanism and then wrapped in a tunnel header.
- It functions at the bottom of the IP stack; therefore, applications and higher-level protocols inherit its behavior.
- It supports IP traffic only.

The remainder of this article discusses VPNs and the use of these technologies by enterprises to do secure remote access (e.g., by traveling employees and sales reps) over the Internet in greater detail.

EASY TO MANAGE AND USE

While squeezing the maximum possible from budget and support staffs, today's enterprises are asking their Information Technology groups (ITGs) to deliver an increasing array of communication and networking services. It appears that the situation is no different at Microsoft Corporation (Redmond, WA). The Microsoft ITG needed to provide secure, Internet-based remote access for its more than 35,000 mobile sales personnel, telecommuters, and consultants around the world.

Microsoft's ITG is currently using and deploying a custom Windows-based remote dial-up and virtual private networking (VPN) solution by using Windows-based clients and enhanced Windows 2000® RAS (Remote Access Server) technology available in the Windows 2000 Option Pack (formerly named Windows NT 5.0). Users are given quick, easy, and low-cost network access. Additional user services are provided with new Windows-based network services from UUnet Technologies, Inc.[3]

Integrated RAS-VPN Clients

According to Microsoft, its ITG has learned that the widespread adoption and use of technology largely depends on how easy and transparent the experience is for the end user. Likewise, Microsoft's ITG has learned not to deploy technologies for which complexity results in an increased support burden on its limited support staff. Microsoft's ITG provided a single client interface with central management to simultaneously make the remote access solution easy to use and manage.

Single Client. A single client is used for both the direct dial-up and virtual private network connections. Users utilize the same client interface for secure transparent access, whether dialing directly to the enterprise network or connecting via a VPN, by using Windows integrated dial-up networking technology (DUN) and Microsoft Connection Manager. In fact, users do not need to concern themselves with which method is employed.

Central Management. Central management is used for remote dial-up and VPN access phone numbers. According to Microsoft, its ITG has found that one of the most common support problems traveling users face is determining and managing local access phone numbers. This problem translates into one of the principal reasons for support calls to Microsoft's user support centers. Using the Connection Manager Administration Kit (CMAK) wizard (which is part of Microsoft's remote access solution), Microsoft's ITG preloads each client PC with an electronic phone book that includes every dial-up remote access phone number for Microsoft's network. The Windows solution also allows phone books to be centrally integrated and managed from a single remote location, and clients to be updated automatically.

WINDOWS COMMUNICATION PLATFORM

In order to provide a flexible and comprehensive network solution, the open extensibility of the Windows 2000 allows Microsoft's ITG to preserve its current hardware network investments while partnering with UUnet Technologies, Inc. According to Microsoft, the Windows platform enabled its ITG to integrate the best-of-breed network services and applications to best meet its client and network administration needs.

High-Speed Internet Access on the Road

Microsoft employees can also connect to high-speed Internet access by plugging into public IPORT[4] jacks in hotels, airports, cafes, and remote locations. The Microsoft ITG integrates the IPORT[5] pay-per-use Internet access features into its custom remote access solution. According to Microsoft, this high-bandwidth, easily available connection helps Microsoft employees be more productive and have a better online experience while on the road.

Secure Internet Access and VPN

Microsoft's ITG, like its counterparts at every enterprise, must ensure that the edge of its network is secure while still providing all employees with the freedom needed to access information worldwide. Microsoft's ITG has also deployed Microsoft Proxy Server to securely separate the LAN from the Internet to meet this need.

To ensure that no intruders compromise the edge of network, the Microsoft Proxy Server firewall capabilities protect Microsoft's network from unauthorized access from the Internet by providing network address translation and dynamic IP-level filtering. Microsoft's ITG uses the powerful caching services in Microsoft Proxy Server to expedite the delivery of information at the same time.

The Proxy Server is able to service subsequent user requests of already-requested information without having to generate additional network traffic by reusing relevant cached information. In addition, in order to operate at peak efficiency with the utmost security, ITG uses Microsoft Proxy Server to enable the Microsoft intranet and remote employees.

RAS Reporting and Internal Usage Chargeback (Billing)

Microsoft pays a substantial amount for remote access fees due to the need to maintain private leased lines and dedicated 800 numbers like many large enterprises with a multitude of branch offices and remote employees. In addition, according to Microsoft, the sheer number of LAN entry points and autonomy afforded its international divisions made centralized accounting and retail reporting for remote access use and roaming users important.

Microsoft's ITG is deploying a VPN solution — bolstered with centralized accounting and reporting of enterprisewide remote access and VPN use — by using Windows 2000, integrated user domain directory, and RADIUS services. As part of this solution, Microsoft is also deploying TRU RADIUS Accountant™ for Windows 2000 from Telco Research.[6]

Furthermore, Microsoft's ITG is also able to generate detailed reporting of remote access and VPN network use for internal cost-accounting purposes while using familiar Windows 2000 management tools by using Telco Research's product. In addition, Microsoft's ITG is able to quickly and easily deploy a turnkey reporting solution built on the intrinsic communication services of Windows 2000 in this manner. According to Microsoft, while maintaining the flexibility to accommodate future change, they receive better security as a result, reduced implementation costs, and enhanced reporting to improve remote access management and chargeback service.

VIP Services: Economical Internet Access And VPN

By working with UUnet Technologies, Inc. (the largest Internet service provider in the world), the Microsoft ITG supplemented its private data network infrastructure and RAS with VPN services. Microsoft's VPN solution is integrated with the UUnet Radius Proxy servers through the Windows 2000 native support for RADIUS under this relationship.

Through the Windows 2000 Remote Access Service integrated RADIUS support, Microsoft's ITG made reliable and secure local access to UUnet Technologies IP network available to all Microsoft mobile employees. This resulted in the delivery of high-quality VPN services over the UUnet Technologies, Inc. infrastructure at a reduced cost. The ITG conservatively estimates that this use of VPN service as an alternative to traditional remote access will save Microsoft more than $7 million per year in remote access fees alone. Additional savings are expected from the elimination of call requests for RAS phone numbers and greatly reduced remote access configuration support.

The ITG utilized the integrated support for RADIUS-based authentication available from the Windows Directory in Windows 2000. This allowed them to retain all existing authentication rights for both Internet and LAN access, avoiding change or redundant replication of directory, and provided for enhanced network security.

According to Microsoft, their ITG was able to instantly extend network access to its more than 50,000 employees in more than 100 countries through its relationship with UUnet Technologies. So that Microsoft employees can access information locally anywhere with reliability guarantees and the support of UUnet, UUnet Technologies' transcontinental backbone provides access throughout North America, Europe, and the Asia–Pacific region.

PLANNING FOR THE FUTURE

Finally, Microsoft's ITG wanted to ensure that its current investment in the remote access infrastructure would not only be able to meet today's needs, but also enable it to make the most of opportunities provided by the digital convergence of network-aware applications in the near future. Evidence of an increased need for higher degrees of client/server network application integration is found in the momentum of Windows 2000 as a platform for IP telephony, media-streaming technologies, and the migration to PBX systems based on Windows 2000.

The flexibility needed to economically address current and future needs of Microsoft's ITG is provided through the use of Windows 2000 as the backbone of the remote access solution. Through partnerships with multiple service providers such as UUnet Technologies, the selection of a Windows-based solution allows ITG the freedom to both centrally manage and incrementally extend the Microsoft direct-dial and VPN infrastructure at a controlled pace and in an open manner.

In order to connect Microsoft subsidiaries, branch offices, and extranet partners securely to the enterprise network over private and public networks, Windows 2000 Routing, RAS, and VPN services — along with tight integration with Microsoft Proxy Server — are already enabling Microsoft's ITG to seamlessly extend its RAS–VPN infrastructure. Furthermore, to meet Microsoft's enterprise needs into the future, the broad application support enjoyed by the Windows communication platform ensures that ITG will continue to have access to a host of rich application services made available by developers and service providers, such as ATCOM, Inc., Telco-Research, and UUnet Technologies, Inc.

CONCLUSION AND SUMMARY

As explained in this article, Windows 2000 native VPN services allow users or enterprises to reliably and securely connect to remote servers, branch offices, or other enterprises over public and private networks. Despite the fact that this communication occurs over a public internetwork in all of these cases, the secure connection appears to the user as a private network communication. Windows VPN technology is designed to address issues surrounding the current enterprise trend toward increased telecommuting and widely distributed global operations, where workers must be able to connect to central resources and where enterprises must be able to efficiently communicate with each other.

This article provided an in-depth discussion of virtual private networking, and described the basic requirements of useful VPN technologies — user authentication, address management, data encryption, key management, and multiprotocol support. It discussed how Layer 2 protocols, specifically PPTP and L2TP, meet these requirements, and how IPSec (a Layer 3 protocol) will meet these requirements in the future.

Every VPN solution needs to address the technological issues cited in the preceding text and provide the flexibility to address enterprise issues like network interoperability, rich application integration, and infrastructure transparency. Enterprise infrastructure decisions need to be made in a manner that empowers client access to local connections and client utilization of the network in a transparent manner to bolster economy and productivity.

Furthermore, escalating remote access and telecommuting needs and an increase in the use of distributed enterprise models like extranets require pragmatic remote access solutions that are easy to use, economical, and flexible enough to meet the changing needs of every enterprise. To support its 50,000+ employees worldwide with best-of-breed remote access and virtual private networking (VPN) services, Microsoft capitalizes on the built-in communication services included in Windows®, integrated VPN firewall and caching support from Microsoft® Proxy Server, and complementary services from partners such as UUnet Technologies, Inc., Telco Research, and ATCOM, Inc.

The remote access infrastructure that Microsoft's Redmond, WA, headquarters uses for its 15,000 HQ employees consists of four dedicated VPN server computers running the Windows 2000 network operating system. Each machine runs three 400-MHz new Pentium III processors, with 204MB of RAM, 3 × 3 GB of local storage, and three 200-Mbps network interface cards.

The UUnet Technologies, Inc. network that supports Microsoft's wholesale remote access and VPN services provides access to one of the largest IP networks in the world. UUnet's backbone infrastructure features a fully meshed network that extends across both the Atlantic and Pacific and includes direct fiber optic connections between Europe, North America, and Asia. UUnet also provides satellite access services for remote areas that lack Internet connections.

Telco Research's TRU RADIUS Accountant™ for Windows 2000 provides Microsoft's ITG with a single source for reporting internal usage and chargeback (billing) information required to control remote access costs. TRU RADIUS easy-to-use applications provide a turnkey analysis of remote access usage and the data needed to proactively manage Microsoft's remote employee costs across its enterprise.

Microsoft's use of UUnet infrastructure to provision its VPN services to its sales force and mobile users is a testament to the quality and reliability of UUnet's multinational IP network. Using Windows 2000 integrated communication services, both UUnet and Microsoft ITG can centrally update Microsoft remote users with the latest local points of presence (POPs) and RAS connection points as soon they become available around the world.

John Vacca is an information technology consultant and internationally known author based in Pomeroy, OH. Since 1982, John has authored 27 books and more than 330 articles in the areas of Internet and intranet security, programming, systems development, rapid application development, multimedia, and the Internet. John was also a configuration management specialist, computer specialist, and the computer security official for the NASA space station program (Freedom) and the International Space Station program from 1988 until his early retirement from NASA in 1995. John can be reached on the Internet at jvacca@hti.net.

Notes

1. Internet draft documents should be considered works in progress. See http://www.ietf.org for copies of Internet drafts.
2. L2F functions in compulsory tunnels only.
3. For more information on UUnet Technologies, Inc. integrated VIP Services for enterprises using Windows, see http://www.uunet.net.
4. For more information on ATCOM Inc. IPORT solutions, see http://www.atcominfo.com/IPORT or http://www.microsoft.com/industry/hospitality/IPORT/default.htm.
5. IPORT is a trademark of ATCOM, Inc.
6. For information on Telco Research's TRU RADIUS Accountant™ for Windows NT, see http://www.telcoresearch.com.

DATA SECURITY MANAGEMENT

APPLETS AND NETWORK SECURITY: A MANAGEMENT OVERVIEW

Al Berg

INSIDE

Applets and the Web, The Security Issue, Java: Secure Applets, Java: Holes and Bugs, Denial-of-Service Threats, JavaScript: A Different Grind, ActiveX: Microsoft's Vision for Distributed Component Computing, An Ounce of Prevention

INTRODUCTION

Applets are small programs that reside on a host computer and are downloaded to a client computer to be executed. This model makes it very easy to distribute and update software. Because the new version of an application only needs to be placed on the server, clients automatically receive and run the updated version the next time they access the application.

The use of applets is possible because of the increasing bandwidth available to Internet and intranet users. The time required to download the programs has been decreasing even as program complexity has been increasing. The development of cross-platform languages such as Sun Microsystems, Inc.'s Java, Microsoft Corp.'s ActiveX, and Netscape Communications Corp.'s JavaScript has made writing applets for many different computers simple — the same exact Java or JavaScript code can be run on a Windows-based PC, a Macintosh, or a UNIX-based system without any porting or recompiling of code. Microsoft is working to port ActiveX to UNIX and Macintosh platforms.

PAYOFF IDEA

Applets, network-based programs that run on client systems, are one of the newest security concerns of network managers. This article describes how applets work, the threats they present, and what security precautions network managers can take to minimize the security exposures presented by applets.

APPLETS AND THE WEB

The World Wide Web is the place that users are most likely to encounter applets today. Java (and to a lesser degree, JavaScript) has be-

Auerbach Publications

come the Webmaster's tool of choice to add interesting effects to Web sites or to deliver applications to end users. Most of the scrolling banners, animated icons, and other special effects found on today's Web pages depend on applets to work. Some Web pages use applets for more substantial applications. For example, MapQuest (http://www.mapquest.com) uses Java and ActiveX to deliver an interactive street atlas of the entire U.S. *Wired* magazine offers a Java-based chat site that, when accessed over the Web, allows users to download an applet that lets them participate in real-time conferencing.

THE SECURITY ISSUE

Every silver lining has a cloud, and applets are no exception. Applets can present a real security hazard for users and network managers. When Web pages use applets, the commands that tell the client's browser to download and execute the applets are embedded in the pages themselves. Users have no way of knowing whether or not the next page that they download will contain an applet, and most of the time, they do not care. The Internet offers an almost limitless source of applets for users to run; however, no one knows who wrote them, whether they were written with malicious intent, or whether they contain bugs that might cause them to crash a user's computer.

Applets and computer viruses have a lot in common. Both applets and viruses are self-replicating code that executes on the user's computer without the user's consent. Some security experts have gone as far as to say that the corporate network manager should prohibit users from running applets at all. However, applets are becoming an increasingly common part of how users interact with the Internet and corporate intranets, so learning to live safely with applets is important for network managers.

WHAT ARE THE RISKS?

According to Princeton University's Safe Internet Programming (SIP) research team, there have been no publicly reported, confirmed cases of security breaches involving Java, though there have been some suspicious events that may have involved Java security problems. The lack of reported cases is no guarantee that there have not been breaches that either were not discovered or were not reported. But it does indicate that breaches are rare.

As Web surfing increasingly becomes a way to spend money, and applets become the vehicle for shopping, attacks on applets will become more and more profitable, increasing the risk. Sun, Netscape, and Microsoft all designed their applet languages with security in mind.

JAVA: SECURE APPLETS

Java programs are developed in a language similar to C++ and stored as source code on a server. When a client, such as a Web browser, requests a page that references a Java program, the source code is retrieved from the server and sent to the browser, where an integrated interpreter translates the source code statements into machine-independent bytecodes, which are executed by a virtual machine implemented in software on the client. This virtual machine is designed to be incapable of operations that might be detrimental to security, thus providing a secure sandbox in which programs can execute without fear of crashing the client system. Java applets loaded over a network are not allowed to:

- Read from files on the client system.
- Write to files on the client system.
- Make any network connections, except to the server from which they were downloaded.
- Start any client-based programs.
- Define native method calls, which would allow an applet to directly access the underlying computer.

Java was designed to make applets inherently secure. Following are some of the underlying language security features offered by Java:

- All of an applet's array references are checked to make sure that programs will not crash because of a reference to an element that does not exist.
- Complex and troublesome pointer variables (found in some vendors' products) that provide direct access to memory locations in the computer do not exist in Java, removing another cause of crashes and potentially malicious code.
- Variables can be declared as unchangeable at runtime to prevent important program parameters from being modified accidentally or intentionally.

JAVA: HOLES AND BUGS

Although Sun has made every effort to make the Java virtual machine unable to run code that will negatively impact the underlying computer, researchers have already found bugs and design flaws that could open the door to malicious applets.

The fact that Sun has licensed Java to various browser vendors adds another level of complexity to the security picture. Not only can security be compromised by a flaw in the Java specification, but the vendor's implementation of the specification may contain its own flaws and bugs.

DENIAL-OF-SERVICE THREATS

Denial-of-service attacks involve causing the client's Web browser to run with degraded performance or crash. Java does not protect the client system from these types of attacks, which can be accomplished simply by putting the client system into a loop to consume processor cycles, creating new process threads until system memory is consumed, or placing locks on critical processes needed by the browser.

Because denial-of-service attacks can be programmed to occur after a time delay, it may be difficult for a user to determine which page the offending applet was downloaded from. If an attacker is subtle and sends an applet that degrades system performance, the user may not know that their computer is under attack, leading to time-consuming and expensive troubleshooting of a nonexistent hardware or software problem.

Java applets are not supposed to be able to establish network connections to machines other than the server they were loaded from. However, there are applets that exploit bugs and design flaws that allow it to establish a back-door communications link to a third machine (other than the client or server). This link could be used to send information that may be of interest to a hacker. Because many ready-to-use Java applets are available for download from the Internet, it would be possible for an attacker to write a useful applet, upload it to a site where Webmasters would download it, and then sit back and wait for information sent by the applet to reach their systems.

WHAT KIND OF INFORMATION CAN THE APPLET SEND BACK?

Due to another implementation problem found in August 1996 by the Safe Internet Programming research team at Princeton University, the possibilities are literally endless. A flaw found in Netscape Navigator versions 3.0 beta 5 and earlier versions, and Microsoft Internet Explorer 3.0 beta 2 and earlier versions, allows applets to gain full read and write access to the files on a Web surfer's machine. This bug means that the attacker can get copies of any files on the machine or replace existing data or program files with hacked versions.

Giving Java applets the ability to connect to an arbitrary host on the network or Internet opens the door to another type of attack. A malicious applet, downloaded to and running on a client inside of a firewalled system, could establish a connection to another host behind the firewall and access files and programs. Because the attacking host is actually inside the secured system, the firewall will not know that the access is actually originating from outside the network.

Another bug found in August 1996 by the Princeton team affects only Microsoft Internet Explorer version 3.0 and allows applets (which are not supposed to be allowed to start processes on the client machine) to execute any DOS command on the client. This allows the applet to delete

or change files or programs or insert new or hacked program code such as viruses or backdoors. Microsoft has issued a patch (available on its Web site at http://www.microsoft.com/ie) to Internet Explorer that corrects the problem.

Princeton's SIP team also found a hole that would allow a malicious application to execute arbitrary strings of machine code, even though the Java virtual machine is only supposed to be able to execute the limited set of Java bytecodes. The problem was fixed in Netscape Navigator 3.0 beta 6 and Microsoft Internet Explorer 3.0 beta 2.

JAVASCRIPT: A DIFFERENT GRIND

Netscape's JavaScript scripting language may be named Java, but it is distinct from Sun's applet platform. JavaScript is Netscape Navigator's built-in scripting language that allows Webmasters to do cross-platform development of applets that control browser events, objects such as tables and forms, and various activities that happen when users click on an object with their mouse.

Like Java, JavaScript runs applications in a virtual machine to prevent them from performing functions that would be detrimental to the operation of the client workstations. Also like Java, there are several flaws in the implementation of the security features of JavaScript. Some of the flaws found in JavaScript include the ability for malicious applets to:

- Obtain users' E-mail addresses from their browser configuration.
- Track the pages that a user visits and mail the results back to the script author.
- Access the client's file system, reading and writing files.

A list of JavaScript bugs and fixes can be found on John LoVerso's Web page at the Open Software Foundation (http://www.osf.org/~ loverso/javascript/).

ActiveX: Microsoft's Vision for Distributed Component Computing. Microsoft's entry in the applet development tool wars, ActiveX, is very different from Java and presents its own set of security challenges. ActiveX is made up of server and client components, including:

- Controls, which are applets that can be embedded in Web pages and executed at the client. Controls can be written in a number of languages, including Visual Basic and Visual C++.
- Documents that provide access to non-HTML content, such as word processing documents or spreadsheets, from a Web browser.
- The Java virtual machine, which allows standard Java applets to run at the client.

- Scripting, which allows the Web developer to control the integration of controls and Java applets on a Web page.
- The server framework, which provides a number of server-side functions such as database access and data security.

Java applets running in an ActiveX environment (e.g., Microsoft's Internet Explorer Web browser) use the same security features and have the same security issues associated with JavaScript. Microsoft offers a Java development environment (i.e., Visual J++) as well as other sandbox languages (i.e., VBScript, based on Visual Basic and JScript, Microsoft's implementation of Netscape's JavaScript) for the development of applications that are limited as to the functions they can perform.

When developers take advantage of ActiveX's ability to integrate programs written in Visual Basic or C++, the virtual machine model of Java no longer applies. In these cases, compiled binaries are transferred from the server to the Web client for execution. These compiled binaries have full access to the underlying computing platform, so there is no reason that the application could not read and write files on the client system, send information from the client to the server (or another machine), or perform a destructive act such as erasing a disk or leaving a virus behind.

USING AUTHENTICODE FOR ACCOUNTABILITY

Microsoft's approach to security for non-Java ActiveX applications is based on the concept of accountability — knowing with certainty the identity of the person or company that wrote a piece of software and that the software was not tampered with by a third party. Microsoft sees the issues related to downloading applets from the Web as similar to those involved in purchasing software; users need to know where the software is coming from and that it is intact. Accountability also means that writers of malicious code could be tracked down and would have to face consequences for their actions.

The mechanism that Microsoft offers to implement this accountability is called Authenticode. Authenticode uses a digital signature attached to each piece of software downloaded from the Internet. The signature is a cryptographic code attached by the software developer to an applet. Developers must enter a private key (known only to them) to sign their application, assuring their identity. The signature also includes an encrypted checksum of the application itself, which allows the client to determine if the applet has changed since the developer released it.

ACTIVEX: THE DOWNSIDE

This approach provides developers and users with access to feature-rich applications, but at a price. If an application destroys information on a user's computer, accountability will not help recover their data or repair

damage done to their business. Once the culprit has been found, bringing them to justice may be difficult because new computer crimes are developing faster than methods for prosecuting them.

Microsoft acknowledges that Authenticode does not guarantee that end users will never download malicious code to their PCs and that it is a first step in the protection of information assets.

Further information on ActiveX can be found on Microsoft's Web site (http://www.microsoft.com/activex) and at the ActiveX Web site run by CNet Technology Corp. (http://www.activex.com).

AN OUNCE OF PREVENTION

So far, this article has discussed problems posed by applets. Following are some steps that can be taken to lessen the exposure faced by users.

Make Sure the Basics Are Covered

Users need to back up their data and programs consistently, and sensitive data should be stored on secure machines. The surest way to avoid applet security problems is to disable support for applet execution at the browser. If the code cannot execute, it cannot do damage.

Of course, the main downside of this approach is that the users will lose the benefits of being able to run applets. Because the ability to run applets is part of the client browser, turning off applets is usually accomplished at the desktop and a knowledgeable user could simply turn applet support back on. Firewall vendors are starting to provide support for filtering out applets, completely or selectively, before they enter the local network.

Users Should Run the Latest Available Versions of Their Web Browsers

Each new version corrects not only functional and feature issues, but security flaws. If an organization is planning to use applets on its Web pages, it is preferable to either write them internally or obtain them from trusted sources. If applets will be downloaded from unknown sources, a technical person with a good understanding of the applet language should review the code to be sure that it does only what it claims to.

Mark LaDue, a researcher at Georgia Tech has a Web page (available at http://www.math.gatech.edu/~mladue/HostileApplets.html) containing a number of hostile applets available for download and testing. Seeing some real applications may help users recognize new problem applets that may be encountered.

CONCLUSION

IS personnel should monitor the Princeton University Safe Internet Programming group's home page (located at http://www.cs.prince-

ton.edu/sip) for the latest information on security flaws and fixes (under News). It is also a good idea to keep an eye on browser vendors' home pages for news of new versions.

Applets offer users and network managers a whole new paradigm for delivering applications to the desktop. Although, like any new technology, applets present a new set of challenges and concerns, their benefits can be enjoyed while their risks can be managed.

Al Berg is the director of Strategic Technologies at NETLAN Inc. in New York, NY. He can be reached via E-mail at: al_berg@netlan.com.

47

Security for Broadband Internet Access Users

James Trulove

High-speed access is becoming increasingly popular for connecting to the Internet and to corporate networks. The term "high-speed" is generally taken to mean transfer speeds above the 56 kbps of analog modems, or the 64 to 128 kbps speeds of ISDN. There are a number of technologies that provide transfer rates from 256 kbps to 1.544 Mbps and beyond. Some offer asymmetrical uplink and downlink speeds that may go as high as 6 Mbps. These high-speed access methods include DSL, cable modems, and wireless point-to-multipoint access.

DSL services include all of the so-called "digital subscriber line" access methods that utilize conventional copper telephone cabling for the physical link from customer premise to central office (CO). The most popular of these methods is ADSL, or asymmetrical digital subscriber line, where an existing POTS (plain old telephone service) dial-up line does double duty by having a higher frequency digital signal multiplexed over the same pair. Filters at the user premise and at the central office tap off the digital signal and send it to the user's PC and the CO router, respectively.

The actual transport of the ADSL data is via ATM, a factor invisible to the user, who is generally using TCP/IP over Ethernet. A key security feature of DSL service is that the transport media (one or two pairs) is exclusive to a single user. In a typical neighborhood of homes or businesses, individual pairs from each premise are, in turn, consolidated into larger cables of many pairs that run eventually to the service provider's CO. As with a conventional telephone line, each user is isolated from other users in the neighborhood. This is inherently more secure than competing high-speed technologies. The logical structure of an ADSL distribution within a neighborhood is shown in Exhibit 47.1A.

Cable modems (CMs) allow a form of high-speed shared access over media used for cable television (CATV) delivery. Standard CATV video channels are delivered over a frequency range from 54 MHz to several hundred megahertz. Cable modems simply use a relatively narrow band of those frequencies that are unused for TV signal delivery. CATV signals are normally delivered through a series of in-line amplifiers and signal splitters to a typical neighborhood cable segment. Along each of these final segments, additional signal splitters (or taps) distribute the CATV signals to users. Adding two-way data distribution to the segment is relatively easy because splitters are inherently two-way devices and no amplifiers are within the segment. However, the uplink signal from users in each segment must be retrieved at the head of the segment and either repeated into the next up-line segment or converted and transported separately.

As shown in Exhibit 47.1B, each neighborhood segment is along a tapped coaxial cable (in most cases) that terminates in a common-equipment cabinet (similar in design to the subscriber-line interface cabinets used in telephone line multiplexing). This cabinet contains the equipment to filter off the data signal from the neighborhood coax segment and transport it back to the cable head-end. Alternative data routing may be provided between the common equipment cabinets and the NOC (network operations center), often over fiber-optic cables. As a matter of fact, these neighborhood distribution cabinets are often used as a transition point for all CATV signals between fiber-optic transmission links and the installed coaxial cable to the users. Several neighborhood segments may terminate in each cabinet. When a neighborhood has been rewired for fiber distribution and cable modem services, the most often outward sign is the appearance of a four-foot high

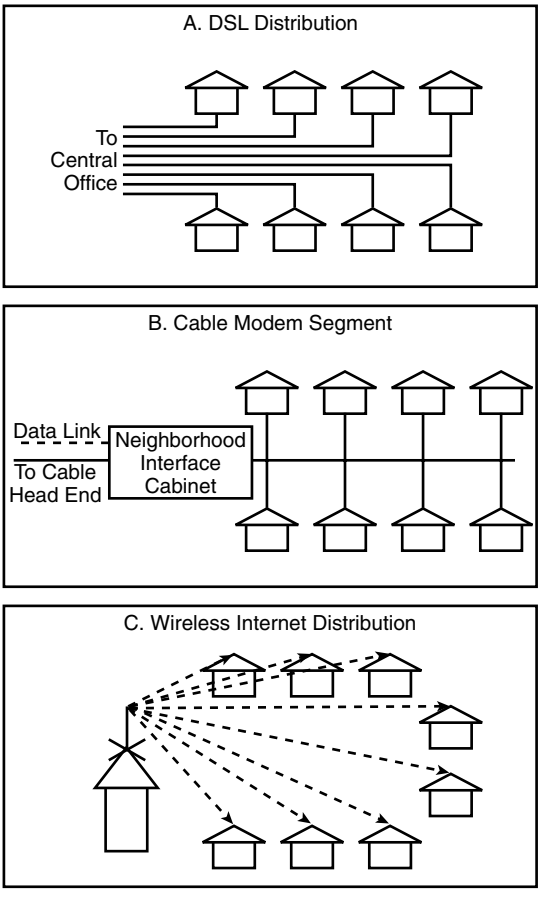


EXHIBIT 47.1 Broadband and wireless Internet access methods.

green or gray metal enclosure. These big green (or gray) boxes are metered and draw electrical power from a local power pole and often have an annoying little light to warn away would-be villains.

Many areas do not have ready availability of cable modem circuits or DSL. Both technologies require the user to be relatively near the corresponding distribution point and both need a certain amount of infrastructure expansion by the service provider. A wireless Internet option exists for high-speed access from users who are in areas that are otherwise unserved. The term "wireless Internet" refers to a variety of noncellular radio services that interconnect users to a central access point, generally with a very high antenna location on a high building, a broadcast tower, or even a mountaintop. Speeds can be quite comparable to the lower ranges of DSL and CM (i.e., 128 to 512 kbps). Subscriber fees are somewhat higher, but still a great value to someone who would otherwise have to deal with low-speed analog dial access.

Wireless Internet is often described as point-to-multipoint operation. This refers to the coverage of several remote sites from a central site, as opposed to point-to-point links that are intended to serve a pair of sites exclusively. As shown in Exhibit 47.1C, remote user sites at homes or businesses are connected by a radio link to a central site. In general, the central site has an omnidirectional antenna (one that covers equally in all radial directions) while remote sites have directional antennas that point at the central antenna.

Wireless Internet users share the frequency spectrum among all the users of a particular service frequency. This means that these remote users must share the available bandwidth as well. As a result, as with the cable modem situation, the actual data throughput depends on how many users are online and active. In addition, all the transmissions are essentially broadcast into the air and can be monitored or intercepted with the proper equipment. Some wireless links include a measure of encryption but the key may still be known to all subscribers to the service.

There are several types of wireless systems permitted in the United States, as with the European Union, Asia, and the rest of the world. Some of these systems permit a single provider to control the rights to a particular frequency allocation. These exclusively licensed systems protect users from unwanted interference from other users and protect the large investment required of the service provider. Other systems utilize a frequency spectrum that is shared and available to all. For example, the 802.11 systems at 2.4 GHz and 5.2 GHz are shared-frequency, nonlicensed systems that can be adapted to point-to-multipoint distribution.

Wireless, or radio-frequency (RF), distribution is subject to all of the same distance limitations, antenna designs, antenna siting, and interference considerations of any RF link. However, in good circumstances, wireless Internet provides a very satisfactory level of performance, one that is comparable to its wired competitors.

Broadband Security Risks

Traditional remote access methods, by their very nature, provide a fair measure of link security. Dial-up analog and dial-on-demand ISDN links have relatively good protection along the path between the user's computer and the access service provider (SP). Likewise, dedicated links to an Internet service provider (ISP) are inherently safe as well, barring any intentional (and unauthorized/illegal) tapping. However, this is not necessarily the case with broadband access methods.

Of the common broadband access methods, cable modems and wireless Internet have inherent security risks because they use shared media for transport. On the other hand, DSL does indeed utilize an exclusive path to the CO but has some more subtle security issues that are shared with the other two methods.

The access-security issue with cable modems is probably the most significant. Most PC users run a version of the Microsoft Windows® operating system, popularly referred to just as Windows. All versions of Windows since Windows 95® have included a feature called peer-to-peer networking. This feature is in addition to the TCP/IP protocol stack that supports Internet-oriented traffic. Microsoft Windows NT® and Windows 2000® clients also support peer-to-peer networking. These personal operating systems share disk, printer, and other resources in a *network neighborhood* utilizing the NetBIOS protocol. NetBIOS is inherently nonroutable although it can be encapsulated within TCP/IP and IPX protocols. A particular network neighborhood is identified by a Workgroup name and, theoretically, devices with different Workgroup names cannot converse.

A standard cable modem is essentially a two-way repeater connected between a user's PC (or local network) and the cable segment. As such, it repeats everything along your segment to your local PC network and everything on your network back out to the cable segment. Thus, all the "private" conversations one might have with one's network-connected printer or other local PCs are available to everyone on the segment. In addition, every TCP/IP packet that goes between one's PC and the Internet is also available for eavesdropping along the cable segment. This is a very serious security risk, at least among those connected to a particular segment. It makes an entire group of cable modem users vulnerable to monitoring, or even intrusion. Specific actions to mitigate this risk are discussed later.

Wireless Internet acts essentially as a shared Ethernet segment, where the segment exists purely in space rather than within a copper medium. It is "ethereal," so to speak. What this means in practice is that every transmission to one user also goes to every authorized (and unauthorized) station within reception range of the central tower. Likewise, a user's transmissions back to the central station are available to anyone who is capable of receiving that user's signal. Fortunately, the user's remote antenna is fairly directional and is not at the great height of the central tower. But someone who is along the path between the two can still pick up the user's signal.

Many wireless Internet systems also operate as a bridge rather than a TCP/IP router, and can pass the NetBIOS protocol used for file and printer sharing. Thus, they may be susceptible to the same type of eavesdropping and intrusion problems of the cable modem, unless they are protected by link encryption.

In addition to the shared-media security issue, broadband security problems are more serious because of the vast communication bandwidth that is available. More than anything else, this makes the broadband user valuable as a potential target. An enormous amount of data can be transferred in a relatively short period of time. If the broadband user operates mail systems or servers, these may be more attractive to someone wanting to use such resources surreptitiously.

Another aspect of broadband service is that it is "always on," rather than being connected on-demand as with dial-up service. This also makes the user a more accessible target. How can a user minimize exposure to these and other broadband security weaknesses?

Increasing Broadband Security

The first security issue to deal with is visibility. Users should immediately take steps to minimize exposure on a shared network. Disabling or hiding processes that advertise services or automatically respond to inquiries effectively shields the user's computer from intruding eyes. Shielding the computer will be of benefit whether the user is using an inherently shared broadband access, such as with cable modems or wireless, or has DSL or dial-up service. Also, remember that the user might be on a shared Ethernet at work or on the road. Hotel systems that offer high-speed access through an Ethernet connection are generally shared networks and thus are subject to all of the potential problems of any shared broadband access.

Shared networks clearly present a greater danger for unauthorized access because the Windows networking protocols can be used to detect and access other computers on the shared medium. However, that does not mean that users are unconditionally safe in using other access methods such as DSL or dial-up. The hidden danger in DSL or dial-up is the fact that the popular peer-to-peer networking protocol, NetBIOS, can be transported over TCP/IP. In fact, a common attack is a probe to the IP port that supports this.

There are some specific steps users can take to disable peer networking if they are a single-PC user. Even if there is more than one PC in the local network behind a broadband modem, users can take action to protect their resources.

Check Vulnerability

Before taking any local-PC security steps, users might want to check on their vulnerabilities to attacks over the Web. This is easy to do and serves as both a motivation to take action and a check on security steps. Two sites are recommended: www.grc.com and www.symantec.com/securitycheck. The grc.com site was created by Steve Gibson for his company, Gibson Research Corp. Users should look for the "shields up" icon to begin the testing. GRC is free to use and does a thorough job of scanning for open ports and hidden servers.

The Symantec URL listed should take the user directly to the testing page. Symantec can also test vulnerabilities in Microsoft Internet Explorer as a result of ActiveX controls. Potentially harmful ActiveX controls can be inadvertently downloaded in the process of viewing a Web page. The controls generally have full access to the computer's file system, and can thus contain viruses or even hidden servers. As is probably known, the Netscape browser does not have these vulnerabilities, although both types of browsers are somewhat vulnerable to Java and JavaScript attacks. According to information on this site, the online free version at Symantec does not have all the test features of the retail version, so users must purchase the tool to get a full test.

These sites will probably convince users to take action. It is truly amazing how a little demonstration can get users serious about security. Remember that this eye-opening experience will not decrease security in any way … it will just decrease a user's false sense of security!

Start by Plugging Holes in Windows

To protect a PC against potential attacks that might compromise personal data or even harm a PC, users will need to change the Windows Networking default configurations. Start by disabling file and printer sharing, or by password-protecting them, if one must use these features. If specific directories must be shared with other users on the local network, share just that particular directory rather than the entire drive. Protect each resource with a unique password. Longer passwords, and passwords that use a combination of upper/lower case, numbers, and allow punctuation, are more secure.

Windows Networking is transported over the NetBIOS protocol, which is inherently unroutable. The advantage to this feature is that any NetBIOS traffic, such as that for printer or file sharing, is blocked at any WAN router. Unfortunately, Windows has the flexibility of encapsulating NetBIOS within TCP/IP packets, which are quite routable. When using IP Networking, users may be inadvertently enabling this behavior. As a matter of fact, it is a little difficult to block. However, there are some steps users can take to isolate their NetBIOS traffic from being routed out over the Internet.

The first step is to block NetBIOS over TCP/IP. To do this in Windows, simply go to the Property dialog for TCP/IP and disable "NetBIOS over TCP/IP." Likewise, disable "Set this protocol to be the default." Now go to bindings and uncheck all of the Windows-oriented applications, such as Microsoft Networking or Microsoft Family Networking.

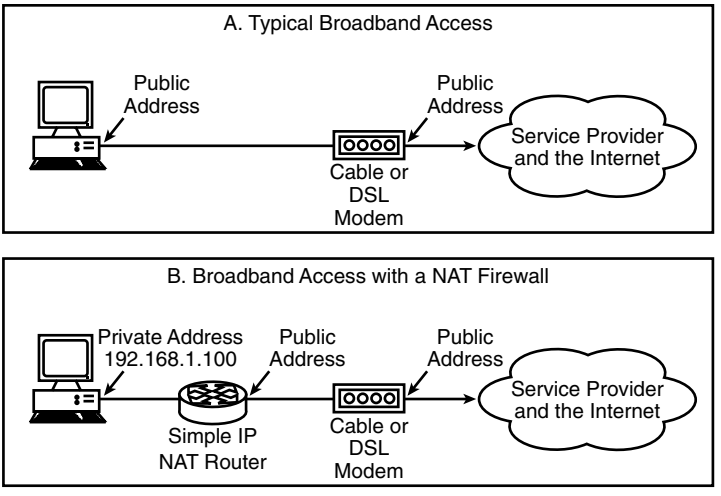


EXHIBIT 47.2 Addition of a NAT firewall for broadband Internet access.

The next step is to give local networking features an alternate path. Do this by adding IPX/SPX compatible protocol from the list in the Network dialog box. After adding IPX/SPX protocol, configure its properties to take up the slack created with TCP/IP. Set it to be the default protocol; check the "enable NetBIOS over IPX/SPX" option, and check the Windows-oriented bindings that were unchecked for TCP/IP. In exiting the dialog, by checking OK, notice that a new protocol has been added, called "NetBIOS support for IPX/SPX compatible Protocol." This added feature allows NetBIOS to be encapsulated over IPX, isolating the protocol from its native mode and from unwanted encapsulation over TCP/IP.

This action provides some additional isolation of the local network's NetBIOS communication because IPX is generally not routed over the user's access device. Be sure that IPX routing, if available, is disabled on the router. This will not usually be a problem with cable modems (which do not route) or with DSL connections because both are primarily used in IP-only networks. At the first IP router link, the IPX will be blocked. If the simple NAT firewall described in the next section is used, IPX will likewise be blocked. However, if ISDN is used for access, or some type of T1 router, check that IPX routing is off.

Now Add a NAT Firewall

Most people do not have the need for a full-fledged firewall. However, a simple routing device that provides network address translation (NAT) can shield internal IP addresses from the outside world while still providing complete access to Internet services. Exhibit 47.2A shows the normal connection provided by a cable or DSL modem. The user PC is assigned a public IP address from the service provider's pool. This address is totally visible to the Internet and available for direct access and, therefore, for direct attacks on all IP ports.

A great deal of security can be provided by masking internal addresses inside a NAT router. This device is truly a router because it connects between two IP subnets, the internal "private" network and the external "public" network. A private network is one with a known private network subnet address, such as 192.168.x.x or 10.x.x.x. These private addresses are nonroutable because Internet Protocol convention allows them to be duplicated at will by anyone who wants to use them. In the example shown in Exhibit 47.2B, the NAT router is inserted between the user's PC (or internal network of PCs) and the existing cable or DSL modem. The NAT router can act as a DHCP (Dynamic Host Control Protocol) server to the internal private network, and it can act as a DHCP client to the service provider's DHCP server. In this manner, dynamic IP address assignment can be accomplished in the same manner as before, but the internal addresses are hidden from external view.

A NAT router is often called a simple firewall because it does the address-translation function of a full-featured firewall. Thus, the NAT router provides a first level of defense. A common attack uses the source IP address of a user's PC and steps through the known and upper IP ports to probe for a response. Certain of

these ports can be used to make an unauthorized access to the user's PC. Although the NAT router hides the PC user's IP address, it too has a valid public IP address that may now be the target of attacks. NAT routers will often respond to port 23 Telnet or port 80 HTTP requests because these ports are used for the router's configuration. The user must change the default passwords on the router, as a minimum; and, if allowable, disable any access to these ports from the Internet side.

Several companies offer simple NAT firewalls for this purpose. In addition, some products are available that combine the NAT function with the cable or DSL modem. For example, LinkSYS provides a choice of NAT routers with a single local Ethernet port or with four switched Ethernet ports. List prices for these devices are less than $200, with much lower street prices.

Install a Personal Firewall

The final step in securing a user's personal environment is to install a personal firewall. The current software environment includes countless user programs and processes that access the Internet. Many of the programs that connect to the Internet are obvious: the e-mail and Web browsers that everyone uses. However, one may be surprised to know that a vast array of other software also makes transmissions over the Internet connection whenever it is active. And if using a cable modem or DSL modem (or router), one's connection is always active if one's PC is on.

For example, Windows 98 has an update feature that regularly connects to Microsoft to check for updates. A virus checker, personal firewall, and even personal finance programs can also regularly check for updates or, in some cases, for advertising material. The Windows update is particularly persistent and can check every five or ten minutes if it is enabled. Advertisements can annoyingly pop up a browser mini-window, even when the browser is not active.

However, the most serious problems arise from the unauthorized access or responses from hidden servers. Chances are that a user has one or more Web server processes running right now. Even the music download services (e.g., MP3) plant servers on PCs. Surprisingly, these are often either hidden or ignored, although they represent a significant security risk. These servers can provide a backdoor into a PC that can be opened without the user's knowledge. In addition, certain viruses operate by planting a stealth server that can be later accessed by an intruder.

A personal firewall will provide a user essential control over all of the Internet accesses that occur to or from his PC. Several products are on the market to provide this function. Two of these are Zone Alarm from Zone Labs (www.zonelabs.com) and Black Ice Defender from Network Ice (www.networkice.com). Other products are available from Symantec and Network Associates. The use of a personal firewall will alert the user to all traffic to or from his broadband modem and allow the user to choose whether he wants that access to occur. After an initial setup period, Internet access will appear perfectly normal, except that unwanted traffic, probes, and accesses will be blocked.

Some of the products alert the user to unwanted attempts to connect to his PC. Zone Alarm, for example, will pop up a small window to advise the user of the attempt, the port and protocol, and the IP address of the attacker. The user can also observe and approve the ability of his applications to access the Internet. After becoming familiar with the behavior of these programs, the user can direct the firewall to always block or allow access. In addition, the user can explicitly block server behavior from particular programs. A log is kept of actions so that the user can review the firewall activities later, whether or not he disables the pop-up alert window.

Thus far, this chapter has concentrated on security for broadband access users. However, after seeing what the personal firewall detects and blocks, users will certainly want to put it on all their computers. Even dial-up connections are at great risk from direct port scanning and NetBIOS/IP attacks. After installation of a personal firewall, it is not unusual to notice probes beginning within the first 30 seconds after connecting. And if one monitors these alerts, one will continue to see such probes blocked over the course of a session. Do not be alarmed. These probes were happening before the firewall was installed, just without the user's knowledge. The personal firewall is now blocking all these attempts before they can do any harm. Broadband users with a consistent public IP address will actually see a dramatic decrease over time in these probes. The intruders do not waste time going where they are unwelcome.

Summary

Broadband access adds significant security risks to a network or a personal computer. The cable modem or DSL connection is normally always active and the bandwidth is very high compared to slower dial-up or ISDN methods. Consequently, these connections make easy targets for intrusion and disruption. Wireless Internet users have similar vulnerabilities, in addition to possible eavesdropping through the airwaves. Cable modem users suffer additional exposure to nonroutable workgroup protocols, such as Windows-native NetBIOS.

Steps should be taken in three areas to help secure PC resources from unwanted intrusions.

1. Eliminate or protect Windows workgroup functions such as file and printer sharing. Change the default passwords and enable IPX encapsulation if these functions are absolutely necessary.
2. Add a simple NAT firewall/router between the access device and PCs. This will screen internal addresses from outside view and eliminate most direct port scans.
3. Install and configure a personal firewall on each connected PC. This will provide control over which applications and programs have access to Internet resources.

48

New Perspectives on VPNs

Keith Pasley, CISSP, CNE

Wide acceptance of security standards in IP and deployment of quality-of-service (QoS) mechanisms like Differentiated Services (DiffServ) and Resource Reservation Protocol (RSVP) within Multi-Protocol Label Switching (MPLS) is increasing the feasibility of virtual private networks (VPNs). VPNs are now considered mainstream; most service providers include some type of VPN service in their offerings, and IT professionals have grown familiar with the technology. Also, with the growth of broadband, more companies are using VPNs for remote access and telecommuting. Specifically, the small-office/home-office market has the largest growth projections according to industry analysts. However, where once lay the promise of IPSec-based VPNs, it is now accepted that IPSec does not solve all remote access VPN problems.

As user experience with VPNs has grown, so have user expectations. Important user experience issues such as latency, delay, legacy application support, and service availability are now effectively dealt with through the use of standard protocols such as MPLS and improved network design. VPN management tools that allow improved control and views of VPN components and users are now being deployed, resulting in increased scalability and lower ongoing operational costs of VPNs. At one time it was accepted that deploying a VPN meant installing "fat"-client software on user desktops, manual configuration of encrypted tunnels, arcane configuration entry into server-side text-based configuration files, intrusive network firewall reconfigurations, minimal access control capability, and a state of mutual mystification due to vendor hype and user confusion over exactly what the VPN could provide in the way of scalability and manageability. New approaches to delivering on the objective of secure yet remote access are evolving, as shown by the adoption of alternatives to that pure layer 3 tunneling VPN protocol, IPSec. User feedback to vendor technology, the high cost of deploying and managing large-scale VPNs, and opportunity cost analysis are helping to evolve these new approaches to encrypting, authenticating, and authorizing remote access into enterprise applications.

Web-Based IP VPN

A granular focus on Web-enabling business applications by user organizations has led to a rethinking of the problem and solution by VPN vendors.

The ubiquitous Web browser is now frequently the "client" of choice for many network security products. The Web-browser-as-client approach solves a lot of the old problems but also introduces new ones. For example, what happens to any residual data left over from a Web VPN session? How is strong authentication performed? How can the remote computer be protected from subversion as an entry point to the internal network while the VPN tunnel is active? Until these questions are answered, Web browser-based VPNs will be limited from completely obsolescing client/server VPNs.

Most Web-based VPN solutions claim to deliver applications, files, and data to authorized users through any standard Web browser. How that is accomplished differs by vendor. A trend toward turnkey appliances is influencing the development of single-purpose, highly optimized and scalable solutions based on both proprietary and open-source software preinstalled on hardware. A three-tiered architecture is used by most of these vendors. This architecture consists of a Web browser, Web server/middleware, and back-end application.

The Web browser serves as the user interface to the target application. The Web server/middleware is the core component that translates the LAN application protocol and application requests into a Web browser-presentable format. Transport Layer Security (TLS) and Secure Socket Layer (SSL) are the common tunneling protocols used. Authentication options include user name and password across TLS/SSL, two-factor tokens such as RSA SecureID, and (rarely) Web browser-based digital certificates. Due to the high business value assigned to e-mail access, resilient hardware design and performance tuning of software to specific hardware is part of the appeal of the appliance approach. Redundant I/O, RAID 1 disk subsystems, redundant power supplies, hot-swappable cooling fans and disk drives, failover/clustering modes, dual processors, and flash memory-based operating systems are features that help ensure access availability. Access control is implemented using common industry-standard authentication protocols such as Remote Access Dial-In User Service (RADIUS, RFC 2138) and Lightweight Directory Access Protocol (LDAP, RFCs 2251–2256).

Applications

E-mail access is the number-one back-end application for this class of VPN. E-mail has become the lifeblood of enterprise operations. Imagine how a business could survive for very long if its e-mail infrastructure were not available. However, most Web-based e-mail systems allow cleartext transmissions of authentication and mail messages by default. A popular Web mail solution is to install a server-side digital certificate and enable TLS/SSL between the user browsers and the Web mail server. The Web mail server would proxy mail messages to the internal mail server. Variations to this include using a mail security appliance (Mail-VPN) that runs a hardened operating system and Web mail reverse proxy. Another alternative is to install the Web mail server on a firewall DMZ. The firewall would handle Web mail authentication and message proxying to and from the Web server on the DMZ. A firewall rule would be configured to only allow the DMZ Web server to connect to the internal mail server using an encrypted tunnel from the DMZ. E-mail gateways such as the McAfee series of e-mail security appliances focus on anti-virus and content inspection with no emphasis on securing the appliance itself from attack. Depending on how the network firewall is configured, this type of solution may be acceptable in certain environments.

On the other end of the spectrum, e-mail infrastructure vendors such as Mirapoint focus on e-mail components such as message store and LDAP directory server, but they offer very little integrated security of the appliance platform or the internal e-mail server. In the middle is the in-house solution, cobbled together using open-source components and cheap hardware with emphasis on low costs over resiliency, security, and manageability. Another class of Web mail security is offered by remote access VPN generalists such as Netilla, Neoteris, and Whale Communications. These vendors rationalize that the issue with IPSec VPNs is not that you cannot build an IPSec VPN tunnel between two IPSec gateways; rather, the issue is in trying to convince the peer IT security group to allow an encrypted tunnel through its firewall. Therefore, these vendors have designed their product architectures to use common Web protocols such as TLS/SSL and PPTP to tunnel to perimeter firewalls, DMZ, or directly to applications on internal networks.

VPN as a Service: MPLS-Based VPNS

Multi-Protocol Label Switching (MPLS) defines a data-link layer service (see Exhibit 48.1) based on an Internet Engineering Task Force specification (RFC 3031). MPLS specification does *not* define encryption or authentication. However, IPSec is a commonly used security protocol to encrypt IP data carried across an MPLS-based network. Similarly, various existing mechanisms can be used for authenticating users of MPLS-based networks. The MPLS specification defines a network architecture and routing protocol that efficiently forwards and allows prioritization of packets containing higher layer protocol data. Its essence is in the use of so-called labels. An MPLS label is a short identifier used to identify a group of packets that is forwarded in the same manner, such as along the same path, or given the same treatment. The MPLS label is inserted into existing protocol headers or can shimmed between protocol headers, depending on the type of device used to forward packets and overall network implementation.

For example, labels can be shimmed between the data-link and network layer headers or they can be encoded in layer 2 headers. The label is then used to route the so-called labeled packets between MPLS nodes. A network node that participates in MPLS network architectures is called a *label switch router* (LSR). The particular treatment of a labeled packet by an LSR is defined through the use of protocols that assign and distribute

EXHIBIT 48.1 MPLS topologies

Intranet/closed group
Simplest
Each site has routing knowledge of all other VPN sites
BGP updates are propagated between provider edge routers
Extranet/overlapping
Access control to prevent unwanted access
Strong authentication
Centralized firewall and Internet access
Use network address translation
Inter-provider
BGP4 updates exchange
Sub-interface for VPNs
Sub-interface for routing updates
Dial-up
Establish L2TP tunnel to virtual network gateway
Authenticate using RADIUS
Virtual routing and forwarding info downloaded as part authentication/authorization
Hub-and-spoke Internet access
Use a sub-interface for Internet
Use a different sub-interface for VPN

labels. Existing protocols have been extended to allow them to distribute MPLS LSP information, such as label distribution using BGP (MPLS-BGP). Also, new protocols have been defined explicitly to distribute LSP information between MPLS peer nodes. For example, one such newly defined protocol is the Label Distribution Protocol (LDP, RFC 3036). The route that a labeled packet traverses is termed a *label switched path* (LSP). In general, the MPLS architecture supports LSPs with different label stack encodings used on different hops. Label stacking defines the hierarchy of labels defining packet treatment for a packet as it traverses an MPLS internetwork. Label stacking occurs when more than one label is used, within a packet, to forward traffic across an MPLS architecture that employs various MPLS node types. For example, a group of network providers can agree to allow MPLS labeled packets to travel between their individual networks and still provide consistent treatment of the packets (i.e., maintain prioritization and LSP). This level of interoperability allows network service providers the ability to deliver true end-to-end service-level guarantees across different network providers and network domains. By using labels, a service provider and organizations can create closed paths that are isolated from other traffic within the service provider's network, providing the same level of security as other private virtual circuit (PVC)-style services such as Frame Relay or ATM.

Because MPLS-VPNs require modifications to a service provider's or organization's network, they are considered network-based VPNs (see Exhibit 48.2). Although there are topology options for deploying MPLS-VPNs down to end users, generally speaking, MPLS-VPNs do not require inclusion of client devices and tunnels usually terminate at the service provider edge router.

From a design perspective, most organizations and service providers want to set up bandwidth commitments through RSVP and use that bandwidth to run VPN tunnels, with MPLS operating within the tunnel. This design allows MPLS-based VPNs to provide guaranteed bandwidth and application quality-of-service features within that guaranteed bandwidth tunnel. In real terms, it is now possible to not only run VPNs but also enterprise resource planning applications, legacy production systems, and company e-mail, video, and voice telephone traffic over a single MPLS-based network infrastructure. Through the use of prioritization schemes within MPLS, such as Resource Reservation Protocol (RSVP), bandwidth can be reserved for specific data flows and applications. For example, highest prioritization can be given to performance-sensitive traffic that has to be delivered with minimal latency and packet loss and requires confirmation of receipt. Examples include voice and live video streaming, videoconferencing, and financial transactions. A second priority level could then be defined to allow traffic that is mission critical yet only requires an enhanced level of performance. Examples include FTP (e.g., CAD files, video clips) and ERP applications. The next highest priority can be assigned to traffic that does not require specific prioritization, such as e-mail and general Web browsing.

A heightened focus on core competencies by companies, now more concerned with improving customer service and reducing cost, has led to an increase in outsourcing of VPN deployment and management. Service

EXHIBIT 48.2 Sample MPLS Equipment Criteria

Hot standby loadsharing of MPLS tunnels
Authentication via RADIUS, TACACS+, AAA
Secure Shell (SSH) access
Secure Copy (SCP)
Multi-level access modes (EXEC, standard, etc.)
ACL support to protect against DoS attacks
Traffic engineering support via RSVP-TE, OSPF-TE, ISIS-TE
Scalability via offering a range of links: 10/100 Mbps Ethernet, Gigabit Ethernet, 10 Gigabit Ethernet, to OC-3c ATM, OC-3c SONET, OC-12c SONET, and OC-48c SONET
Redundant, hot-swappable interface modules
Rapid fault detection and failover
Network layer route redundancy protocols for resiliency; Virtual Router Redundancy Protocol (VRRP, RFC 2338) for layer 3 MPLS-VPN; Virtual Switch Redundancy Protocol (VSRP); and RSTP for layer 2 MPLS-VPN
Multiple queuing methods (e.g., weighted fair queuing, strict priority, etc.)
Rate limiting
Single port can support tens of thousands of tunnels

providers have responded by offering VPNs as a service using the differentiating capability of MPLS as a competitive differentiator. Service providers and large enterprises are typically deploying two VPN alternatives to traditional WAN offerings such as Frame Relay, ATM, or leased line: IPSec-encrypted tunnel VPNs and MPLS-VPNs. Additional flexibility is an added benefit because MPLS-based VPNs come in two flavors: layer 2 and layer 3. This new breed of VPN based on Multi-Protocol Label Switching (RFC 3031) is emerging as the most marketed alternative to traditional pure IP-based VPNs. Both support multicast routing via Internet Group Membership Protocol (IGMP, RFC 2236), which forwards only a single copy of a transmission to only the requesting port. The appeal of MPLS-based VPNs includes their inherent any-to-any reachability across a common data link. Availability of network access is also a concern of secure VPN design. This objective is achieved through the use of route redundancy along with routing protocols that enhance network availability, such as BGP. MPLS-VPNs give users greater control, allowing them to customize the service to accommodate their specific traffic patterns and business requirements. As a result, they can lower their costs by consolidating all of their data communications onto a single WAN platform and prioritizing traffic for specific users and applications. The resulting simplicity of architecture, efficiencies gained by consolidation of network components, and ability to prioritize traffic make MPLS-VPNs a very attractive and scalable option.

Layer 2 MPLS-VPN

Layer 2 MPLS-VPNs, based on the Internet Engineering Task Force's (IETF) Martini draft or Kompella draft, simply emulate layer 2 services such as Frame Relay, ATM, or Ethernet. With the Martini approach, a customer's layer 2 traffic is encapsulated when it reaches the edge of the service provider network, mapped onto a label-switched path, and carried across a network. The Martini draft describes point-to-point VPN services across virtual leased lines (VLLs), transparently connecting multiple subscriber sites together, independent of the protocols used. This technique takes advantage of MPLS label stacking, whereby more than one label is used to forward traffic across an MPLS architecture. Specifically, two labels are used to support layer 2 MPLS-VPNs. One label represents a point-to-point virtual circuit, while the second label represents the tunnel across the network. The current Martini drafts define encapsulations for Ethernet, ATM, Frame Relay, Point-to-Point Protocol, and High-level Data Link Control protocols. The Kompella draft describes another method for simplifying MPLS-VPN setup and management by combining the auto-discovery capability of BGP (to locate VPN sites) with the signaling protocols that use the MPLS labels. The Kompella draft describes how to provide multi-point-to-multi-point VPN services across VLLs, transparently connecting multiple subscriber sites independent of the protocols used. This approach simplifies provisioning of new VPNs. Because the packets contain their own forwarding information (e.g., attributes contained in the packet's label), the amount of forwarding

state information maintained by core routers is independent of the number of layer 2 MPLS-VPNs provisioned over the network. Scalability is thereby enhanced because adding a site to an existing VPN in most cases requires reconfiguring only the service provider edge router connected to the new site.

Layer 2 MPLS-VPNs are transparent, from a user perspective, much in the same way the underlying ATM infrastructure is invisible to Frame Relay users. The customer is still buying Frame Relay or ATM, regardless of how the provider configures the service. Because layer 2 MPLS-VPNs are virtual circuit based, they are as secure as other virtual circuit- or connection-oriented technologies such as ATM. Because layer 2 traffic is carried transparently across an MPLS backbone, information in the original traffic, such as class-of-service markings and VLAN IDs, remains unchanged. Companies that need to transport non-IP traffic (such as legacy IPX or other protocols) may find layer 2 MPLS-VPNs the best solution. Layer 2 MPLS-VPNs also may appeal to corporations that have private addressing schemes or prefer not to share their addressing information with service providers. In a layer 2 MPLS-VPN, the service provider is responsible only for layer 2 connectivity; the customer is responsible for layer 3 connectivity, which includes routing. Privacy of layer 3 routing is implicitly ensured. Once the service provider edge (PE) router provides layer 2 connectivity to its connected customer edge (CE) router in an MPLS-VPN environment, the service provider's job is done. In the case of trouble-shooting, the service provider need only prove that connectivity exists between the PE and CE. From a customer perspective, traditional, pure layer 2 VPNs function in the same way. Therefore, there are few migration issues to deal with on the customer side. Configuring a layer 2 MPLS-VPN is similar in process to configuring a traditional layer 2 VPN. The "last mile" connectivity, Frame Relay, HDLC, and PPP must be provisioned.

In a layer 2 MPLS-VPN environment, customers can run any layer 3 protocol they would like, because the service provider is delivering only layer 2 connectivity.

Most metropolitan area networks using MPLS-VPNs provision these services in layer 2 of the network and offer them over a high-bandwidth pipe. An MPLS-VPN using the layer 3 BGP approach is quite a complex implementation and management task for the average service provider; the layer 2 approach is much simpler and easier to provision.

Layer 3

Layer 3 MPLS-VPNs are also known as IP-enabled or Private-IP VPNs. The difference between layer 2 and layer 3 MPLS-VPNs is that, in layer 3 MPLS-VPNs, the labels are assigned to layer 3 IP traffic flows, whereas layer 2 MPLS-VPNs encode or shim labels between layer 2 and 3 protocol headers. A traffic flow is a portion of traffic, delimited by a start and stop time, that is generated by a particular source or destination networking device. The traffic flow concept is roughly equivalent to the attributes that make up a call or connection. Data associated with traffic flows are aggregate quantities reflecting events that take place in the duration between the start and stop times of the flow. These labels represent unique identifiers and allow for the creation of label switched paths (LSPs) within a layer 3 MPLS-VPN.

Layer 3 VPNs offer a good solution when the customer traffic is wholly IP, customer routing is reasonably simple, and the customer sites are connected to the SP with a variety of layer 2 technologies. In a layer 3 MPLS-VPN environment, internetworking depends on both the service provider and customer using the same routing and layer 3 protocols. Because pure IPSec VPNs require each end of the tunnel to have a unique address, special care must be taken when implementing IPSec VPNs in environments using private IP addressing based on network address translation. Although several vendors provide solutions to this problem, this adds more management complexity in pure IPSec VPNs.

One limitation of layer 2 MPLS-VPNs is the requirement that all connected VPN sites, using the same provider, use the same data-link connectivity. On the other hand, the various sites of a layer 3 MPLS-VPN can connect to the service provider with any supported data-link connectivity. For example, some sites may connect with Frame Relay circuits and others with Ethernet. Because the service provider in a layer 3 MPLS-VPN can also handle IP routing for the customer, the customer edge router need only participate with the provider edge router. This is in contrast to layer 2 MPLS-VPNs, wherein the customer edge router must deal with an unknown number of router peers. The traditional layer 2 problem of $n^*(n-1)/2$ inherent to mesh topologies carries through to layer 2 MPLS-VPNs as well. Prioritization via class of service is available in layer 3 MPLS-VPNs because the provider edge router has visibility into the actual IP data layer. As such, customers can assign priorities to traffic flows, and service providers can then provide a guaranteed service level for those IP traffic flows.

Despite the complexities, service providers can take advantage of layer 3 IP MPLS-VPNs to offer secure differentiated services. For example, due to the use of prioritization protocols such as DiffServ and RSVP, service providers are no longer hindered by business models based on flat-rate pricing or time and distance. MPLS allows them to meet the challenges of improving customer service interaction, offer new differentiated premium services, and establish new revenue streams.

Summary

VPN technology has come a long way since its early beginnings. IPSec is no longer the only standardized option for creating and managing enterprise and service provider VPNs. The Web-based application interface is being leveraged to provide simple, easily deployable, and easily manageable remote access and extranet VPNs. The strategy for use is as a complementary — not replacement — remote access VPN for strategic applications that benefit from Web browser user interfaces. So-called clientless or Web browser-based VPNs are targeted to users who frequently log onto their corporate servers several times a day for e-mails, calendar updates, shared folders, and other collaborative information sharing. Most of these new Web browser-based VPNs use hardware platforms using a three-tiered architecture consisting of a Web browser user interface, reverse proxy function, and reference monitor-like middleware that transforms back-end application protocols into browser-readable format for presentation to end users. Benefits of this new approach include ease of training remote users and elimination of compatibility issues when installing software on remote systems. Drawbacks include lack of support for legacy applications and limited throughput and scalability for large-scale and carrier-class VPNs.

The promise of any-to-any carrier-class and large-enterprise VPNs is being realized as MPLS-VPN standards develop and technology matures. Interservice provider capability allows for the enforcement of true end-to-end quality-of-service (QoS) guarantees across different provider networks. Multi-Protocol Label Switching can be accomplished at two levels: layer 2 for maximum flexibility, low-impact migrations from legacy layer 2 connectivity, and layer 3 for granular service offerings and management of IP VPNs. MPLS allows a service provider to deliver many services using only one network infrastructure. Benefits for service providers include reduced operational costs, greater scalability, faster provisioning of services, and competitive advantage in a commodity-perceived market. Large enterprises benefit from more efficient use of available bandwidth, increased security, and extensible use of existing well-known networking protocols. Users benefit from the increased interoperability among multiple service providers and consistent end-to-end service guarantees as MPLS products improve. In MPLS-based VPNs, confidentiality, or data privacy, is enhanced by the use of labels that provide virtual tunnel separation. Note that encryption is not accounted for in the MPLS specifications. Availability is provided through various routing techniques allowed by the specifications. MPLS only provides for layer 2 data-link integrity. Higher-layer controls should be applied accordingly.

Further Reading

http://www.mplsforum.org/
www.mplsworld.com
http://www.juniper.net/techcenter/techpapers/200012.html
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t5/vpn.htm
http://www.nortelnetworks.com/corporate/technology/mpls/doclib.html
http://advanced.comms.agilent.com/insight/2001–08/
http://www.ericsson.com/datacom/emedia/qoswhite_paper_317.pdf
http://www.riverstonenet.com/technology/whitepapers.shtml
http://www.equipecom.com/whitepapers.html
http://www.convergedigest.com/Bandwidth/mpls.htm
http://www.convergedigest.com/Bandwidth/mpls.htm

49

An Examination of Firewall Architectures

Paul A. Henry, CISSP

> Today, the number-one and number-two (in sales) firewalls use a technique known as stateful packet filtering, or SPF. SPF has the dual advantages of being fast and flexible and this is why it has become so popular. Notice that I didn't even mention security, as this is not the number-one reason people choose these firewalls. Instead, SPF is popular because it is easy to install and doesn't get in the way of business as usual. It is as if you hired a guard for the entry to your building who stood there waving people through as fast as possible.
>
> **— Rik Farrow,**
>
> **World-renowned independent security consultant**
>
> **July 2000, Foreword**
>
> **Tangled Web — Tales of Digital Crime from the Shadows of Cyberspace**

> Firewall customers once had a vote, and voted in favor of transparency, performance and convenience instead of security; nobody should be surprised by the results.
>
> **— From an e-mail conversation with Marcus J. Ranum,**
>
> **"Grandfather of Firewalls," Firewall Wizard Mailing List, October 2000**

Firewall Fundamentals: A Review

The current state of *in*security in which we find ourselves today calls for a careful review of the basics of firewall architectures.

The level of protection that *any* firewall is able to provide in securing a private network when connected to the public Internet is directly related to the architectures chosen for the firewall by the respective vendor. Generally speaking, most commercially available firewalls utilize one or more of the following firewall architectures:

- Static packet filter
- Dynamic (stateful) packet filter
- Circuit-level gateway
- Application-level gateway (proxy)
- Stateful inspection
- Cutoff proxy
- Air gap

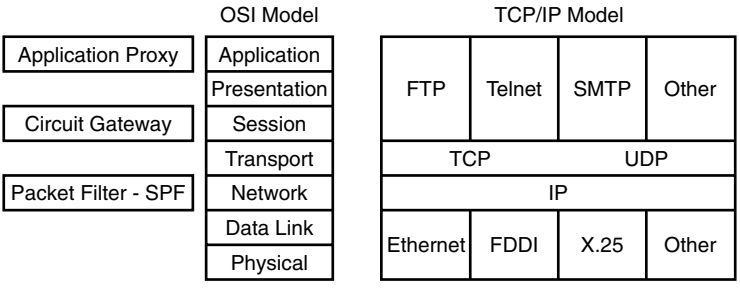


EXHIBIT 49.1 Firewall architectures.

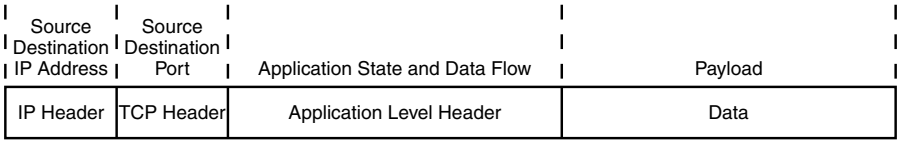


EXHIBIT 49.2 IP packet structure.

Network Security: A Matter of Balance

Network security is simply the proper balance of trust and performance.

All firewalls rely on the inspection of information generated by protocols that function at various layers of the OSI (Open Systems Interconnection) model. Knowing the OSI layer at which a firewall operates is one of the keys to understanding the different types of firewall architectures.

- Generally speaking, the higher up the OSI layer the architecture goes to examine the information within the packet, the more processor cycles the architecture consumes.
- The higher up in the OSI layer at which an architecture examines packets, the greater the level of protection the architecture provides because more information is available upon which to base decisions.

Historically, there had always been a recognized trade-off in firewalls between the level of trust afforded and speed (throughput). Faster processors and the performance advantages of symmetric multi-processing (SMP) have narrowed the performance gap between the traditional fast packet filters and high overhead-consuming proxy firewalls.

One of the most important factors in any successful firewall deployment is *who* makes the trust/performance decisions: (1) the firewall vendor, by limiting the administrator's choices of architectures, or (2) the administrator, in a robust firewall product that provides for multiple firewall architectures.

In examining the firewall architectures in Exhibit 49.1, looking within the IP packet, the most important fields are (see Exhibits 49.2 and 49.3):

- IP Header
- TCP Header
- Application-Level Header
- Data/payload Header

Static Packet Filter

The packet-filtering firewall is one of the oldest firewall architectures. A static packet filter operates at the network layer or OSI layer 3 (see Exhibit 49.4).

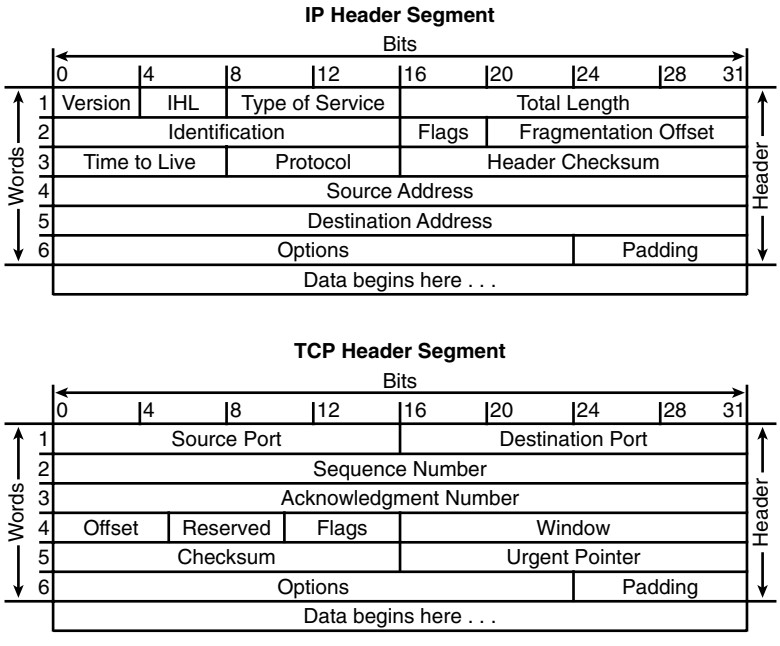


EXHIBIT 49.3 IP header segment versus TCP header segment.

The decision to accept or deny a packet is based upon an examination of specific fields within the packet's IP and protocol headers (see Exhibit 49.5):

- Source address
- Destination address
- Application or protocol
- Source port number
- Destination port number

Before forwarding a packet, the firewall compares the IP Header and TCP Header against a user-defined table — rule base — containing the rules that dictate whether the firewall should deny or permit packets to pass. The rules are scanned in sequential order until the packet filter finds a specific rule that matches the criteria specified in the packet-filtering rule. If the packet filter does not find a rule that matches the packet, then it imposes a default rule. The default rule explicitly defined in the firewall's table *typically* instructs the firewall to drop a packet that meets none of the other rules.

There are two schools of thought on the default rule used with the packet filter: (1) ease of use and (2) security first. *Ease of use* proponents prefer a default *allow all* rule that permits all traffic unless it is explicitly denied by a prior rule. *Security first* proponents prefer a default *deny all* rule that denies all traffic unless explicitly allowed by a prior rule.

Within the static packet-filter rules database, the administrator can define rules that determine which packets are accepted and which packets are denied. The IP Header information allows the administrator to write rules that can deny or permit packets to and from a specific IP address or range of IP addresses. The TCP Header information allows the administrator to write service-specific rules, that is, allow or deny packets to or from ports related to specific services.

The administrator can write rules that allow certain services such as HTTP from any IP address to view the Web pages on the protected Web server. The administrator can also write rules that block a certain IP address or entire ranges of addresses from using the HTTP service and viewing the Web pages on the protected server. In the same respect, the administrator can write rules that allow certain services such as SMTP from a trusted IP address or range of IP addresses to access files on the protected mail server. The administrator could also

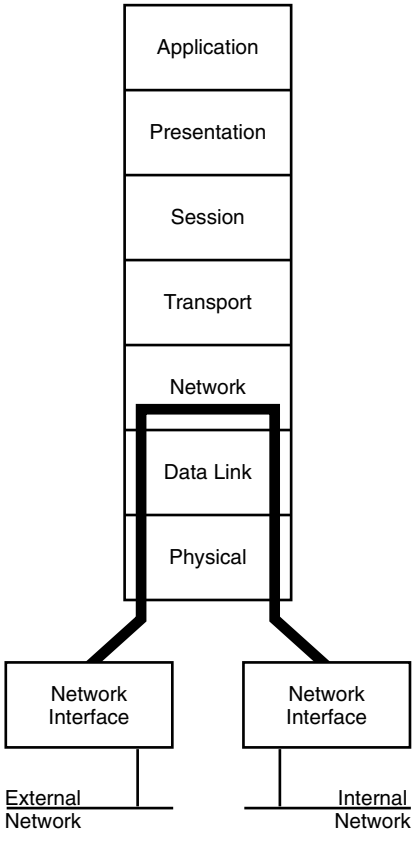


EXHIBIT 49.4 Static packet filter operating at the network layer.

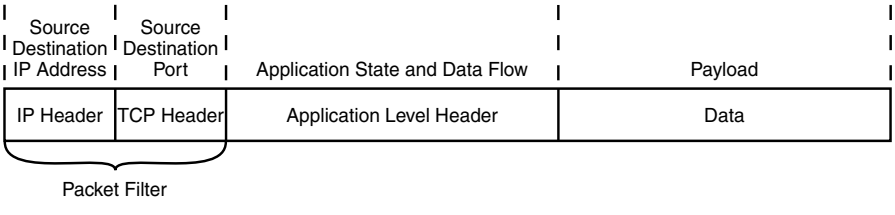


EXHIBIT 49.5 Static packet filter IP packet structure.

write rules that block access for certain IP addresses or entire ranges of addresses to access the protected FTP server.

The configuration of packet-filter rules can be difficult because the rules are examined in sequential order. Great care must be taken in the order in which packet-filtering rules are entered into the rule base. Even if the administrator manages to create effective rules in the proper order of precedence, a packet filter has one inherent limitation:

> A packet filter only examines data in the IP Header and TCP Header; it cannot know the difference between a real and a forged address. If an address is present and meets the packet-filter rules along with the other rule criteria, the packet will be allowed to pass.

Suppose the administrator took the precaution to create a rule that instructed the packet filter to drop any incoming packets with unknown source addresses. This packet-filtering rule would make it more difficult, but

not impossible, for a hacker to access at least some trusted servers with IP addresses. The hacker could simply substitute the actual source address on a malicious packet with the source address of a known trusted client. This common form of attack is called *IP address spoofing*. This form of attack is very effective against a packet filter. The CERT Coordination Center has received numerous reports of IP spoofing attacks, many of which resulted in successful network intrusions. Although the performance of a packet filter can be attractive, this architecture alone is generally not secure enough to keep out hackers determined to gain access to the protected network.

Equally important is what the static packet filter does *not* examine. Remember that in the static packet filter, only specific protocol headers are examined: (1) Source–Destination IP Address and (2) Source–Destination Port numbers (services). Hence, a hacker can hide malicious commands or data in unexamined headers. Further, because the static packet filter does not inspect the packet payload, the hacker has the opportunity to hide malicious commands or data within the packet's payload. This attack methodology is often referred to as a *covert channel attack* and is becoming more popular.

Finally, the static packet filter is *not state aware*. Simply put, the administrator must configure rules for both sides of the conversation to a protected server. To allow access to a protected Web server, the administrator must create a rule that allows both the inbound request from the remote client as well as the outbound response from the protected Web server. Of further consideration is that many services such as FTP and e-mail servers in operation today require the use of dynamically allocated ports for responses, so an administrator of a static packet-filtering-based firewall has little choice but to open up an entire range of ports with static packet-filtering rules.

Static packet filter considerations include:

- Pros:
  - — Low impact on network performance
  - — Low cost, now included with many operating systems
- Cons:
  - — Operates only at network layer and therefore only examines IP and TCP Headers
  - — Unaware of packet payload; offers low level of security
  - — Lacks state awareness; may require numerous ports be left open to facilitate services that use dynamically allocated ports
  - — Susceptible to IP spoofing
  - — Difficult to create rules (order of precedence)
  - — Only provides for a low level of protection

Dynamic (Stateful) Packet Filter

The dynamic (stateful) packet filter is the next step in the evolution of the static packet filter. As such, it shares many of the inherent limitations of the static packet filter with one important difference: *state awareness*.

The typical dynamic packet filter, like the static packet filter, operates at the network layer or OSI layer 3. An advanced dynamic packet filter may operate up into the transport layer — OSI layer 4 (see Exhibit 49.6) — to collect additional state information.

Most often, the decision to accept or deny a packet is based on examination of the packet's IP and Protocol Headers:

- Source address
- Destination address
- Application or protocol
- Source port number
- Destination port number

In simplest terms, the typical dynamic packet filter is *aware* of the difference between a new and an established connection. Once a connection is established, it is entered into a table that typically resides in RAM. Subsequent packets are compared to this table in RAM, most often by software running at the operating system (OS) kernel level. When the packet is found to be an existing connection, it is allowed to pass without any further

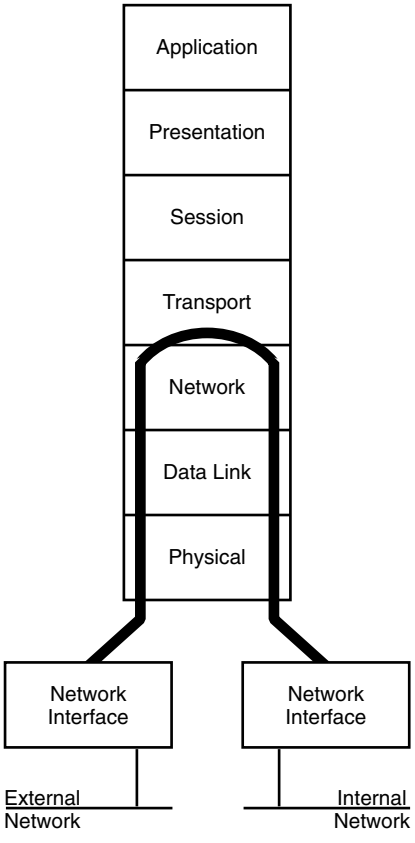


EXHIBIT 49.6 Advanced dynamic packet filter operating at the transport layer.

inspection. By avoiding having to parse the packet-filter rule base for each and every packet that enters the firewall and by performing this already- established connection table test at the kernel level in RAM, the dynamic packet filter enables a measurable performance increase over a static packet filter.

There are two primary differences in dynamic packet filters found among firewall vendors:

1. Support of SMP
2. Connection establishment

In writing the firewall application to fully support SMP, the firewall vendor is afforded up to a 30 percent increase in dynamic packet filter performance for each additional processor in operation. Unfortunately, many implementations of dynamic packet filters in current firewall offerings operate as a single-threaded process, which simply cannot take advantage of the benefits of SMP. Most often to overcome the performance limitation of their single-threaded process, these vendors require powerful and expensive RISC processor-based servers to attain acceptable levels of performance. As available processor power has increased and multi-processor servers have become widely utilized, this single-threaded limitation has become much more visible. For example, vendor *A* running on an expensive RISC-based server offers only 150 Mbps dynamic packet filter throughput, while vendor *B* running on an inexpensive off-the-shelf Intel multi-processor server can attain dynamic packet filtering throughputs of above 600 Mbps.

Almost every vendor has its own proprietary methodology for building the connection table; but beyond the issues discussed above, the basic operation of the dynamic packet filter for the most part is essentially the same.

In an effort to overcome the performance limitations imposed by their single-threaded, process-based dynamic packet filters, some vendors have taken dangerous shortcuts when establishing connections at the firewall. RFC guidelines recommend following the three-way handshake to establish a connection at the firewall.

One popular vendor will open a new connection upon receipt of a single SYN packet, totally ignoring RFC recommendations. In effect, this exposes the servers behind the firewall to single-packet attacks from spoofed IP addresses.

Hackers gain great advantage from anonymity. A hacker can be much more aggressive in mounting attacks if he can remain hidden. Similar to the example in the examination of a static packet filter, suppose the administrator took the precaution to create a rule that instructed the packet filter to drop any incoming packets with unknown source addresses. This packet-filtering rule would make it more difficult, but, again, not impossible for a hacker to access at least some trusted servers with IP addresses. The hacker could simply substitute the actual source address on a malicious packet with the source address of a known trusted client. In this attack methodology, the hacker assumes the IP address of the trusted host and must communicate through the three-way handshake to establish the connection before mounting an assault. This provides additional traffic that can be used to trace back to the hacker.

When the firewall vendor fails to follow RFC recommendations in the establishment of the connection and opens a connection without the three-way handshake, the hacker can simply spoof the trusted host address and fire any of the many well-known single-packet attacks at the firewall, or servers protected by the firewall, while maintaining complete anonymity. One presumes that administrators are unaware that their popular firewall products operate in this manner; otherwise, it would be surprising that so many have found this practice acceptable following the many historical well-known single-packet attacks like LAND, Ping of Death, and Tear Drop that have plagued administrators in the past.

Dynamic packet filter considerations include:

- Pros:
  - — Lowest impact of all examined architectures on network performance when designed to be fully SMP-compliant
  - — Low cost, now included with some operating systems
  - — State awareness provides measurable performance benefit
- Cons:
  - — Operates only at network layer, and therefore only examines IP and TCP Headers
  - — Unaware of packet payload, offers low level of security
  - — Susceptible to IP spoofing
  - — Difficult to create rules (order of precedence)
  - — Can introduce additional risk if connections can be established without following the RFC-recommended three-way handshake
  - — Only provides for a low level of protection

Circuit-Level Gateway

The circuit-level gateway operates at the session layer — OSI layer 5 (see Exhibit 49.7). In many respects, a circuit-level gateway is simply an extension of a packet filter in that it typically performs basic packet filter operations and then adds verification of proper handshaking and the legitimacy of the sequence numbers used to establish the connection.

The circuit-level gateway examines and validates TCP and User Datagram Protocol (UDP) sessions before opening a connection, or circuit, through the firewall. Hence, the circuit-level gateway has more data to act upon than a standard static or dynamic packet filter.

Most often, the decision to accept or deny a packet is based upon examining the packet's IP and TCP Headers (see Exhibit 49.8):

- Source address
- Destination address
- Application or protocol
- Source port number
- Destination port number
- Handshaking and sequence numbers

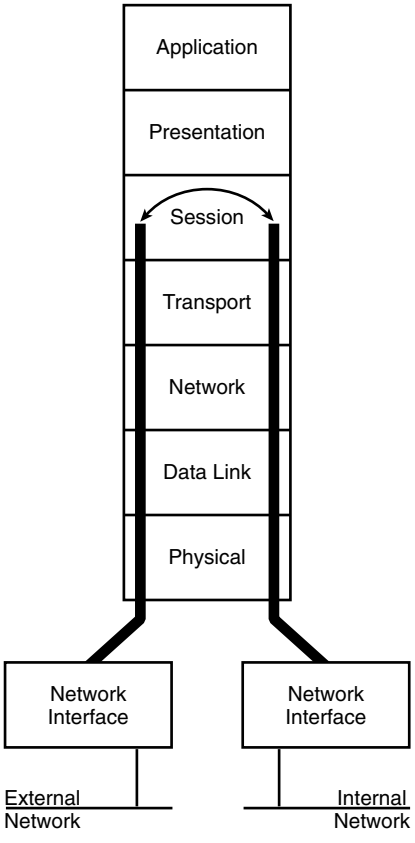


EXHIBIT 49.7 Circuit-level gateway operating at the session layer.

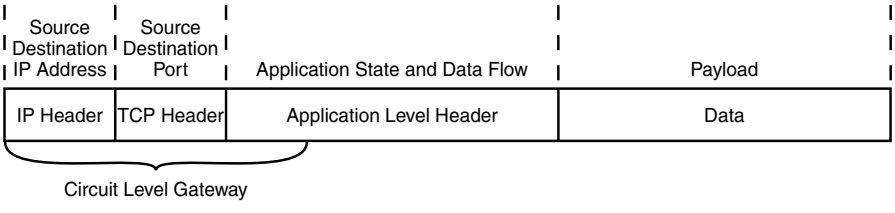


EXHIBIT 49.8 Circuit-level gateway IP packet structure.

Similar to a packet filter, before forwarding the packet, a circuit-level gateway compares the IP Header and TCP Header against a user-defined table containing the rules that dictate whether the firewall should deny or permit packets to pass. The circuit-level gateway then determines that a requested session is legitimate only if the SYN flags, ACK flags, and sequence numbers involved in the TCP handshaking between the trusted client and the untrusted host are logical.

If the session is legitimate, the packet-filter rules are scanned until one is found that agrees with the information in a packet's full association. If the packet filter does not find a rule that applies to the packet, then it imposes a default rule. The default rule explicitly defined in the firewall's table *typically* instructs the firewall to drop a packet that meets none of the other rules.

The circuit-level gateway is literally a step up from a packet filter in the level of security it provides. Further, like a packet filter operating at a low level in the OSI model, it has little impact on network performance. However, once a circuit-level gateway establishes a connection, any application can run across that connection because a circuit-level gateway filters packets only at the session and network layers of the OSI model. In other words, a circuit-level gateway cannot examine the data content of the packets it relays between a trusted network and an untrusted network. The potential exists to slip harmful packets through a circuit-level gateway to a server behind the firewall.

Circuit-level gateway considerations include:

- Pros:
  - — Low to moderate impact on network performance
  - — Breaks direct connection to server behind firewall
  - — Higher level of security than a static or dynamic (stateful) packet filter
- Cons:
  - — Shares many of the same negative issues associated with packet filters
  - — Allows any data to simply pass through the connection
  - — Only provides for a low to moderate level of security

Application-Level Gateway

Like a circuit-level gateway, an application-level gateway intercepts incoming and outgoing packets, runs proxies that copy and forward information across the gateway, and functions as a proxy server, preventing any direct connection between a trusted server or client and an untrusted host. The proxies that an application-level gateway runs often differ in two important ways from the circuit-level gateway:

1. The proxies are application specific.
2. The proxies examine the entire packet and can filter packets at the application layer of the OSI model (see Exhibit 49.9).

Unlike the circuit-level gateway, the application-level gateway accepts only packets generated by services they are designed to copy, forward, and filter. For example, only an HTTP proxy can copy, forward, and filter HTTP traffic. If a network relies only on an application-level gateway, incoming and outgoing packets cannot access services for which there is no proxy. For example, if an application-level gateway ran FTP and HTTP proxies, only packets generated by these services could pass through the firewall. All other services would be blocked.

The application-level gateway runs proxies that examine and filter individual packets, rather than simply copying them and recklessly forwarding them across the gateway. Application-specific proxies check each packet that passes through the gateway, verifying the contents of the packet up through the application layer (layer 7) of the OSI model. These proxies can filter on particular information or specific individual commands in the application protocols the proxies are designed to copy, forward, and filter. As an example, an FTP application-level gateway can filter on dozens of commands to allow a high degree of granularity on the permissions of specific users of the protected FTP service.

Current-technology application-level gateways are often referred to as *strong application proxies*. A strong application proxy extends the level of security afforded by the application-level gateway. Instead of copying the entire datagram on behalf of the user, a strong application proxy actually creates a brand-new empty datagram inside the firewall. Only those commands and data found acceptable to the strong application proxy are copied from the original datagram outside the firewall to the new datagram inside the firewall. Then, and only then, is this new datagram forwarded to the protected server behind the firewall. By employing this methodology, the strong application proxy can mitigate the risk of an entire class of covert channel attacks.

An application-level gateway filters information at a higher OSI layer than the common static or dynamic packet filter, and most automatically create any necessary packet-filtering rules, usually making them easier to configure than traditional packet filters.

By facilitating the inspection of the complete packet, the application-level gateway is one of the most secure firewall architectures available. However, historically some vendors (usually those that market stateful inspec-

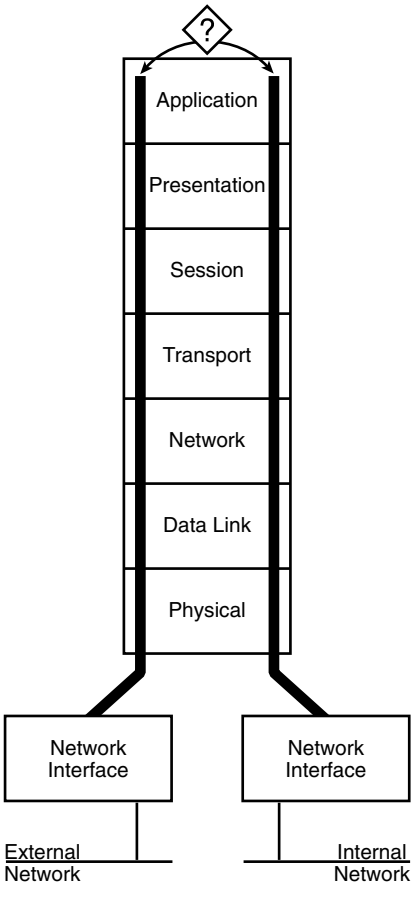


EXHIBIT 49.9 Proxies filtering packets at the application layer.

tion firewalls) and users made claims that the security an application-level gateway offers had an inherent drawback — a lack of transparency.

In moving software from older 16-bit code to current technology's 32-bit environment, and with the advent of SMP, many of today's application-level gateways are just as transparent as they are secure. Users on the public or trusted network in most cases do not notice that they are accessing Internet services through a firewall.

Application-level gateway considerations include:

- Pros:
  - — Application gateway with SMP affords a moderate impact on network performance.
  - — Breaks direct connection to server behind firewall, eliminating the risk of an entire class of covert channel attacks.
  - — Strong application proxy that inspects protocol header lengths can eliminate an entire class of buffer overrun attacks.
  - — Highest level of security.
- Cons:
  - — Poor implementation can have a high impact on network performance.
  - — Must be written securely. Historically, some vendors have introduced buffer overruns within the application gateway.

— Vendors must keep up with new protocols. A common complaint of application-level gateway users is lack of timely vendor support for new protocols.
— A poor implementation that relies on the underlying OS Inetd daemon will suffer from a severe limitation to the number of allowed connections in today's demanding high simultaneous session environment.

Stateful Inspection

Stateful inspection combines the many aspects of dynamic packet filtering, and circuit-level and application-level gateways. While stateful inspection has the inherent ability to examine all seven layers of the OSI model (see Exhibit 49.10), in the majority of applications observed by the author, stateful inspection was operated only at the network layer of the OSI model and used only as a dynamic packet filter for filtering all incoming and outgoing packets based on source and destination IP addresses and port numbers. While the vendor claims this is the fault of the administrator's configuration, many administrators claim that the operating overhead associated with the stateful inspection process prohibits its full utilization.

While stateful inspection has the inherent ability to inspect all seven layers of the OSI model, most installations only operate as a dynamic packet filter at the network layer of the model.

As indicated, stateful inspection can also function as a circuit-level gateway, determining whether the packets in a session are appropriate. For example, stateful inspection can verify that inbound SYN and ACK flags and sequence numbers are logical. However, in most implementations the stateful inspection-based firewall operates

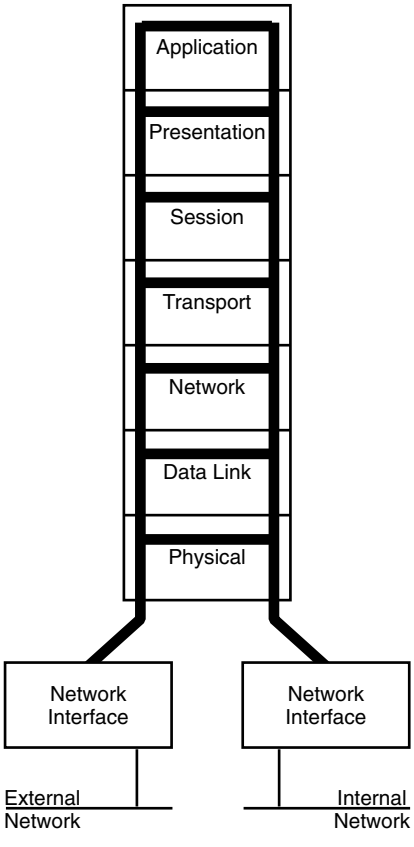


EXHIBIT 49.10 Stateful inspection examining all seven layers of the OSI model.

only as a dynamic packet filter and, dangerously, allows new connections to be established with a single SYN packet. A unique limitation of one popular stateful inspection implementation is that it does not provide the ability to inspect sequence numbers on outbound packets from users behind the firewall. This leads to a flaw whereby internal users can easily spoof the IP address of other internal users to open holes through the associated firewall for inbound connections.

Finally, stateful inspection can mimic an application-level gateway. Stateful inspection can evaluate the contents of each packet up through the application layer and ensure that these contents match the rules in the administrator's network security policy.

Better Performance, But What about Security?

Like an application-level gateway, stateful inspection can be configured to drop packets that contain specific commands within the Application Header. For example, the administrator could configure a stateful inspection firewall to drop HTTP packets containing a *Put* command. However, historically the performance impact of application-level filtering by the single-threaded process of stateful inspection has caused many administrators to abandon its use and to simply opt for dynamic packet filtering to allow the firewall to keep up with network load requirements. In fact, the default configuration of a popular stateful inspection firewall utilizes dynamic packet filtering and not stateful inspection of the most popular protocol on today's Internet — HTTP traffic.

Do Current Stateful Inspection Implementations Expose the User to Additional Risks?

Unlike an application-level gateway, stateful inspection does not break the client/server model to analyze application-layer data. An application-level gateway creates two connections: one between the trusted client and the gateway, and another between the gateway and the untrusted host. The gateway then copies information between these two connections. This is the core of the well-known proxy versus stateful inspection debate. Some administrators insist that this configuration ensures the highest degree of security; other administrators argue that this configuration slows performance unnecessarily. In an effort to provide a secure connection, a stateful inspection-based firewall has the ability to intercept and examine each packet up through the application layer of the OSI model. Unfortunately, because of the associated performance impact of the single-threaded stateful inspection process, this configuration is not the one typically deployed.

Looking beyond marketing hype and engineering theory, stateful inspection relies on algorithms within an inspection engine to recognize and process application-layer data. These algorithms compare packets against known bit patterns of authorized packets. Vendors have claimed that, theoretically, they are able to filter packets more efficiently than application-specific proxies. However, most stateful inspection engines represent a single-threaded process. With current-technology, SMP-based application-level gateways operating on multi-processor servers, the gap has dramatically narrowed. As an example, one vendor's SMP-capable multi-architecture firewall that does not use stateful inspection outperforms a popular stateful inspection-based firewall up to 4:1 on throughput and up to 12:1 on simultaneous sessions. Further, due to limitations in the inspection language used in stateful inspection engines, application gateways are now commonly used to fill in the gaps.

Stateful inspection considerations include:

- Pros:
  - — Offers the ability to inspect all seven layers of the OSI model and is user configurable to customize specific filter constructs.
  - — Does not break the client/server model.
  - — Provides an integral dynamic (stateful) packet filter.
  - — Fast when operated as dynamic packet filter; however, many SMP-compliant dynamic packet filters are actually faster.
- Cons:
  - — The single-threaded process of the stateful inspection engine has a dramatic impact on performance, so many users operate the stateful inspection-based firewall as nothing more than a dynamic packet filter.

— Many believe the failure to break the client/server model creates an unacceptable security risk because the hacker has a direct connection to the protected server.
— A poor implementation that relies on the underlying OS Inetd daemon will suffer from a severe limitation to the number of allowed connections in today's demanding high simultaneous session environment.
— Low level of security. No stateful inspection-based firewall has achieved higher than a Common Criteria EAL 2. Per the Common Criteria EAL 2 certification documents, EAL 2 products are not intended for use in protecting private networks when connecting to the public Internet.

Cutoff Proxy

The cutoff proxy is a hybrid combination of a dynamic (stateful) packet filter and a circuit-level proxy. In the most common implementations, the cutoff proxy first acts as a circuit-level proxy in verifying the RFC-recommended three-way handshake and then switches over to a dynamic packet filtering mode of operation. Hence, it initially works at the session layer — OSI layer 5 — and then switches to a dynamic packet filter working at the network layer — OSI layer 3 — after the connection is completed (see Exhibit 49.11).

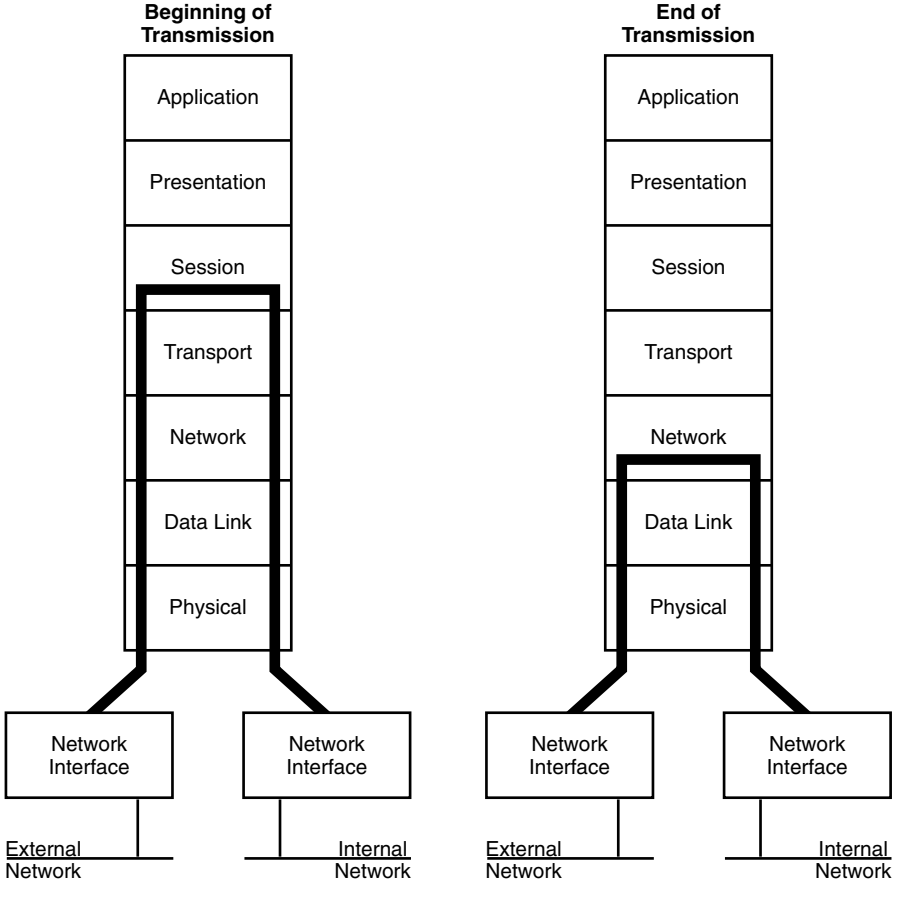


EXHIBIT 49.11 Cutoff proxy filtering packets.

The cutoff proxy verifies the RFC-recommended three-way handshake and then switches to a dynamic packet filter mode of operation.

Some vendors have expanded the capability of the basic cutoff proxy to reach all the way up into the application layer to handle limited authentication requirements (FTP type) before switching back to a basic dynamic packet-filtering mode of operation.

We pointed out what the cutoff proxy does; now, more importantly, we need to discuss what it does *not* do. The cutoff proxy is not a traditional circuit-level proxy that breaks the client/server model for the duration of the connection. There is a direct connection established between the remote client and the protected server behind the firewall. This is not to say that a cutoff proxy does not provide a useful balance between security and performance. At issue with respect to the cutoff proxy are vendors who exaggerate by claiming that their cutoff proxy offers a level of security equivalent to a traditional circuit-level gateway with the added benefit of the performance of a dynamic packet filter.

In clarification, this author believes that all firewall architectures have their place in Internet security. If your security policy requires authentication of basic services and examination of the three-way handshake and does *not* require breaking of the client/server model, the cutoff proxy is a good fit. However, administrators must be fully aware and understand that a cutoff proxy clearly is not equivalent to a circuit-level proxy because the client/server model is not broken for the duration of the connection.

Cutoff proxy considerations include:

- Pros:
  - — There is less impact on network performance than in a traditional circuit gateway.
  - — IP spoofing issue is minimized as the three-way connection is verified.
- Cons:
  - — Simply put, it is not a circuit gateway.
  - — It still has many of the remaining issues of a dynamic packet filter.
  - — It is unaware of packet payload and thus offers low level of security.
  - — It is difficult to create rules (order of precedence).
  - — It can offer a false sense of security because vendors incorrectly claim it is equivalent to a traditional circuit gateway.

Air Gap

The latest entry into the array of available firewall architectures is the air gap. At the time of this writing, the merits of air gap technology remain hotly debated among the security-related Usenet news groups. With air gap technology, the external client connection causes the connection data to be written to a SCSI e-disk (see Exhibit 49.12). The internal connection then reads this data from the SCSI e-disk. By breaking the direct connection between the client to the server and independently writing to and reading from the SCSI e-disk, the respective vendors believe they have provided a higher level of security and a resultant "air gap."

Air gap vendors claim that, while the operation of air gap technology resembles that of the application-level gateway (see Exhibit 49.13), an important difference is the separation of the content inspection from the "front end" by the isolation provided by the air gap. This may very well be true for those firewall vendors that implement their firewalls on top of a standard commercial operating system. But with the current-technology firewall operating on a kernel-hardened operating system, there is little distinction. Simply put, those vendors that chose to implement kernel-level hardening of the underlying operating system utilizing multi-level security (MLS) or containerization methodologies provide no less security than current air gap technologies.

The author finds it difficult to distinguish air gap technology from application-level gateway technology. The primary difference appears to be that air gap technology shares a common SCSI e-disk, while application-level technology shares common RAM. One must also consider the performance limitations of establishing the air gap in an external process (SCSI drive) and the high performance of establishing the same level of separation in a secure kernel-hardened operating system running in kernel memory space.

Any measurable benefit of air gap technology has yet to be verified by any recognized third-party testing authority. Further, the current performance of most air gap-like products falls well behind that obtainable by

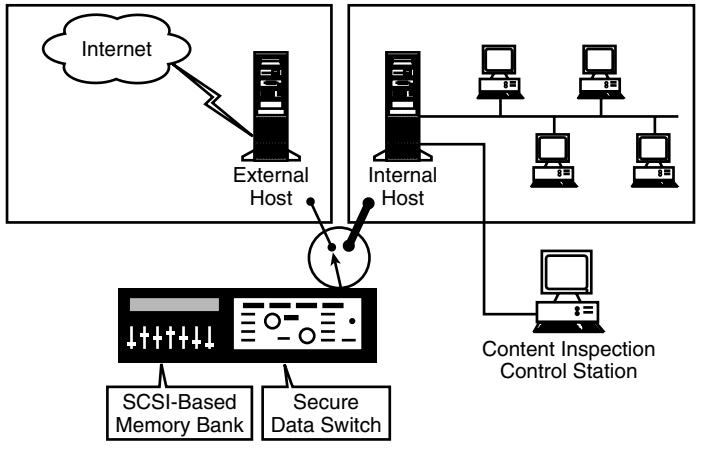


EXHIBIT 49.12 Air gap architecture.

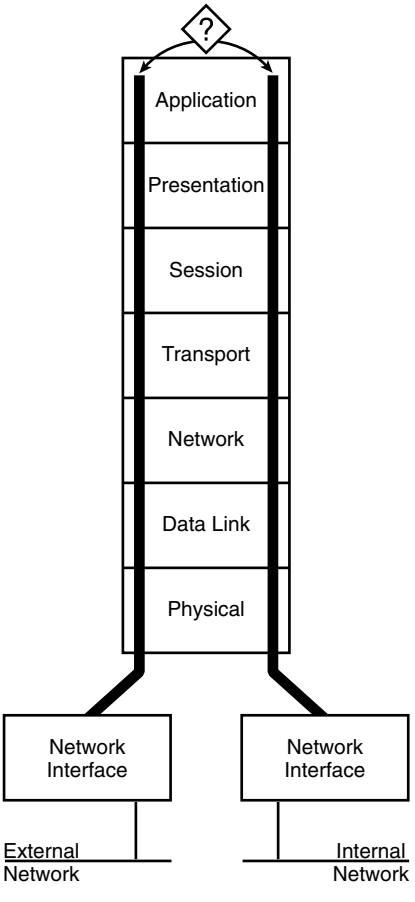


EXHIBIT 49.13 Air gap operating at the application layer.

traditional application-level gateway based products. Without a verifiable benefit to the level of security provided, the necessary performance costs are prohibitive for many system administrators.

Air gap considerations include:

- Pros:
  - It breaks direct connection to the server behind the firewall, eliminating the risk of an entire class of covert channel attacks.
  - Strong application proxy that inspects protocol header lengths can eliminate an entire class of buffer overrun attacks.
  - As with an application-level gateway, an air gap can potentially offer a high level of security.
- Cons:
  - It can have a high negative impact on network performance.
  - Vendors must keep up with new protocols. A common complaint of application-level gateway users is the lack of timely response from a vendor to provide application-level gateway support for a new protocol.
  - It is currently not verified by any recognized third-party testing authority.

Other Considerations

ASIC-Based Firewalls

Looking at typical ASIC-based offerings, the author finds that virtually all are VPN/firewall hybrids. These hybrids provide fast VPN capabilities but most often are only complemented with a limited single-architecture stateful firewall capability.

Today's security standards are in flux, so ASIC designs must be left programmable or "soft" enough that the full speed of ASICs simply cannot be unleashed.

ASIC technology most certainly brings a new level of performance to VPN operations. IPSec VPN encryption and decryption run inarguably better in hardware than in software. However, in most accompanying firewall implementations, a simple string comparison (packet to rule base) is the only functionality that is provided within the ASIC. Hence, the term "ASIC-based firewall" is misleading at best. The majority of firewall operations in ASIC-based firewalls are performed in software operating on microprocessors. These firewall functions often include NAT, routing, cutoff proxy, authentication, alerting, and logging.

When you commit to an ASIC, you eliminate the flexibility necessary to deal with future Internet security issues. Network security clearly remains in flux. While an ASIC can be built to be *good enough* for a particular purpose or situation, is *good enough* today really *good enough* for tomorrow's threats?

Hardware-Based Firewalls

The term *hardware-based firewall* is another point of confusion in today's firewall market. For clarification, most hardware-based firewalls are products that have simply eliminated the spinning media (hard disk drive) associated with the typical server or appliance-based firewalls.

Most hardware firewalls are either provided with some form of solid-state disk, or they simply boot from ROM, load the OS and application from firmware to RAM, and then operate in a manner similar to a conventional firewall.

The elimination of the spinning media is both a strength and a weakness of a hardware-based firewall. *Strength* is derived from limited improvements in MTBF and environmental performance by eliminating the spinning media. *Weakness* is present in severe limitations to the local alerting and logging capability, which most often requires a separate logging server to achieve any usable historical data retention.

Other Considerations: A Brief Discussion of OS Hardening

One of the most misunderstood terms in network security with respect to firewalls today is *OS hardening* or *hardened OS*. Many vendors claim their network security products are provided with a hardened OS. What

you will find in virtually all cases is that the vendor simply turned off or removed unnecessary services and patched the operating system or OS for known vulnerabilities. Clearly, this is not a hardened OS but really a *patched OS*.

What Is a Hardened OS?

A hardened OS (see Exhibit 49.14) is one in which the vendor has modified the kernel source code to provide for a mechanism that clearly provides a security perimeter among the non-secure application software, the secure application software, and the network stack. This eliminates the risk of the exploitation of a service running on the hardened OS that could otherwise provide root-level privilege to the hacker.

In a hardened OS, the security perimeter is established using one of two popular methodologies:

1. *Multi-Level Security (MLS):* establishes a perimeter through the use of labels assigned to each packet and applies rules for the acceptance of said packets at various levels of the OS and services
2. *Compartmentalization:* provides a sandbox approach whereby an application effectively runs in a dedicated kernel space with no path to another object within the kernel

Other security-related enhancements typically common in kernel-level hardening methodologies include:

- Separation of event logging from root
- Mandatory access controls
- File system security enhancements
- Log EVERYTHING from all running processes

What Is a Patched OS?

A patched OS is typically a commercial OS from which the administrator turns off or removes all unnecessary services and installs the latest security patches from the OS vendor. A patched OS has had no modifications made to the kernel source code to enhance security.

Is a Patched OS as Secure as a Hardened OS?

Simply put, no. A patched OS is only secure until the next vulnerability in the underlying OS or allowed services is discovered. An administrator may argue that when he has completed installing his patches and turning off services, his OS is, in fact, secure. The bottom-line question is: with more than 100 new vulnerabilities being posted to Bug Traq each month, how long will it remain secure?

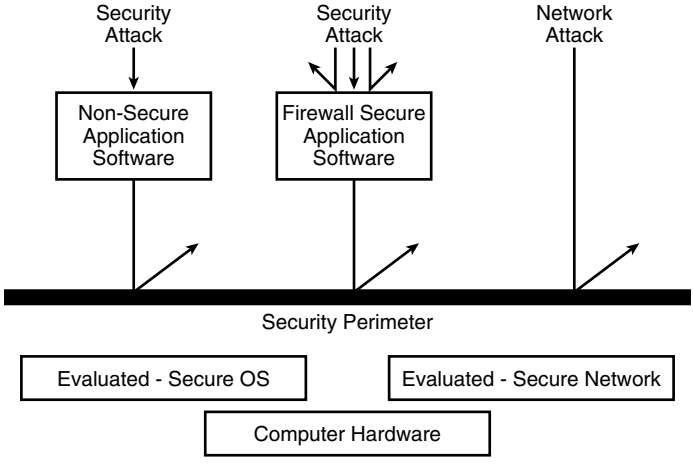


EXHIBIT 49.14 Hardened OS.

How Do You Determine if a Product Is Provided with a Hardened OS?

If the product was supplied with a commercial OS, you can rest assured that it is *not* a hardened OS. The principal element here is that, to harden an OS, you must own the source code to the OS so you can make the necessary kernel modification to harden the OS. If you really want to be sure, ask the vendor to provide third-party validation that the OS is, in fact, hardened at the kernel level, (e.g., http://www.radium.ncsc.mil/tpep/epl/historical.html).

Why Is OS Hardening Such an Important Issue?

Too many in the security industry have been lulled into a false sense of security. Decisions on security products are based primarily on popularity and price, with little regard for the actual security the product can provide.

Where Can You Find Additional Information about OS Vulnerabilities?

- www.securiteam.com
- www.xforce.iss.net
- www.rootshell.com
- www.packetstorm.securify.com
- www.insecure.org/sploits.html

Where Can You Find Additional Information about Patching an OS?

More than 40 experts in the SANS community have worked together over a full year to create the following elegant and effective scripts:

- For Solaris, http://yassp.parc.xerox.com/
- For Red Hat Linux, http://www.sans.org/newlook/projects/bastille_ linux.htm

Lance Spitzner (http://www.enteract.com/~lspitz/pubs.html) has written a number of excellent technical documents, including:

- Armoring Linux
- Armoring Solaris
- Armoring NT

Stanford University (http://www.stanford.edu/group/itss-ccs/security/Bestuse/Systems/) has also released a number of informative technical documents:

- Red Hat Linux
- Solaris
- SunOS
- AIX 4.x
- HPUX
- NT

Conclusion

Despite claims by various vendors, no single firewall architecture is the "holy grail" in network security. It has been said many times and in many ways by network security experts: if you believe any one technology is going to solve the Internet security problem, you do not understand the technology and you do not understand the problem.

Unfortunately for the Internet community at large, many administrators today design the security policy for their organizations around the limited capabilities of a specific vendor's product. The author firmly believes all firewall architectures have their respective place or role in network security. Selection of any specific firewall architecture should be a function of the organization's security policy and should not be based solely on the limitation of the vendor's proposed solution. The proper application of multiple firewall architectures to

support the organization's security policy in providing the acceptable balance of trust and performance is the only viable methodology in securing a private network when connecting to the public Internet.

One of the most misunderstood terms in network security with respect to firewalls today is *OS hardening*, or *hardened OS*. Simply put, turning off or removing a few unnecessary services and patching for known product vulnerabilities does not build a hardened OS. Hardening an OS begins with modifying the OS software at the kernel level to facilitate building a security perimeter. This security perimeter isolates services and applications from providing root access in the event of application- or OS-provided service compromise. Effectively, only a properly implemented hardened OS with a barrier at the kernel level will provide for an impenetrable firewall platform.

References

This text is based on numerous books, white papers, presentations, vendor literature, and various Usenet newsgroup discussions I have read or participated in throughout my career. Any failure to cite any individual for anything that in any way resembles a previous work is unintentional.

Deploying Host-Based Firewalls across the Enterprise: A Case Study

Jeffery Lowder, CISSP

Because hosts are exposed to a variety of threats, there is a growing need for organizations to deploy host-based firewalls across the enterprise. This chapter outlines the ideal features of a host-based firewall — features that are typically not needed or present in a purely *personal* firewall software implementation on a privately owned PC. In addition, the author describes his own experiences with, and lessons learned from, deploying agent-based, host-based firewalls across an enterprise. The author concludes that host-based firewalls provide a valuable additional layer of security.

A SEMANTIC INTRODUCTION

Personal firewalls are often associated with (and were originally designed for) home PCs connected to "always-on" broadband Internet connections. Indeed, the term *personal firewall* is itself a vestige of the product's history: originally distinguished from *enterprise* firewalls, *personal* firewalls were initially viewed as a way to protect home PCs.[1] Over time, it was recognized that personal firewalls had other uses. The security community began to talk about using personal firewalls to protect notebooks that connect to the enterprise LAN via the Internet and eventually protecting notebooks that physically reside on the enterprise LAN.

Consistent with that trend — and consistent with the principle of defense-in-depth — it can be argued that the time has come for the potential usage of personal firewalls to be broadened once again. Personal firewalls should really be viewed as *host-based* firewalls. As soon as one makes the distinction between host-based and network-based firewalls, the additional use of a host-based firewall becomes obvious. Just as organizations deploy host-based *intrusion detection systems* (IDS) to provide an additional detection capability for critical servers, organizations should consider deploying host-based *firewalls* to provide an additional layer of access control for critical servers (e.g., exchange servers, domain controllers, print servers, etc.). Indeed, given that many host-based firewalls have an IDS capability built in, it is conceivable that, at least for some small organizations, host-based firewalls could even *replace* specialized host-based IDS software.

The idea of placing one firewall behind another is not new. For years, security professionals have talked about using so-called internal firewalls to protect especially sensitive back-office systems.[2] However, internal firewalls, like network-based firewalls in general, are still dedicated devices. (This applies to both firewall appliances such as Cisco's PIX and software-based firewalls such as Symantec's Raptor.) In contrast, host-based firewalls require no extra equipment. A host-based firewall is a firewall software package that runs on a preexisting server or client machine. Given that a host-based firewall runs on a server or client machine (and is responsible for protecting *only* that machine), host-based firewalls offer greater functionality than network-based firewalls, even including internal firewalls that are dedicated to protecting a single machine. Whereas both network- and host-based firewalls have the ability to filter inbound and outbound network connections, only host-based firewalls possess the *additional* capabilities of blocking network connections linked to specific programs and preventing the execution of mail attachments.

To put this into proper perspective, consider the network worm and Trojan horse program QAZ, widely suspected to be the exploit used in the November 2000 attack on Microsoft's internal network. QAZ works by hijacking the NOTEPAD.EXE program. From the end user's perspective, Notepad still appears to run normally; but each time Notepad is launched, QAZ sends an e-mail message (containing the IP address of the infected machine) to some address in China.[3] Meanwhile, in the background, the Trojan patiently waits for a connection on TCP port 7597, through which an intruder can upload and execute any applications.[4] Suppose QAZ were modified to run over TCP port 80 instead.[5] While all firewalls can block outbound connections on TCP port 80, implementing such a configuration would interfere with legitimate traffic. Only a host-based firewall can block an outbound connection on TCP port 80 associated with NOTEPAD.EXE and notify the user of the event. As Steve Riley notes, "Personal firewalls

that monitor outbound connections will raise an alert; seeing a dialog with the notice 'Notepad is attempting to connect to the Internet' should arouse anyone's suspicions."[6]

STAND-ALONE VERSUS AGENT-BASED FIREWALLS

Host-based firewalls can be divided into two categories: stand-alone and agent-based.[7] Stand-alone firewalls are independent of other network devices in the sense that their configuration is managed (and their logs are stored) on the machine itself. Examples of stand-alone firewalls include ZoneAlarm, Sygate Personal Firewall Pro, Network Associates' PGP Desktop Security, McAfee Personal Firewall,[8] Norton Internet Security 2000, and Symantec Desktop Firewall.

In contrast, agent-based firewalls are not locally configured or monitored. Agent-based firewalls are configured from (and their logs are copied to) a centralized enterprise server. Examples of agent-based firewalls include ISS RealSecure Desktop Protector (formerly Network ICE's Black ICE Defender) and InfoExpress's CyberArmor Personal Firewall.

We chose to implement agent-based firewall software on our hosts. While stand-alone firewalls are often deployed as an enterprise solution, we wanted the agent-based ability to centrally administer and enforce a consistent access control list (ACL) across the enterprise. And as best practice dictates that the logs of network-based firewalls be reviewed on a regular basis, we wanted the ability to aggregate logs from host-based firewalls across the enterprise into a single source for regular review and analysis.

OUR PRODUCT SELECTION CRITERIA

Once we adopted an agent-based firewall model, our next step was to select a product. Again, as of the time this chapter was written, our choices were RealSecure Desktop Protector or CyberArmor. We used the following criteria to select a product:[9]

- *Effectiveness in blocking attacks.* The host-based firewall should effectively deny malicious inbound traffic. It should also at least be capable of effectively filtering outbound connections. As Steve Gibson argues, "Not only must our Internet connections be fortified to prevent *external intrusion,* they also [must] provide secure management of *internal extrusion.*"[10] By internal extrusion, Gibson is referring to outbound connections initiated by Trojan horses, viruses, and spyware. To effectively filter outbound connections, the host-based firewall must use cryptographic sums. The host-based firewall must first generate cryptographic sums for each authorized application and then regenerate and compare that sum to the one stored in the database before any program (no matter what the filename) is allowed access. If the application

does not maintain a database of cryptographic sums for all authorized applications (and instead only checks filenames or file paths), the host-based firewall may give an organization a false sense of security.

- *Centralized configuration.* Not only did we need the ability to centrally define the configuration of the host-based firewall, we also required the ability to *enforce* that configuration. In other words, we wanted the option to prevent end users from making security decisions about which applications or traffic to allow.
- *Transparency to end users.* Because the end users would not be making any configuration decisions, we wanted the product to be as transparent to them as possible. For example, we did not want users to have to 'tell' the firewall how their laptops were connected (e.g., corporate LAN, home Internet connection, VPN, extranet, etc.) in order to get the right policy applied. In the absence of an attack, we wanted the firewall to run silently in the background without noticeably degrading performance. (Of course, in the event of an attack, we would want the user to receive an alert.)
- *Multiple platform support.* If we were only interested in personal firewalls, this would not have been a concern. (While Linux notebooks arguably might need personal firewall protection, we do not have such machines in our environment.) However, because we are interested in implementing host-based firewalls on our servers as well as our client PCs, support for multiple operating systems is a requirement.
- *Application support.* The firewall must be compatible with all authorized applications and the protocols used by those applications.
- *VPN support.* The host-based firewall must support our VPN implementation and client software. In addition, it must be able to detect and transparently adapt to VPN connections.
- *Firewall architecture.* There are many options for host-based firewalls, including packet filtering, application-level proxying, and stateful inspection.
- *IDS technology.* Likewise, there are several different approaches to IDS technology, each with its own strengths and weaknesses. The number of attacks detectable by a host-based firewall will clearly be relevant here.
- *Ease of use and installation.* As an enterprisewide solution, the product should support remote deployment and installation. In addition, the central administrative server should be (relatively) easy to use and configure.
- *Technical support.* Quality and availability are our prime concerns.
- *Scalability.* Although we are a small company, we do expect to grow. We need a robust product that can support a large number of agents.
- *Disk space.* We were concerned about the amount of disk space required on end-user machines as well as the centralized policy and logging server. For example, does the firewall count the number of times

an attack occurs rather than log a single event for every occurrence of an attack?
- *Multiple policy groups.* Because we have diverse groups of end users, each with unique needs, we wanted the flexibility to enforce different policies on different groups. For example, we might want to allow SQL-Net traffic from our development desktops while denying such traffic for the rest of our employees.
- *Reporting.* As with similar enterprise solutions, an ideal reporting feature would include built-in reports for top intruders, targets, and attack methods over a given period of time (e.g., monthly, weekly, etc.).
- *Cost.* As a relatively small organization, we were especially concerned about the cost of selecting a high-end enterprise solution.

OUR TESTING METHODOLOGY

We eventually plan to install and evaluate both CyberArmor and RealSecure Desktop Protector by conducting a pilot study on each product with a small, representative sample of users. (At the time this chapter was written, we were nearly finished with our evaluation of CyberArmor and about to begin our pilot study of ISS Real Secure.) While the method for evaluating both products according to most of our criteria is obvious, our method for testing one criterion deserves a detailed explanation: effectiveness in blocking attacks. We tested the effectiveness of each product in blocking unauthorized connections in several ways:

- *Remote Quick Scan from HackYourself.com.*[11] From a dial-up connection, we used HackYourself.com's Quick Scan to execute a simple and remote TCP and UDP port scan against a single IP address.
- *Nmap scan.* We used nmap to conduct two different scans. First, we performed an ACK scan to determine whether the firewall was performing stateful inspection or a simple packet filter. Second, we used nmap's operating system fingerprinting feature to determine whether the host-based firewall effectively blocked attempts to fingerprint target machines.
- *Gibson Research Corporation's LeakTest.* LeakTest determines a firewall product's ability to effectively filter *outbound* connections initiated by Trojans, viruses, and spyware.[12] This tool can test a firewall's ability to block LeakTest when it masquerades as a trusted program (OUTLOOK.EXE).
- *Steve Gibson's TooLeaky.* TooLeaky determines whether the firewall blocks unauthorized programs from controlling trusted programs. The TooLeaky executable tests whether this ability exists by spawning Internet Explorer to send a short, innocuous string to Steve Gibson's Web site, and then receiving a reply.[13]
- *Firehole.* Firehole relies on a modified dynamic link library (DLL) that is used by a trusted application (Internet Explorer). The test is whether

the firewall allows the trusted application, under the influence of the malicious DLL, to send a small text message to a remote machine. The message contains the currently logged-on user's name, the name of the computer, and a message claiming victory over the firewall and the time the message was sent.[14]

CONFIGURATION

One of our reasons for deploying host-based firewalls was to provide an additional layer of protection against Trojan horses, spyware, and other programs that initiate outbound network connections. While host-based firewalls are not designed to interfere with Trojan horses that do not send or receive network connections, they can be quite effective in blocking network traffic to or from an unauthorized application when configured properly. Indeed, in one sense, host-based firewalls have an advantage over anti-virus software. Whereas anti-virus software can only detect Trojan horses that match a known *signature*, host-based firewalls can detect Trojan horses based on their network *behavior*. Host-based firewalls can detect, block, and even terminate any unauthorized application that attempts to initiate an outbound connection, even if that connection is on a well-known port like TCP 80 or even if the application causing that connection appears legitimate (NOTEPAD.EXE).

However, there are two well-known caveats to configuring a host-based firewall to block Trojan horses. First, the firewall must block all connections initiated by new applications *by default*. Second, the firewall must not be circumvented by end users who, for whatever reason, click "yes" whenever asked by the firewall if it should allow a new application to initiate outbound traffic. Taken together, these two caveats can cause the cost of ownership of host-based firewalls to quickly escalate. Indeed, other companies that have already implemented both caveats report large numbers of help desk calls from users wanting to get a specific application authorized.[15]

Given that we do not have a standard desktop image and given that we have a very small help desk staff, we decided to divide our pilot users into two different policy groups: pilot-tech-technical and pilot-normal-regular (See Exhibit 10-1).

The first configuration enabled users to decide whether to allow an application to initiate an outbound connection. This configuration was implemented only on the desktops of our IT staff. The user must choose whether to allow or deny the network connection requested by the connection. Once the user makes that choice, the host-based firewall generates a checksum and creates a rule reflecting the user's decision. (See Exhibit 10-2 for a sample rule set in CyberArmor.)

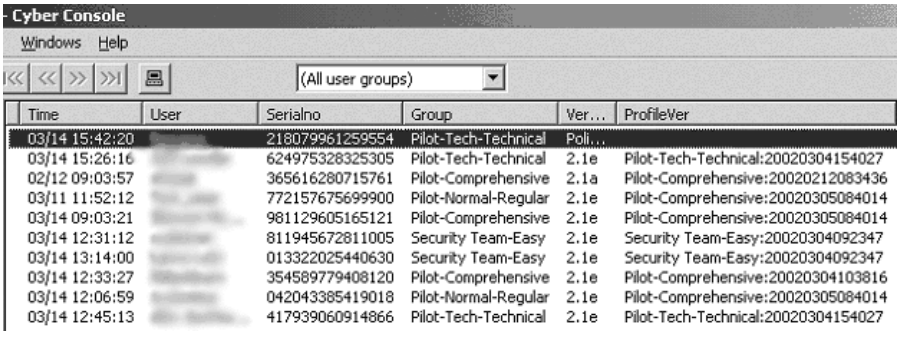


Exhibit 10-1. CyberArmor policy groups.

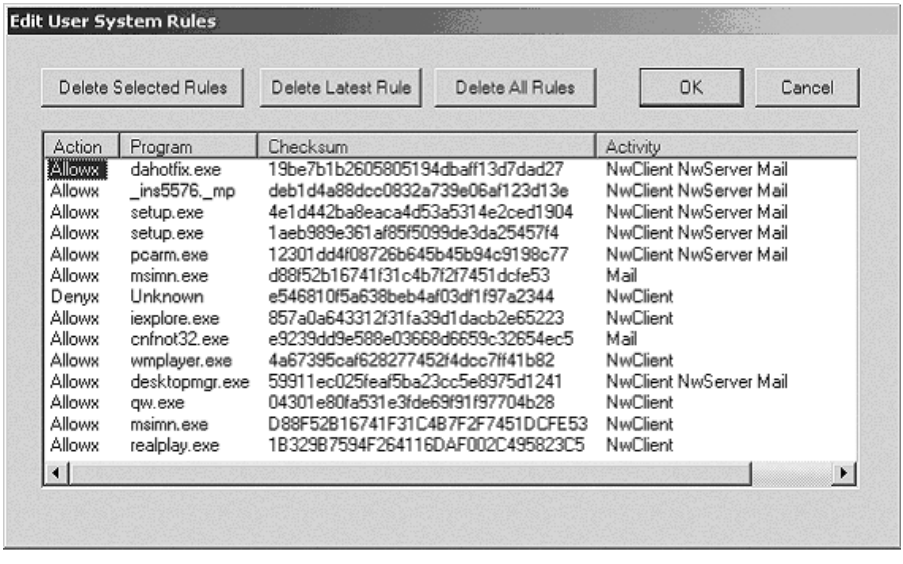


Exhibit 10-2. Sample user-defined rules in CyberArmor.

The second configuration denied all applications by default and only allowed applications that had been specifically authorized. We applied this configuration on all laptops outside our IT organization, because we did not want to allow nontechnical users to make decisions about the configuration of their host-based firewall.

LESSONS LEARNED

Although at the time this chapter was finished we had not yet completed our pilot studies on both host-based firewall products, we had already

learned several lessons about deploying agent-based, host-based firewalls across the enterprise. These lessons may be summarized as follows.

1. Our pilot study identified one laptop with a nonstandard and, indeed, unauthorized network configuration. For small organizations that do not enforce a standard desktop image, this should not be a surprise.
2. The ability to enforce different policies on different machines is paramount. This was evident from our experience with the host-based firewall to restrict outbound network connections. By having the ability to divide our users into two groups, those we would allow to make configuration decisions and those we would not, we were able to get both flexibility and security.
3. As is the case with network-based intrusion detection systems, our experience validated the need for well-crafted rule sets. Our configuration includes a rule that blocks inbound NetBIOS traffic. Given the amount of NetBIOS traffic present on both our internal network as well as external networks, this generated a significant amount of alerts. This, in turn, underscored the need for finely tuned alerting rules.
4. As the author has found when implementing network-based firewalls, the process of constructing and then fine-tuning a host-based firewall rule set is time consuming. This is especially true if one decides to implement restrictions on outbound traffic (and not allow users or a portion of users to make configuration decisions of their own), because one then has to identify and locate the exact file path of each authorized application that has to initiate an outbound connection. While this is by no means an insurmountable problem, there was a definite investment of time in achieving that configuration.
5. We did not observe any significant performance degradation on end user machines caused by the firewall software. At the time this chapter was written, however, we had not yet tested deploying host-based firewall software on critical servers.
6. Our sixth observation is product specific. We discovered that the built-in reporting tool provided by CyberArmor is primitive. There is no built-in support for graphical reports, and it is difficult to find information using the text reporting. For example, using the built-in text-reporting feature, one can obtain an "alarms" report. That report, presented in spreadsheet format, merely lists alarm messages and the number of occurrences. Source IP addresses, date, and time information are not included in the report. Moreover, the alarm messages are somewhat cryptic. (See Exhibit 10-3 for a sample CyberArmor Alarm Report.) While CyberArmor is compatible with Crystal Reports, using Crystal Reports to produce useful reports requires extra software and time.

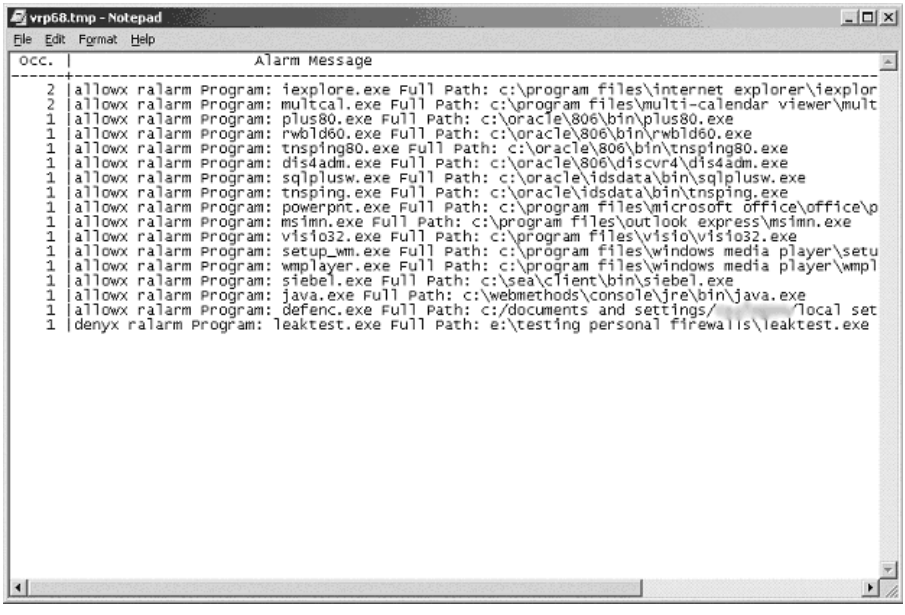


Exhibit 10-3. Sample CyberArmor alarm report.

HOST-BASED FIREWALLS FOR UNIX?

Host-based firewalls are often associated with Windows platforms, given the history and evolution of personal firewall software. However, there is no reason in theory why host-based firewalls cannot (or should not) be implemented on UNIX systems as well. To be sure, some UNIX packet filters already exist, including ipchains, iptables, and ipfw.[16] Given that UNIX platforms have not been widely integrated into commercial host-based firewall products, these utilities may be very useful in an enterprisewide host-based firewall deployment. However, such tools generally have two limitations worth noting. First, unlike personal firewalls, those utilities are packet filters. As such, they do not have the capability to evaluate an outbound network connection according to the application that generated the connection. Second, the utilities are not agent based. Thus, as an enterprise solution, those tools might not be easily scalable. The lack of an agent-based architecture in such tools might also make it difficult to provide centralized reporting on events detected on UNIX systems.

CONCLUSIONS

While host-based firewalls are traditionally thought of as a way to protect corporate laptops and privately owned PCs, host-based firewalls can also provide a valuable layer of additional protection for servers. Similarly, while host-based firewalls are typically associated with Windows platforms,

they can also be used to protect UNIX systems as well. Moreover, host-based firewalls can be an effective tool for interfering with the operation of Trojan horses and similar applications. Finally, using an agent-based architecture can provide centralized management and reporting capability over all host-based firewalls in the enterprise.

Acknowledgments

The author wishes to acknowledge Frank Aiello and Derek Conran for helpful suggestions. The author is also grateful to Lance Lahr, who proofread an earlier version of this chapter.

References

1. Michael Cheek, Personal firewalls block the inside threat. *Gov. Comp. News* 19:3 (3 April 2000). Spotted electronically at <URL:http://www.gcn.com/vol19_no7/reviews/1602-1.html>, February 6, 2002.
2. William R. Cheswick and Steven M. Bellovin, *Firewalls and Internet Security: Repelling the Wily Hacker* (New York: Addison-Wesley, 1994), pp. 53–54.
3. F-Secure Computer Virus Information Pages: QAZ (<URL:http://www.europe.f-secure.com/v-descs/qaz.shtml>, January 2001), spotted February 6, 2002.
4. TROJ_QAZ.A — Technical Details (<URL:http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName = TROJ_QAZ.A&VSect = T>, October 28, 2000), spotted February 6, 2002.
5. Steve Riley, Is Your Generic Port 80 Rule Safe Anymore? (<URL:http://rr.sans.org/firewall/port80.php>, February 5, 2001), spotted February 6, 2002.
6. Steve Riley, Is Your Generic Port 80 Rule Safe Anymore? (<URL:http://rr.sans.org/firewall/port80.php>, February 5, 2001), spotted February 6, 2002.
7. Michael Cheek, Personal firewalls block the inside threat. *Gov. Comp. News* 19:3 (3 April 2000). Spotted electronically at <URL:http://www.gcn.com/vol19_no7/reviews/1602-1.html>, February 6, 2002.
8. Although McAfee is (at the time this chapter was written) currently in Beta testing with its own agent-based product, Personal Firewall 7.5, that product is not scheduled to ship until late March 2002. See Douglas Hurd, The Evolving Threat (<URL:http://www.issa-dv.org/meetings/web/2002/08FEB02/McAfee%20ISSA-DV%20Meeting%20FEB02.pdf>, February 8, 2002), spotted February 8, 2002.
9. Cf. my discussion of network-based firewall criteria in Firewall Management and Internet Attacks in *Information Security Management Handbook* (4th ed., New York: Auerbach, 2000), pp. 118–119.
10. Steve Gibson, LeakTest — Firewall Leakage Tester (<URL:http://grc.com/lt/leaktest.htm>, January 24, 2002), spotted February 7, 2002.
11. Hack Yourself Remote Computer Network Security Scan (<URL:http://hackyourself.com:4000/startdemo.dyn>, 2000), spotted February 7, 2002.
12. Leak Test — How to Use Version 1.x (<URL:http://grc.com/lt/howtouse.htm>, November 3, 2001), spotted February 7, 2002.
13. Steve Gibson, Why Your Firewall Sucks :-) (<URL:http://tooleaky.zensoft.com/>, November 5, 2001), spotted February 8, 2002.
14. By default, this message is sent over TCP port 80 but this can be customized. See Robin Keir, Firehole: How to Bypass Your Personal Firewall Outbound Detection (<URL:http://keir.net/firehole.html>, November 6, 2001), spotted February 8, 2002.
15. See, for example, Barrie Brook and Anthony Flaviani, Case Study of the Implementation of Symantec's Desktop Firewall Solution within a Large Enterprise (<URL:http://www.issa-dv.org/meetings/web/2002/08FEB02/Unisys%20ISSA-DV%20Meeting%20FEB02.pdf>, February 8, 2002), spotted February 8, 2002.

16. See Rusty Russell, Linux IPCHAINS-HOWTO (<URL:http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html>, July 4, 2000), spotted March 29, 2002; Oskar Andreasson, Iptables Tutorial 1.1.9 (<URL:http://people.unix-fu.org/andreasson/iptables-tutorial/iptables-tutorial.html">, 2001), spotted March 29, 2002; and Gary Palmer and Alex Nash, Firewalls (<URL:http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/firewalls.html>, 2001), spotted March 29, 2002. I am grateful to an anonymous reviewer for suggesting I discuss these utilities in this chapter.

ABOUT THE AUTHOR

Jeffery Lowder, CISSP, GSEC, is currently working as an independent information security consultant. His interests include firewalls, intrusion detection systems, UNIX security, and incident response. Previously, he has served as the director, security and privacy, for Elemica, Inc.; senior security consultant for PricewaterhouseCoopers, Inc.; and director, network security, at the U.S. Air Force Academy.

50

Instant Messaging Security Issues

William Hugh Murray, CISSP

> Nothing useful can be said about the security of a mechanism except in the context of a specific application and environment.
>
> **— Robert H. Courtney, Jr.**

> Privacy varies in proportion to the cost of surveillance to the government.
>
> **— Lawrence Lessig**

Instant messaging (IM) has moved from home to office, from a toy to an enterprise application. It has become part of our social infrastructure and will become part of our economic infrastructure. Like most technology, it has many uses — some good, some bad. It has both fundamental and implementation-induced issues. This chapter describes IM and gives examples of its implementation. It describes operation and examines some sample uses. It identifies typical threats and vulnerabilities, and examines the security issues that IM raises. It identifies typical security requirements and the controls available to meet them. Finally, it makes security recommendations for users, operators, enterprises, and parents.

Introduction and Background

Instant messaging, or chat, has been around for about 15 years. However, for most of its life, its use has been sparse and its applications trivial. Its use expanded rapidly with its inclusion in America Online's service. For many children, it was the first application of the Internet and the second application of the computer after games. Although many enterprises still resist it, it is now part of the culture. It is an interesting technology in that it originated in the consumer market and is migrating to the enterprise market. Like Web browsing before it, IM is entering the enterprise from the bottom up — from the user to the enterprise.

There may be as many as 100 million IM users but, because many users have multiple handles and subscribe to multiple services, it is difficult to know with any confidence. K. Petersen of *The Seattle Times* reports that many users have two or more IM clients open most of the time.

For most of its life, IM operated in a fairly benign environment. That is, it operated in the Internet in the days when the Internet was fairly benign. As is true of the Internet in general, business and government have been late to the party.

> On 9/11, communications in the nation, and in New York City in particular, were severely disrupted, mostly by unanticipated load. One could make a phone call out of the city but could not call into the city. Most news sites on the WWW did not respond to many requests; responses were limited to a line or two. Broadcast TV in the city was disrupted by loss of its primary antennas; only a few had backup. Cable TV, and broadcast TV outside the city, worked as intended, in part because they were

not sensitive to load. Cell phones worked well for a few minutes but soon fell over to load. The two-way communication that worked best under load was instant messaging. "First responders" found themselves using pagers (one way), SMS on cell phones, AOL Instant Messaging, BlackBerrys, and other forms of instant messaging.

At the risk of using a cliché, IM is a new paradigm. It is altering the way we see the world and will ultimately change the world. IM is changing the workplace as e-mail did before it. (Yes, e-mail changed the workplace. Although not all of us have been around long enough to notice, it has not always been as it is now.)

I was "chatting" with my colleague, Roger, yesterday. We were talking about a new IM client that we were installing on our PDAs. (We both use Handspring Treo communicators, cell phones integrated with a Palm OS PDA.) He said, "IM is the killer application for PDAs." I was surprised. I told him that I was working on this chapter and asked him to elaborate. He went on to say that for those of us who now work primarily from home and road (includes both of us and many of our colleagues), IM is now our virtual water cooler. It is where we conduct that business that we used to conduct by walking the halls or meeting in the cafeteria. It is also our peek-in-the-office-door to see if it is a convenient time to talk. Even if he plans to call a colleague on the phone, he sends an instant message first. IM complements the other spontaneous things that we do with a PDA.

In the discussion below you will see that IM is a network of people built on a network of hardware. Once the servers and protocols are in place, then its capabilities and its integration with other communication methods are limited only by the sophistication of the software clients. IM is the spontaneous collaboration tool of choice.

Description

This section describes instant messaging (IM) while later sections elaborate by discussing illustrative systems and typical operation.

At its most abstract, IM is a client/server application in which users communicate in short messages in near-real-time. The client performs input and output, the Internet provides transport and connectivity, while the servers provide message addressing, and, optionally, message forwarding.

IM's most popular instantiation is AOL Instant Messaging (AIM). There is an AIM client built into the AOL client. There are also AIM clients built into other applications and application suites.

IM users are represented as named windows on the desktop or within the client application. To send a message to the user represented by a window, one simply places the cursor in the window (making it the active window) and types in a message. That message then appears almost simultaneously in the window on someone else's system that represents the other end of the connection.

At its simplest, traffic is *one-to-one.* However, there is a *group mode* in which A sends an invitation to members of an affinity group to participate in a *one-to-many* or ***many-to-many mode.*** There is a second many-to-many mode where a "chat room" is established. The virtual room may be devoted to a group, a topic, or a discussion. Participants can enter or leave the *room* — that is, the discussion — at will. Participants in the room may be represented by nametags or by icons.

In theory, IM is synchronous: that is, a message from A to B is followed by a response from B to A. In practice, it is more "near synchronous;" that is, in part because of message origination latency, messages may be slightly out of order with two or more simultaneous threads.

IM is a relatively open application. While networks, servers, rooms, or groups may be closed to all but named and designated participants, most of them are open to all comers. The infrastructure (i.e., clients, servers, and connections) are open to all.

IM is also relatively interoperable. While most networks and servers interoperate primarily with their peers, many different clients can interoperate with others and many clients will operate with multiple networks and servers. The Trillian Professional client from Cerulean Studios will support simultaneous connections over the AOL, MS, Yahoo, ICQ, and multiple IRC networks. Time Warner, operator of both AIM and ICQ, has announced plans to permit interoperation of the two. Not only do IM systems interoperate with one another, but also with e-mail and voice mail.

Systems

This section identifies some of the more significant IM systems.

AOL IM

Far and away the most popular consumer IM system is AOL IM (AIM). Measured by numbers of registered users or traffic, no other system comes close. AOL well understands that the value of an IM system grows geometrically with the number of regular users.

While IM is bundled into the AOL client, and while it was originally intended for AOL's dial customers, it also uses the Internet where it is open to all comers. Anyone, AOL customer or not, can register a name on the AIM server. A number of stand-alone clients are available, including one from Netscape, AOL's software subsidiary. AOL encourages ISPs (Internet service providers) and other services to bundle an AOL client into their offering.

ICQ

Time Warner is also the operator of Internet CQ (ICQ). Amateur radio operators will recognize the model. While AOL IM is like the telephone, ICQ is more like a ham radio channel. While it is possible to set up a conference call, the telephone is primarily one-to-one. While it is possible to use a ham radio in one-to-one mode, it is essentially a many-to-many medium.

IRC

While some Internet historians date IM from ICQ in 1996, most recognize Internet Relay Chat (IRC), which originated in 1988, as the granddaddy of all instant messaging. IRC was built as an alternative to and elaboration of the (UNIX-to-UNIX) *talk* command. While IRC servers usually run on UNIX systems, clients are available for Wintel systems, IBM VM, EMACS, Macintosh, NeXTStep, VMS, and others. Early IRC clients were command-line driven and oriented. Many purists still prefer to use it in that mode. However, modern clients use a graphical user interface. For example, BitchX is a GUI client for UNIX/X-Windows systems.

Like ICQ, IRC is fundamentally many-to-many. A user does not connect to another user by username, but rather to a channel by reference to a channel name. Indeed, IRC users do not even have their own registered name. A user's input within a channel is identified only by an arbitrary nickname, which is good only as long as the user remains connected to the channel. A user does not own a nickname. As long as a nickname is not in current use, then anyone can use it. Thus, IRC is even more anonymous than most IM systems. (There was a registry of IRC nicknames, nickserv, but its use was voluntary. A user did not need to register his nickname; channels did not check the registry. Such a voluntary registry had so little value that nickserv has been down since the spring of 1994 and no one has seen fit to establish a replacement.)

There are also Web-based clients for IRC. Like Web-mail servers, these are servers that turn two-tier client/servers into three-tier. The real IRC client operates on a server and then is accessed by a WWW client (i.e., a browser). This means that a user need not have the ICQ client on his own system, but can access IRC from more places and more information will appear in the clear in the "network."

Lotus Sametime Connect

The Lotus Sametime Connect system is offered for enterprise IM and offers such features as exploitation of an existing enterprise directory (Notes server) and end-to-end encryption with key management (based on Lotus Notes public key infrastructure). In addition to text, Sametime supports voice and image.

NetMeeting

NetMeeting (NM) is a full-function collaboration client. While NM uses directories to resolve addresses, it usually operates peer-to-peer in a single network address space (or across address spaces via a proxy). In addition to chat, NM supports voice-chat, moving image, whiteboard (think graphical chat), file transfer, application sharing, and even desktop sharing.

Yahoo!

Yahoo! Messaging is Web based, consumer oriented, and public. It supports both user-to-user messages and chat rooms. There is a user registry but no public user directory; and there is a big directory of chat rooms.

MS Windows Messenger

Windows Messenger is the integration of IM into the MS Windows operating system. It uses the .Net Passport server to register users under their e-mail addresses or a local directory to register them under their usernames. Many of the features of NetMeeting (e.g., file send and receive, voice, video, whiteboard, application sharing, and desktop sharing) are integrated into the Messenger client function.

Others

Additional IM systems include Jabber (enterprise IM), businessim, Akonix (a gateway for enterprise use of public IM), 12planet (enterprise chat server), e/pop (enterprise), and GTV (enterprise IM with public gateway).

Operation

This section describes typical IM operations.

Installing the Client

For most users this is a necessary step and is usually as simple as clicking on an icon and responding to one or two prompts. Most IM clients are included in some other operating system or application the user already has. However, one may have to locate the client of choice in the Internet and download a copy. If one is an AOL or MSN user, IM is included in the clients for these networks. (Sometimes, the issue is getting rid of one of these.) The user may be prompted to set one or two global options at installation time.

Starting the Client

Starting the client is usually as simple as clicking on an icon. IM clients are often in the start-up list and many will try to put themselves there at installation time.

Sign-up

For many systems, new users must register their user IDs, "screen-names," handles, or aliases. In consumer systems, this may be as simple as responding to a prompt or two from the client program. In enterprise systems, it may be automatic for those who are already in the employee or user directory but may involve completing and signing a form and getting management approval for those who are not.

Populating Contact Lists

A sometimes necessary and always useful step is to populate one's contact or buddy list. This is usually as simple as entering the contact's username. Optionally, users can be organized into groups. Most clients will check usernames against the registry and report names that the registry does not recognize.

Connection

Connecting the client to the service is usually as simple as starting the software. It may even be automatic at system start-up. The client and server look to one another like an IP address and a port number. For most consumer and enterprise systems, this information is embedded in the client software and not visible or meaningful to the user. For IRC networks or multi-network clients, it may involve identifying and entering an IP address.

Log-on

IM services may require the user to log on with his handle. Client applications usually remember this value so that it can be selected from a drop-down list or entered by default. Most IM services also expect a passphrase. Again, clients usually include the ability to remember passphrases and enter them automatically. The security implication should be clear. Log-on to IM services is unusually persistent; in most systems it does not time-out.

> weemanjr (a.k.a. Tigerbait, Gatorbait, or Bitesize) recently visited me. He used my laptop and client software to log on to AOL IM. In fact, he did it so often that he set the default screen name to weemanjr, stored his passphrase, and set the client to log him on automatically. While I cannot see his passphrase, I do have beneficial use of it. Note that weemanjr might have connected from a place more hostile.

Contact Lists

Most client applications have the capability to store the names of an arbitrary number of contacts or correspondents and to organize them into folders. The collection of names of a user's correspondents is called a contact list or "buddy list." One enterprise IM system, Lotus Sametime Connect, provides two separate contact lists: one for insiders, based on the Lotus Notes directory server, and one for outsiders registered on the AOL IM server.

At log-on time, the contact list is restored to the client application. It may have been stored on the client side or the server side. Other things equal, the client side is more resistant to disclosure but not available from as many places as when stored on the server side. After the contact list is restored, it can be run against the server and the status of the each contact reflected in the client application contact list window.

I also have use of weemanjr's buddy list. It has two folders: "buddies" and "girls." The handles of the buddies suggest that they are male skateboard buddies or fellow game players. The handles of the girls suggest that they are (self-identified) flirts, flirting and gossiping being the principal activities of girls of weemanjr's age. Young people often use their birth dates to qualify otherwise common and descriptive names. Therefore, this buddy list leaks information, not only about the gender of the party, but also her age. This information suggests that weemanjr may have correspondents who do not know the code or are a little too old to interest him.

Sending Messages

When one clicks on the name or icon of a contact, the client application will attempt to open a connection to the contact; if the attempt is successful, then an application window associated with the sender will open on the receiver's system. The client application will put into the window identifying and state information. This information can include the recipient's name, online/offline, time since last activity, and, optionally, the capabilities of his client (e.g., voice, image, icon display, file send/receive).

One can type a message into the (bottom half of the) window; when new-line/return is keyed, the message is sent. All messages are displayed in the upper half of the window identified by the name of the sender.

Groups

One can invite multiple recipients to join an *ad hoc* group. A window will be opened on all participating client applications. All traffic among all participants in the group will appear in the associated window on all the windows. Each message will be labeled with the name of its sender. The group disappears when the last user leaves it.

Channels and Rooms

Channels and rooms are persistent discussions, usually associated with a topic or subject. Users can join a channel or a room at will, see all the traffic, send messages, and leave at will. Traffic can be labeled with the name of the sender. Depending on the application, the window may or may not show the handles of those connected to the channel or room; there may be unnoticed "lurkers." Channels, rooms, and their traffic may persist, even after the last user disconnects.

Sending and Receiving Files

Depending on the functionality included in the client application, one can "drag and drop" links, e-mail addresses, "emoticons" (e.g., smiley face), or other (arbitrary) objects into a connection window. If and how these appear on the recipient's system is a function of the recipient's application.

The sender drags the tag or icon of an object (e.g., program or data file) into the window representing an IM connection to another user. A window will open on the system of the receiver asking whether or not he wants to receive the file. If so, he is prompted for the location (e.g., folder or directory) in which to store it and the name to assign to it.

> Consider that weemanjr might easily have contaminated my system with a virus by accepting a file sent to him in IM.

Applications

The most general application of IM is to carry on a *conversation* between two or more people. For children, this conversation is a form of *socializing*; for adults, it might be. Subjects include current events, sports, queries, gossip, etc.

Depending on the support built into the client, many other applications can "piggyback" on (be encapsulated within) IM. For example, many clients support file transfer.

Similarly, the client can support the passing of sounds, voices, images, moving images, other arbitrary objects, applications, or even control of an entire system. The most sophisticated IM client, MS NetMeeting, supports all of these simultaneously. (NetMeeting is in a class by itself. It is so much more sophisticated than other IM clients that it is often not recognized as a member of the class.) Because the role of the server is message forwarding and addressing, no change in the functionality of the server may be required to achieve this level of sophistication.

IM for *customer and user support* has become an essential part of many business strategies. Telephone support personnel also use it as a "back-channel" to get assistance while they are talking to their customers or subscribers.

Consulting, design, and programming teams use IM for *collaboration,* even when they are all sitting around the same table. It adds so much to productivity that many of us simply refuse to work without it.

In the enterprise, IM supplements the public address, bulletin boards, and e-mail for making *announcements.* It is particularly useful for such announcements as virus warnings or weather emergencies where timeliness is essential.

Finally, IM is used for the "*grapevine,*" the alternative communication channel that most organizations resist but which, nonetheless, may be essential to their efficiency.

Capabilities

Bots

Some servers and clients support the ability to run processes other than simple addressing and forwarding. This capability exists to support easy functional extension of the application, that is, to make it easy to introduce new software. One IRC client (Bitchx) resulted from incorporating functionality added to an earlier client via a sophisticated script.

These added programs can be completely arbitrary. They can be written and instantiated by anyone with sufficient privilege or special knowledge. Those servers with this capability can be viewed as general-purpose computing engines attached to the Internet.

Most have security controls (e.g., lock-words or passphrases) to prevent their being contaminated or co-opted as attack engines. However, that leaves many that can be exploited. We have seen "bot wars" in which one or more bots are used to mount exhaustive attacks against the controls of otherwise more secure bots.

Rogue hackers use IM servers to hide the origin of attacks. In one scenario, compromised systems connect to a chat room and wait for a message. The rogue hacker then connects to that room and uses it to send a message containing the time and target of an exhaustive or denial-of-service attack. Said another way, the channel or room is used to coordinate all the listening and attacking systems.

Icons

Many client applications implement the capability for one user to send another user an icon to identify the sending user's window on the receiving user's system. Because these images might be offensive, most of these applications also include the capability to control the inclusion of the icon, even to display it a few bits at a time to avoid an ugly surprise.

Vulnerabilities

The vulnerabilities of IM are not likely to surprise anyone. They are the same vulnerabilities that we see in other parts of the Internet. Nonetheless, it is useful, if not necessary, to enumerate them. They fall into the same fundamental classes.

Fundamental Vulnerabilities

Fundamental vulnerabilities are those that are inherent in the environment or the application. They do not result from any action or inaction; they just are. They can be compensated for but they cannot be eliminated.

The biggest fundamental vulnerability of IM is that it is open. It is open as to services; anyone can put one up. Networks are open as to servers; by default, anyone can add one. IM is open as to users; again, by default, anyone can enroll for a service. This makes the network vulnerable to interference or contamination and the traffic vulnerable to leakage. While it is possible to create closed IM populations or networks, such closed populations and networks are significantly less useful than the open ones. Moreover, many client applications make it easy for users and clients to create connections between two otherwise disjointed networks.

User anonymity is a second fundamental vulnerability. The use of handles or aliases is the standard in IM. The strength of the bond between these aliases and a unique identity varies from spurious to sufficient to localize errors but sufficiently loose as to effectively hide malice. This dramatically reduces user accountability and, in some cases, can be used to successfully hide the identity of responsible parties. It seems to invite malice.

Because any kind of data hiding involves prearrangement between the sender and the receiver, most traffic in the IM moves in the clear. This means it may leak in the network. While this is offset by the fact that most of the traffic is trivial, it means that, in general, IM might not be suitable for enterprise applications. Moreover, the use of IM is so casual and spontaneous that users do cross the line between trivial traffic and sensitive traffic without even realizing it.

Implementation-Induced Vulnerabilities

Implementation-induced vulnerabilities do not have to exist. They are introduced by acts, omissions, or choices of the implementers. Most are the result of error or oversight.

Most implementation-induced vulnerabilities in IM are not unique to it. They are shared with the rest of the Internet. They include poor-quality software, often not identified with its provenance. Like much of the software in the Internet, this software *does not check or control its input* and is vulnerable to contamination by that input (the dreaded buffer overflow). Like much of the software in the Internet, it contains *escape mechanisms* that enable the knowledgeable to escape the application and its controls. Many servers are vulnerable to *interference from other applications* running in the same hardware or software environment. Much of this software employs *in-band controls.*

In some services, user data, (e.g., buddy lists and directory entries) are stored on servers. This is a legitimate design choice; it makes the application more portable. For example, one can use one's buddy list from one's (wireless) PDA or from an airport or coffee shop kiosk. However, it replaces millions of little targets with two or three large ones. It magnifies the consequences of a successful attack against those servers. Such a successful attack results in the compromise of the confidentiality of large amounts of data. Some of this data may be sensitive to disclosure. For example, contact lists encapsulate information about personal associations; directory entries may contain information about personal interests, not to say compulsions. To some degree, users have not thought about the sensitivity of this information. To some extent they are willing to share it in this context. Many do not care in any case. However, some would not want to have it posted on the Internet.

Operator-Induced Vulnerabilities

To the extent that we rely on IM for anything, we rely on the operators of the servers. In some, perhaps even most, cases, we have contracts with the operators. These agreements contain the terms of service for the service; these TOS bind mostly the user. In general, the operators promise "best efforts," but to the extent we can rely on them for anything, we can rely on what the TOS promises.

However, some services (e.g., IRC) are collaborative in nature. There is no single provider to whom we can look. The network may be no stronger than the weakest server in it.

User-Induced Vulnerabilities

Similarly, the things that users do to introduce vulnerabilities should be familiar.

Weak Passwords

Although IM passwords can be attacked (on the servers) by bots, most client applications do not enforce strong password rules. By default, most IM applications permit the user to store the user's password and submit it automatically. And although most clients will automatically enter long pass-phrases, users still prefer short ones.

Use of Default Settings

Users prefer default configurations; they simplify setup and encapsulate special knowledge about the use of a product. For events such as receipt of a message, client applications seem to default to "ask." For example, if the user does not specify whether or not to receive a message, the Trillian client will ask. However, for other choices, it may not ask. The default setting is to send the message when the Enter key is pressed. This may result in the message being sent accidentally before it is reviewed. One might not even understand that there is a safer option.

Accepting Bait Objects

Users can always compromise their systems and enterprise networks by accepting bait objects. Said from the attacker's perspective, when all else fails, exploit user behavior. As we have seen, IM has grown from being text-only to include arbitrary objects. All that is necessary to compromise a user is to find bait that he does not resist. Bait for individuals may exploit knowledge of their interests. Fishing in chat rooms exploits the fact that at a big enough party, some people will eat the soggy potato chips. Every fisherman knows that if the fish are not biting, change the bait. If they still do not bite, move to a new spot. IM is a big space with a lot of fish.

Other

All lists of vulnerabilities should end with "other." Although we are pretty good at identifying broad categories of vulnerabilities, no group of people is likely to identify all the dumb things that users will do.

Issues

This section discusses some of the security-related issues surrounding IM.

Policy and Awareness

Most damage from the use of IM will be done in error by otherwise well-intentioned users. As with most technology, the problems are really people problems. If management must rely on user behavior, it is essential that it describes that behavior to users. Management may set almost any policy that it likes but it may not be silent.

One useful rule is that security policy should treat all communications media consistently. Users should be able to choose the most efficient medium for a message. They should not be forced to choose an inefficient medium simply to satisfy arbitrary rules, security or otherwise.

Efficiency

Management questions whether IM really improves productivity enough to compensate for its intrusiveness and its potential to distract users from work. It is instructive that management no longer asks the same question about the most intrusive technology of all, the telephone. In any case, it is not as if management has much choice. The pattern of growth for the use of IM is well established and is not likely to reverse, or even level off. Management had best get used to it; workers will. Workers will integrate IM into their work styles as they have the telephone, the computer, and e-mail. It will not be seen as a distraction but simply as part of the workspace.

> When I first entered business in the early 1950s, desks did not come with a telephone by default. It was a perk just to have one's name on the directory. I say "on" because it was often only one or two pages in length. There was no direct-inward-dialing (DID); all incoming calls went through the operator. Some business phones did not even have dials; the operator completed outbound calls. In the world of flat-rate telephone service, I no longer try to recover the cost of business phone calls from my clients.

Personal Use

A significant policy issue for all communications is that of personal use. Management has a fundamental responsibility to conserve the resources of the enterprise. It must instruct users as to how enterprise resources may be consumed. With regard to personal use, IM should be treated the same as the telephone or the mailroom. If management permits personal use of the telephone, then it should permit personal use of IM under similar rules.

> As recently as 20 years ago, my employer sent me a detailed accounting of all toll calls made from the phone assigned to me. I was expected to identify those that were "personal" and write a check to the cashier to cover those calls. Those of you too young to remember it will say, "How quaint." Even then, the cost of those "personal" calls was trivial when compared to the value of my time spent on them. Sometime in these 20 years, as the cost of telephone calls has plummeted, the total cost of accounting for personal use began to exceed the reduction in expenses that could be achieved, and we stopped doing that. Now, workers bring their cell phones to work and make and receive their personal calls on them.

Anonymity

As we have already noted, the use of aliases and "handles" is the default in IM. While these handles may be related to name, role, or (e-mail) address, they are often related to a persona that the user would like to project. Some users have many. Directory entries are also used, as much to project this image as to inform.

Depending on the service or environment, the handle may or may not be bound to the user's identity. For example, AOL IM users must assert a name as the destination for messages. However, AIM permits the user to assert more than one arbitrary name. However, once registered, a name belongs to the user. He may abandon it; but unless and until he does so, it is his. IRC reserves a nickname only for the life of a connection.

Visibility

The other side of anonymity is visibility — that is, how the IM system makes one known to other users. A system that hides you completely may not be useful at all. However, one that makes one very visible may leak more information than the subject realizes. If A sends a message to B, A may receive a message that says B is/is not online. If A and B are in each other's contact list, there may be information available to each about the status (online/offline, active/inactive, home/away) of the other. Many servers will return information about all of those in the user's contact list when the user registers on the server.

> When weemanjr is connected and logged on to AIM, the icon next to his name in my client lights up. If I pass my cursor over his icon, I am given information about the state of his connection, for example, whether or not he is online, how long he has been online or when he was last seen; whether he is connected via the AOL dial-up client or via the Internet, and what the capabilities of his client

are. Of course, I must know his ID, weemanjr. I might assume that his IM name is the same as his e-mail address or AOL screen name but I would be wrong. However, if one made that assumption about me, one would be correct.

Intrusion

At its best and from time to time, instant messages intrude. Although they are not as intrusive as spam, and certainly less intrusive than the telephone, they are still intrusive. Most client applications provide controls to permit the user to reject traffic from specified users; the permissive policy. Indeed, they permit the rejection of all traffic except that from specified users: the restrictive policy. In either case, some action is required on the part of the user to elect and administer the policy.

Leakage

To the extent that the enterprise worries about the security of IM, it is usually concerned with the leakage of confidential information. IM can leak information in many ways. The user can leak information inadvertently or from motives such as anger or spite. Information can leak in transmission. It can leak to privileged users of servers or from compromised servers. It can leak through directories or registries.

Note that contact lists can be stored locally or on the server. Although servers need be trusted to some degree or another, information stored there is vulnerable to leakage. The aggregation of this information on a server is a more attractive target than the individual records stored on the client side.

Enterprise IM systems will record some traffic in logs. These logs become targets and may leak information.

Wireless

Increasingly, IM includes wireless. Most Internet-enabled cell phones include an IM client, usually for AOL IM or Yahoo! There are AOL and Yahoo! clients for Palm OS and Windows Pocket PC devices. While traffic to these devices may be partially hidden by the transport mechanism, these devices do not yet support end-to-end encryption.

IM is also used over wireless LAN technology (802.11) to laptops. These devices can support both link encryption (e.g., SSL) and end-to-end encryption. Wireless LAN encryption, standard (WEP) or proprietary, may be useful or indicated where one is aware of wireless links. However, the real issue is that cheap wireless makes the transport layer unreliable. This should be compensated for by the use of end-to-end encryption.

Immediacy

When the IM "send" key is pressed, any damage that might be done has already been done. Neither the user nor management gets a second chance. Premature or accidental sends may result if the send key is the same as the return or new-line key. Some IM applications permit one to set the client preferences so that sending a message requires strong intent.

Late Binding

As we have seen, IM manifests a distinct preference for late programmability; that is, it may be easy to modify the function of the client application program. After all, much of IM was "built by programmers for programmers." One implication of this is that it is difficult to rely on consistent behavior from these offerings.

Fraud

IM, with anonymity or even without it, is used to perpetrate all kinds of scams and frauds. Users tend to believe messages that pop up on their screens, particularly if they appear to come from trusted sources. For example, a message might suggest that the recipient enter a passphrase, enter a command, or click on an icon or a link. This is a way of getting that action invoked with the identity and privileges of the recipient.

Trust

As a general rule, IM users rely upon their ability to recognize one another by content; they do not rely on the environment, and trust is not much of an issue. However, in the future, populations will be larger, and the requirement for trusted directories and registries will also be higher.

Surveillance

Management can use surveillance as a control to direct or restrain the use of communication in general and IM in particular. In some cases, it should do so. However, if surveillance of any communication medium becomes pervasive, or even routine, that will stifle its use and diminish its value. Management's interest in the content of communication must be balanced against the right of the worker to reasonable privacy.

IM is some place between telephone and e-mail in terms of spontaneity and in terms of the value and permanence of the record that it leaves. Similarly, the cost and utility of automated surveillance of IM is also between that of the telephone and that of e-mail. Those who have automated surveillance of voice telephone will certainly want to automate surveillance of IM. However, those who have not automated surveillance of e-mail will certainly not want to automate surveillance of IM.

Any record of surveillance of communication is more sensitive to disclosure than the original communication itself. It becomes a target of attack and of "fishing expeditions." Good practice suggests that such a record be used early and then destroyed.

Offensive Content

At least at the margins, society, including the Internet, contains some ugliness. IM is no exception to this. This is troubling, in part because IM is an application that children like and because its favorite application for children is socializing. Children also use IM to satisfy (sexual) curiosity that they are discouraged from satisfying in other places. They use it to practice saying things that they are inhibited from saying aloud and face-to-face.

Coupled with the routine hiding or misrepresentation of user identity (e.g., age, gender, appearance, class, role), the result is that children may be exposed to ugliness and even to seduction. One might make a case that the Internet may be safer from seduction than home, school, church, mall, or playground, but that is small comfort, particularly if it is likely.

Similar behavior or content in the enterprise may compromise the enterprise's responsibility to provide a commodious workplace. Said another way, the enterprise may be held responsible for protecting its employees from ugliness, even if they seek it out.

Discipline

IM space is very tolerant but it does have standards of polite behavior. As with any other social population, there are sanctions for violating these standards. As with any rude behavior, the first sanction is shunning by the community. Those who behave in a rude manner will find themselves "blocked," that is, ostracized.

The service provider may impose harsher sanctions. For example, AOL vigorously enforces its terms of service.

> Littleone was "in an ICQ chat room." He used language that violated the AOL terms of service. This was language that littleone was not likely to have used without the cloak of anonymity provided by IM. It was language that littleone would not want his mother to hear, from him or anyone else. His mother, the account owner, reminded him of the language after she received a call from AOL support representatives. The support reps told her that if she could not clean up littleone's act, they would cancel her account.

While one cannot be completely banned from IRC, channel owners can and do block rude users by IP address. They have been known to ostracize entire domains or address ranges in order to enforce their standards of behavior.

Enterprise management exercises a great deal of power and discipline. IM is a part of the workplace and management is responsible and accountable for what happens there. Because management can be held accountable for some user IM behavior, it must exercise some control. At a minimum, management must tell workers

what use is appropriate and what is not. As with any other security violation, management can use disciplinary measures — from reprimand to termination.

Controls

As you might expect, IM comes with controls that can be used to protect its users and its traffic. The user, parents and guardians, or managers can use these features to manage risk. However, keep in mind that IM is inherently high risk and will usually remain so even with the prudent application of these controls.

Enrollment

Many IM systems require a user to register a unique public identifier. Other users will use this identifier to address messages to him. The service will use this identifier to find the network address to which to send the messages. At the same time, the user may be required to exchange a secret with the service. This passphrase will be used to authenticate the user to ensure that the service sends messages to only the party intended by the sender.

While some systems will accept only one enrollment from those who are already its users, most will permit an arbitrary number from just about anyone.

Directories

Services may maintain a directory of users and their addresses. Users can use this directory to locate the identifier of those to whom they wish to send a message. In many public systems, the information in the directory is supplied by the user and is not reliable. Some service providers may use account and billing information to improve the association between a user identifier and, for example, a real name and address. For example, AOL maintains a directory of its users. Access to this directory is available to AOL subscribers. AOL permits subscribers to limit access to their own directory entries. In private systems, management may own the directory and ensure that all users are authorized, properly named, and that any descriptive information (e.g., department, function, or role) in the directory is reliable.

Identification and Authentication

Most IM applications provide controls that can be used to identify and authenticate senders and recipients. Most permit both the identifier and the passphrase to be of a length sufficient to make identity both obvious and difficult to forge. However, many implement a preference for connectivity over security; that is, they start, connect, and even log on automatically. This recognizes that value goes up with the number and persistence of connections. It requires that the password or passphrase be stored locally. Because the value of connectivity is so high, the connection does not time out. Thus, once the machine has been properly initialized, the connection(s) and the identity are available to anyone with access to the machine. It may not be sufficient to learn the passphrase but it is sufficient to use it for a while. Of course, it is very difficult to protect a system from someone who has physical access to it in a running state, so this is as much a physical security issue as an I&A one.

Thus, passwords resist attack on the server at the expense of requiring that the desktop be supervised or that the screen and keyboard time out while maintaining the connection (as with Windows NT or 2000).

On the other hand, storing passwords and entering them automatically means that errors and retries do not rise (rapidly) with length. Long names make identity more patent and reduce addressing errors. Long passphrases resist exhaustive and guessing attacks.

Although passwords are the only authenticators supported by IM programs, these can be complemented by any strong authentication methods used on the client machine. For example, if the BIOS and OS passwords are used, then these protect the stored IM password.

Preferences

Client applications enable the user to specify preferences. Many of these are security relevant. The user may be able to specify what is to happen at system start, at client start, at connect, and on receipt of a message. For

example, the user may say start the client at system start, connect and log on at application start, load contact list and contact status at application start, and then set "away" status and default away message. The user may be able set alarm events, sounds, and actions. He may be able to specify events and messages to log, where to store the log, and what program to use to view it (e.g., Notepad, Excel). The user may be able to specify the default directory for storing received files. He may be able to specify whether to accept icons automatically, never to accept them, or to ask the user.

Blocking

IM applications provide the user with the ability to block messages from all users by default and from specified users. Blocking reduces the chances of intrusion, harassment, or offensive content.

Blocking at the client is based on sender name. It is used to protect the recipient from intrusion, ugliness, and spam. By default, a message from a sender not in the recipient's contact list may be blocked; the user will be asked if he wishes to receive the message and add the sender to the contact list.

Blocking can also be done at the enterprise perimeter or server. Here it can be based on sender name or recipient name. Sender name blocking works as above. Blocking on recipient name might be used as an upstream control to protect the recipient from a denial-of-service attack where the sender name is randomized. Products are available for centralized administration of blocking across a network or a user population.

Direct Connection

Some client applications enable users to connect directly to one another so that the traffic does not go through the server and cannot be seen by the privileged users of that server.

Encryption

Similarly, some enterprise IM client applications enable users to encrypt their communications. Many IM applications encrypt using (one-way) SSL user-to-server and server-to-user. This implementation requires that the message be decrypted from A's key and re-encrypted under that of B at the server. This means that the server must be trusted not to leak the message content. The IM server is trusted to some degree in any case; within the enterprise, it may be highly trusted. The advantage of this system is that information can be encrypted many-to-many between non-peer clients. The only requirement is that all clients support SSL.

A few products enable traffic to be encrypted end-to-end but only to peer systems. For example, Trillian Professional clients can communicate directly and encrypt their sessions end-to-end. Although this requires an extra election on the part of the users and a little additional setup time, it does lower the risk of leakage between the systems. Lotus Sametime Connect uses the Lotus Notes PKI to automatically create end-to-end IM sessions between two or more users within the enterprise while permitting unencrypted sessions to other users registered on the AIM server outside the enterprise.

Logging

Enterprise IM clients and services offer logging capabilities, including logs that are made at the server and are not under the control of the user. This permits the traffic to be audited for evidence of information leakage, fraud, harassment, or other prohibited activity (e.g., order solicitation by stockbrokers, prohibited use of healthcare information). Although it might be possible to log telephone traffic in a similar way, the cost of auditing those logs would be prohibitive. As enterprises come to understand this, IM becomes not only a permissible medium for this kind of communication, but also the preferred medium.

Enterprise management should keep in mind that the value of logs decreases rapidly with time but that their nuisance value increases. Their value for ensuring that you do the right thing decreases as their potential to demonstrate that you did not do the right thing goes up. Logs may contain sensitive information and may be targets. Access controls over their use are necessary to ensure that they are useful but do not leak information.

Reporting

Enterprise IM products report both IM usage and message traffic content. Properly privileged users and administrators not only see the content of the traffic, but also can map it back to the descriptive information about the sender and recipient in the directory and registry servers. Some products permit this information to be viewed by means of a thin client (Web browser).

Auditing

Auditing can be viewed as the reconciliation of what happened to what was intended and expected. It can also be viewed as the review of the logs to understand their content. There are data reduction, analysis, and visualization products that the manager or auditor can use to help him convert the log contents into information to guide policy formation and problem remediation. These products include general-purpose tools such as sorts, spreadsheets, databases, and data-mining tools. They also include specialized tools that encapsulate special knowledge about what to look for, how to find it, and what to do with it.

Filtering

Products are available to filter messages and other data objects for keywords suggesting sensitive or inappropriate content or virus signatures. They can be used to resist information leakage and system and network contamination. For efficient use, these products require both policy (to specify what traffic should not flow) and administration (to convert that policy into rules that the filter can use). They add latency to the message flow and produce false positives that might block legitimate traffic. They are most applicable in such regulated enterprises as healthcare and financial services where not only policy but also regulations are available to guide rule writing.

As IM use increases and computers become more efficient, filter applications can be expected to become more effective and efficient.

Alarms and Messaging

Products that filter IM traffic for viruses and sensitive content will generate alarms. These alarms must be communicated to those who are in a position to initiate the necessary corrective action. Failure to respond consistently to alarms will invite or encourage abuse.

Recommendations

Like safety on the highway or security on the telephone, security in IM will be the result of the efforts of users and institutions. Because no one person or institution can achieve security by acting alone, the following recommendations are organized by role.

- General:
  - — Prefer the AOL IM registry for a reasonable balance between connectivity and order.
  - — Prefer MS NetMeeting for complete functionality and end-to-end traffic hiding.
  - — Prefer enterprise directories for reliability and authenticity.
- For enterprises:
  - — Publish and enforce appropriate policies. Consider personal use, software, and content (including threatening, sexually explicit, or ugly). Consider leakage of proprietary information.
  - — Prefer enterprise IM client and server application products.
  - — Use only management-chosen and -trusted applications, from reliable sources, and in tamper-evident packaging.

- Prefer closed networks and enterprise-managed servers for security.
- Control traffic at the perimeter or gateways; use appropriate firewalls and proxies.
- Use enterprise directories.
- Require long passphrases.
- Require or prefer direct client-to-client connections and end-to-end encryption for enterprise data.
- Log and audit traffic; except where discouraged by regulation, destroy the log as soon as the audit has been completed.
- Filter traffic where indicated by policy or regulation.

• For network and server operators:
- Publish and enforce appropriate terms of service.
- Configure servers as single application systems.
- Do not permit late changes to system; do not run script or command processors (no "bots").
- Provide secure channel for (out-of-band) server controls.
- Consider separate device for registry database.

• For users:
- Use the most functionally limited client that meets your requirements.
- Prefer popular consumer systems such as AOL, MS Messenger, and Yahoo!.
- Use the most limited settings sufficient for your intended use.
- Accept messages and other data objects (e.g., files, icons, images) only from those already known to you; block all other traffic by default.
- Choose your username(s) to balance your privacy against ease-of-use for your contacts.
- Use long passphrases to resist exhaustive attacks.
- Place only necessary data in public directories.
- Use the "ask me" setting for most preferences until you have identified a pattern of response.
- Do not accept unexpected objects; do not respond to unexpected prompts or messages.
- Do not enter objects or text strings suggested by others into your client.

• For parents and guardians:
- Know your children's contacts.
- Use blocking controls to limit the contacts of young children to people known to you.

• As children mature, balance protection against privacy.

Conclusion

IM, like much of modern technology, is an inherently risky technology. On the other hand, it is also a very productive and efficient technology. As with the telephone and e-mail, its value will increase with the number of regular users. At some point it will reach critical mass, the point at which the benefit to users gives them such a competitive advantage over non-users that non-users are forced to cross over.

This year we have seen a huge increase in the number of enterprise IM products and a significant increase in the number of IM products on office desktops. The rest of us had best get ready.

As with most technology, the value of IM must be balanced against its risk, and the risk must be managed. Both management and end users must make the trade-offs between utility and security. However, we should react to this technology with prudence — not fear. IM will become part of our economic infrastructure as it has become part of our social infrastructure. We should build it accordingly. Modern enterprise IM tools provide the enterprise with valuable tools to enable them to achieve a reasonable balance between risk and reward.

Most enterprises will decide to rely on users to manage the content of IM the way that they rely on them to manage the content of phone calls, e-mail, and snail mail. Some will prefer this medium because it can leave a usable record. A small number will elect to use automated recording, surveillance, and filtering to demonstrate efforts to comply with contracts or government regulations. We should use these tools where there is a genuine requirement. We should resist the temptation to use them simply because they are cheap.

E-mail Security

Bruce A. Lobree

WHEN THE FIRST TELEGRAPH MESSAGE WAS FINALLY SENT, THE START OF THE ELECTRONIC COMMUNICATIONS AGE WAS BORN. Then about 50 years ago, people working on a mainframe computer left messages or put a file in someone else's directory on a Direct Access Storage Device (DASD) drive, and so the first electronic messaging system was born. Although most believe that electronic mail, or e-mail as it is called today, was started with the ARPA net, that is not the case. Electronic communication has been around for a much longer period than that, and securing that information has always been and will always be a major issue for both government and commercial facilities as well as the individual user.

When Western Telegraph started telegraphing money from point to point, this represented the beginnings of electronic transfers of funds via a certified system. Banks later began connecting their mainframe computers with simple point-to-point connections via SNA networks to enhance communications and actual funds transfers. This enabled individual operators to communicate with each other across platforms and systems enabling expedited operations and greater efficiencies at reduced operating costs.

When computer systems started to "talk" to each other, there was an explosion of development in communications between computer users and their respective systems. The need for connectivity grew as fast as the interest in it was developed by the corporate world. The Internet, which was originally developed for military and university use, was quickly pulled into this communications systems with its redundant facilities and fail-safe design, and was a natural place for electronic mail to grow toward.

Today (see Exhibit 3-1), e-mail, electronic chat rooms, and data transfers are happening at speeds that make even the most forward-thinking people wonder how far it will go. Hooking up networks for multiple-protocol communications is mandatory for any business to be successful. Electronic mail must cross multiple platforms and travel through many networks for it to go from one point to another. Each time it moves between networks and connections, this represents another point where it can be intercepted, modified, copied, or in worst-case scenario stopped altogether.

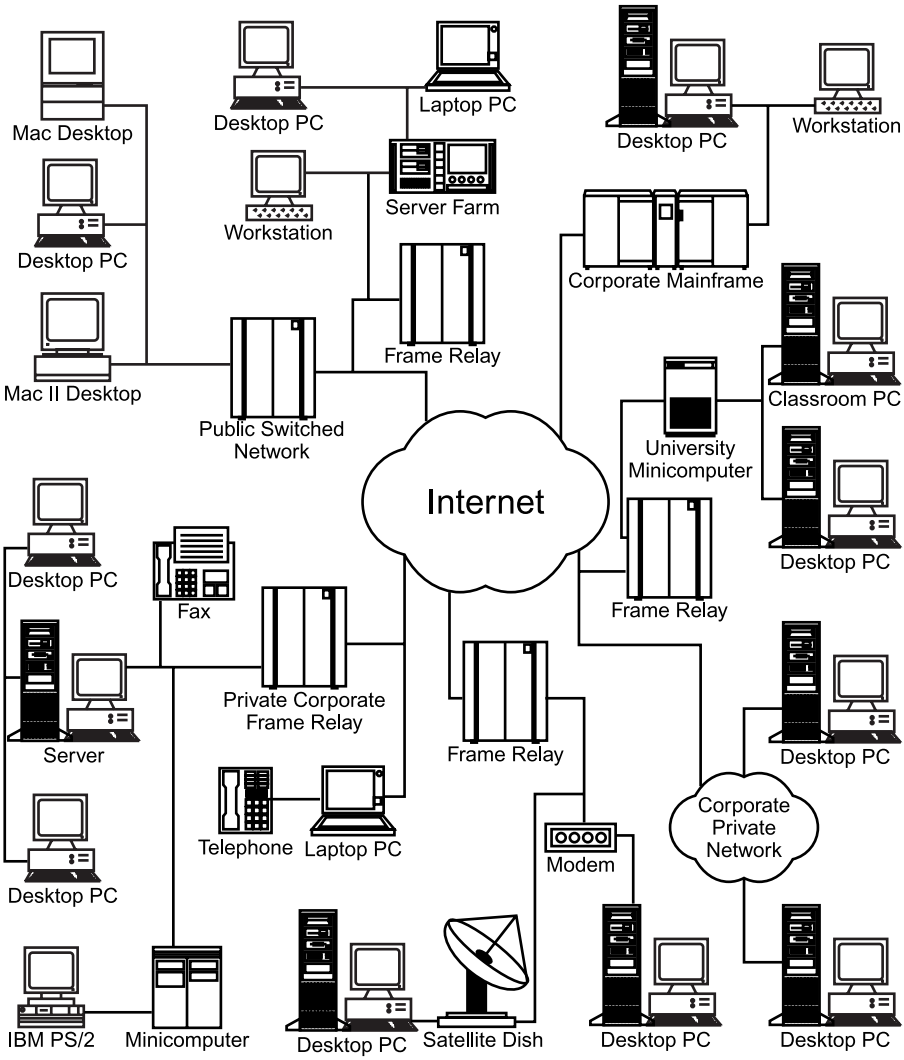


Exhibit 3-1. Internet connectivity.

Chat rooms on the Internet are really modified e-mail sites that allow multiple parties to read the mail simultaneously, similar to looking at a note stuck on a bulletin board. These services allow users to "post" a message to a board that allows several people to view it at once. This type of communication represents a whole new level of risk. There is controlling who has access to the site, where the site is hosted, how people gain access to the site, and many other issues that are created by any type of shared communications. The best example of a chat room is a conference call that has a publicly available phone number that can be looked up in

any phone book. The difference is that when someone joins the conference, the phone usually has a tone indicating the arrival of a new individual or group. With many chat rooms, no such protocol exists and users may not know who is watching or listening to any particular session if there is no specific user authentication method in use.

Today, e-mail is a trusted form of corporate communications that just about every major government and corporation in the world are using. Internal networks move communications in cleartext with sometimes little to no authentication. These business-critical messages are moved across public phone lines that are utilized for internal communications. This traffic in most cases is never even questioned as to authenticity and data can be listened to and intercepted.

Messages are presumed to be from the original author although there is no certificate or signature verifying it. By today's standards, e-mail has become a de facto trusted type of communications that is considered legal and binding in many cases. Even today, for a document to be considered legal and binding, it must contain a third-party certificate of authenticity, or some form of binding notary. However, in the case of e-mail, people consider this a form of electronic signature although it is so easy to change the senders' names without much effort. It is possible for the most untrained person to quickly figure out how to change their identity in the message and the recipient quickly gets information that may cause major damage, both financially or reputationally.

What makes matters even worse is the sue-happy attitude that is prevalent in the United States and is quickly spreading around the globe. There have already been several cases that have tested these waters and proven fatal to the recipient of the message as well as the falsified sender. These cases have taken to task system administrators to prove where electronic information has come from and where it went. Questions like who actually sent it, when was it sent, and how can one prove it, became impossible to answer without auditing tools being in place that cover entire networks with long-term report or audit data retention.

Today, e-mail traffic is sharing communications lines with voice, video, audio, and just about anything else that can be moved through wire and fiber optics. Despite the best frame relay systems, tied to the cleanest wires with the best filters, there is still going to be bleed over of communications in most wired types of systems (note that the author has seen fiber-optic lines tapped). This is not as much an issue in fiber optic as it is in copper wire. System administrators must watch for capacity issues and failures. They must be able to determine how much traffic will flow and when are the time-critical paths for information. For example, as a system administrator, one cannot take down a mail server during regular business times. However, with a global network, what is business time, and when is traffic

flow needed the most? These and many other questions must be asked before any mail system can be implemented, serviced, and relied upon by the specified user community.

Once the system administrator has answered all their questions about number of users, and the amount of disk space to be allocated to each user for the storage of e-mail, a system requirement can be put together. Now the administrative procedures can be completed and the configuration of the system can be put together. The administrator needs to figure out how to protect it without impacting the operational functionality of the system. The amount of security applied to any system will directly impact the operational functionality and speed at which the system can function.

Protection of the mail server becomes even more important as the data that moves through it becomes more and more mission critical. There is also the issue of protecting internal services from the mail server that may be handling traffic that contains viruses and Trojan horses. Viruses and Trojan horses as simple attachments to mail can be the cause for anything from simple annoyances or unwanted screen displays, all the way to complete destruction of computing facilities. Executives expect their mail to be "clean" of any type of malicious type of attachment. They expect the mail to always be delivered and always come from where the "FROM" in the message box states it came from.

The author notes that no virus can hurt any system until it is activated today. This may change as new programs are written in the future. This means that if a virus is going to do anything, the person receiving it via mail must open the mail message and then attempt to view or run the attachment. Simply receiving a message with an attached virus will not do anything to an individual's system. This hoax about a virus that will harm one's machine in this fashion is urban legend in this author's opinion.

Cookies or applets received over the Internet are a completely different subject matter that is not be discussed here. Users, however, must be aware of them and know how to deal with them. From a certain perspective, these can be considered a form of mail; however, by traditional definition, they are not.

TYPES OF MAIL SERVICES

Ever since that first message was sent across a wire using electricity, humanity has been coming up with better ways to communicate and faster ways to move that data in greater volume in smaller space. The first mainframe mail was based on simple SNA protocols and only used ASCII formatted text. The author contends that it was probably something as simple as a person leaving a note in another person's directory (like a Post-It on your computer monitor). Today, there is IP-based traffic that is moved through

many types of networks using many different systems of communications and carries multiple fonts, graphics, and sound and other messages as attachments to the original message.

With all the different types of mail systems that exist on all the different types of operating systems, choosing which e-mail service to use is like picking a car. The only environment that utilizes one primary mail type is Mac OS. However, even in this environment, one can use Netscape or Eudora to read and create mail. With the advent of Internet mail, the possibility of integration of e-mail types has become enormous. Putting multiple mail servers of differing types on the same network is now a networking and security nightmare that must be overcome.

Sendmail

Originally developed by Eric Allman in 1981, Sendmail is a standard product that is used across multiple systems. Regardless of what e-mail program is used to create e-mail, any mail that goes beyond the local site is generally routed via a mail transport agent. Given the number of "hops" any given Internet mail message takes to reach its destination, it is likely that every piece of Internet e-mail is handled by a Sendmail server somewhere along it's route.

The commercially available version of Sendmail began in 1997 when Eric Allman and Greg Olson formed Sendmail, Inc. The company still continues to enhance and release the product with source code and the right to modify and redistribute. The new commercial product line focuses on cost-effectiveness with Web-based administration and management tools, and automated binary installation.

Sendmail is used by most Internet service providers (ISPs) and shipped as the standard solution by all major UNIX vendors, including Sun, HP, IBM, DEC, SGI, SCO, and others. This makes the Sendmail application very important in today's Internet operations.

The Sendmail program was connected to the ARPAnet, and was home to the INGRES project. Another machine was home to the Berkeley UNIX project and had recently started using UUCP. Software existed to move mail within ARPAnet, INGRES, and BerkNet, but none existed to move mail between these networks. For this reason, Sendmail was created to connect the individual mail programs with a common protocol.

The first Sendmail program was shipped with version 4.1c of the Berkeley Software Distribution or BSD (the first version of Berkeley UNIX to include TCP/IP). From that first release to the present (with one long gap between 1982 and 1990), Sendmail was continuously improved by its authors. Today, version 8 is a major rewrite that includes many bug fixes and significant enhancements that take this application far beyond its original conception.

Other people and companies have worked on their versions of the Sendmail programs and injected a number of improvements, such as support for database management (dbm) files and separate rewriting of headers and envelopes. As time and usage of this application have continued, many of the original problems with the application and other related functions have been repaired or replaced with more efficient working utilities.

Today, there are major offshoots from many vendors that have modified Sendmail to suit their particular needs. Sun Microsystems has made many modifications and enhancements to Sendmail, including support for Network Information Service (NIS) and NIS+ maps. Hewlett-Packard also contributed many fine enhancements, including 8BITMIME (multi-purpose Internet mail extensions that worked with 8-bit machines limited naming controls, which do not exist in the UNIX environment) support.

This explosion of Sendmail versions led to a great deal of confusion. Solutions to problems that work for one version of Sendmail fail miserably with others. Beyond this, configuration files are not portable, and some features cannot be shared. Misconfiguration occurs as administrators work with differing types of products, thus creating further problems with control and security.

Version 8.7 of Sendmail introduced multicharacter options and macro names, new interactive commands. Many of the new fixes resolved the problems and limitations of earlier releases. More importantly, V8.7 has officially adopted most of the good features from IDA, KJS, Sun, and HP's Sendmail, and kept abreast of the latest standards from the Internet Engineering Task Force (IETF). Sendmail is a much more developed and user-friendly tool that has an international following and complies with much needed e-mail standards.

From that basic architecture, there are many programs today that allow users to read mail — Eudora, MSmail, Lotus Notes, and Netscape Mail are some of the more common ones. The less familiar ones are BatiMail or Easymail for UNIX, and others. These products will take an electronically formatted message and display it on the screen after it has been written and sent from another location. This allows humans to read, write, and send electronic mail using linked computers systems.

Protecting E-mail

Protecting e-mail is no easy task and every administrator will have his own interpretation as to what constitutes strong protection of communication. The author contends that strong protection is only that protection needed to keep the information secured for as long as it has value. If the information will be forever critical to the organization's operation, then it will need to be protected at layer two (the data-link level) of the IP stack.

This will need to be done in a location that will not be accessible to outsiders, except by specific approval and with proper authentication.

The other side of that coin is when the information has a very short valued life. An example would be that the data becomes useless once is has been received. In this case, the information does not need to be protected any longer than it takes for it to be transmitted. The actual transmission time and speed at which this occurs may be enough to ensure security. The author assumes that this type of mail will not be sent on a regular basis or at predetermined times. Mail that is sent on a scheduled basis or very often is easier to identify and intercept than mail sent out at random times. Thieves have learned that credit card companies send out their plastic on a specific date of every month and know when to look for it; this same logic can be applied to electronic mail as well.

Which ever side one's data is on, it is this author's conviction that all data should be protected to one level of effort or one layer of communication below what is determined to be needed to ensure sufficient security. This will ensure that should one's system ever be compromised, it will not be due to a flaw in one's mail service, and the source of the problem will be determined to have come from elsewhere. The assumption is that the hardware or tapes that hold the data will be physically accessible. Therefore, it is incumbent on the data to be able to protect itself to a level that will not allow the needed data to be compromised.

The lowest level of protection is the same as the highest level of weakness. If one protects the physical layer (layer 1 of the IP stack) within a facility, but does not encrypt communications, then when one's data crosses public phone lines, it is exposed to inspection or interception by outside sources.

When the time comes to actually develop the security model for a mail system, the security person will need to look at the entire system. This means that one must include all the communications that are under one's control, as well as that which is not. This will include public phone lines, third-party communications systems, and everything else that is not under one's physical and logical control.

IDENTIFYING THE MAILER

Marion just received an electronic message from her boss via e-mail. Marion knows this because in the "FROM" section is her boss' name. Marion absolutely knows this because who could possibly copy the boss' name into their own mailbox for the purpose of transmitting a false identity? The answer: anyone who goes into their preferences and changes the identity of the user and then restarts their specific mail application.

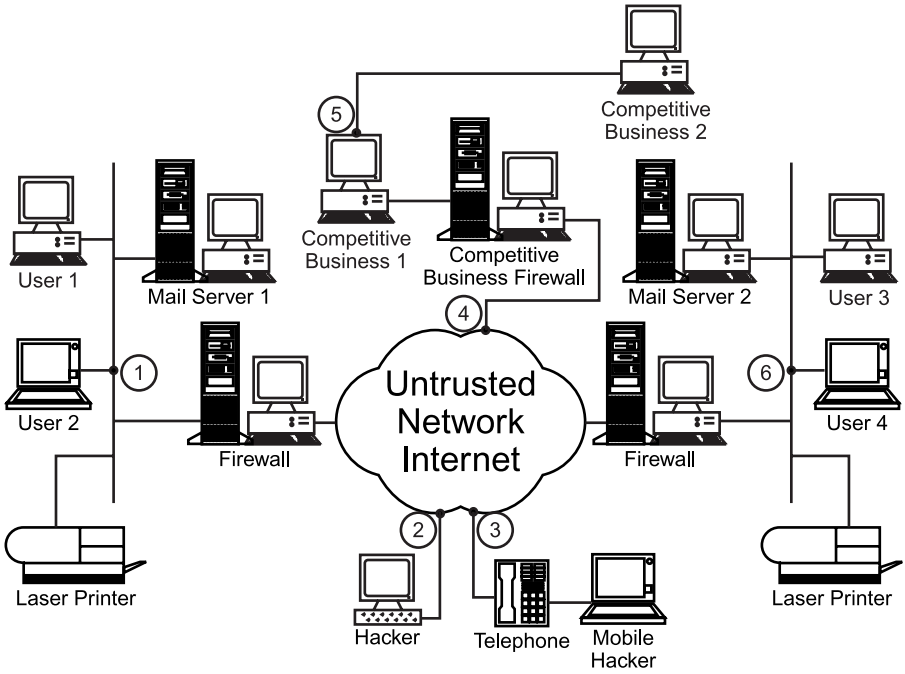


Exhibit 3-2. Unsecured network.

Whether talking about physical mail or electronic mail, the issue of authentication is an important subject. Authenticating the source of the communication and securing the information while in transit is critical to the overall security and reliability of the information being sent. In the physical world, a letter sent in a sealed, certified, bonded envelope with no openings is much safer and more reliable than a postcard with a mass mail stamp on it. So it goes with electronic mail as well.

Spoofing or faking an ID in mail is a fairly easy thing to do. Thankfully, not too many people know how to do it yet, and most will not consider it. To understand all the points of risk, one needs to understand how mail is actually sent. Not just what program has been implemented — but also the physical architecture of what is happening when one sends it.

In Exhibit 3-2, there are several points of intercept where a message can be infiltrated. Each point of contact represents another point of interception and risk. This includes the sender's PC which may store an original copy of the message in the sent box.

Network Architecture for Mail

User 1 wants to send an e-mail to User 4. If user 4 is connected to their network, then the mail will travel directly from User 1 to User 4 if all systems

between the two users are functioning correctly. If User 4 is not connected, then the mail will be stored on User 4's mail server for later pickup. If any mail server in the path is not functioning correctly, the message may stop in transit until such time as it can be retransmitted, depending on the configuration of the particular mail servers.

For mail to go from one user to another, it will go through User 1's mail server. Then it will be routed out through the corporate firewall and off to User 4's firewall via the magic of IP addressing. For the purpose of this chapter, one assumes that all of the routing protocols and configuration parameters have been properly configured to go from point User 1 to point User 4. As a user, it is presumed that one's mail is sent across a wire that is connected from one point to another with no intercepting points. The truth is that it is multiple wires with many connections and many points of intercept exist, even in the simplest of mail systems.

With the structure of our communications systems being what it is today, and the nature of the environment and conditions under which people work, that assumption is dangerously wrong. With the use of electronic frame relay connections, multi-server connections, intelligent routers and bridges, a message crosses many places where it could be tapped into by intruders or fail in transmission all together.

Bad E-mail Scenario

One scenario that has played out many times and continues today looks like this (see Exhibit 3-2):

1. User 1 writes and sends a message to User 4.
2. The message leaves User 1's mailbox and goes to the mail server, where it is recorded and readied for transmission by having the proper Internet packet information added to its header.
3. Then the mail server transmits the data out onto the Internet through the corporate firewall.
4. A hacker who is listening to the Internet traffic copies the data as it moves across a shared link using a sniffer (an NT workstation in promiscuous mode will do the same thing).
5. Your competition is actively monitoring the Internet with a sniffer and also sees your traffic and copies it onto their own network.
6. Unbeknownst to your competition, they have been hacked into and now share that data with a third party without even knowing about it.
7. The mail arrives at the recipient's firewall where it is inspected (recorded maybe) and sent onto the mail server.
8. The recipient goes out and gathers his mail and downloads it to his local machine without deleting the message from the mail server.

This message has now been shared with at least three people who can openly read it and has been copied onto at least two other points where it can be retrieved at a later date. There are well-known court cases where this model has been utilized to get old copies of mail traffic that have not been properly deleted and then became a focal point in the case.

As a security officer, it will be your job to determine the points of weakness and also the points of data gathering, potentially, even after the fact. How will one protect these areas; who has access to them; and how are they maintained are all questions that must be answered. To be able to do that, one needs to understand how e-mail works and what its intended use really was yesterday and how it is used today.

This form of communication was originally intended to just link users for the purpose of communicating simple items. It is the author's belief that the original creators of e-mail never initially intended for it to be used in so many different ways for so many different types of communications and information protocols.

Intercept point 1 in Exhibit 3-2 represents the biggest and most common weakness. In 1998, the Federal Bureau of Investigation reported that most intercepted mail and computer problems were created internally to the company. This means that one's risk by internal employees is greater than outside forces. The author does not advocate paranoia internally, but common sense and good practice. Properly filtering traffic through routers and bridges and basic network protection and monitoring of systems should greatly reduce this problem.

Intercept points 2 through 4 all share the same risk — the Internet. Although this is considered by some to be a known form of communications, it is not a secure one. It is a well-known fact that communications can be listened in on and recorded by anyone with the most basic of tools. Data can be retransmitted, copied, or just stopped, depending on the intent of the hacker or intruder.

Intercept points 5 and 6 are tougher to spot and there may be no way to have knowledge of or about if they are compromised. This scenario has an intruder listening from an unknown point that one has no way of seeing. This is to say, one cannot monitor the network they are connected to or may not see their connection on one's monitoring systems. The recipient does not know about them and is as blind to their presence as you are. Although this may be one of the most unlikely problems, it will be the most difficult to resolve. The author contends that the worst-case scenario is when the recipients' mail is intercepted inside their own network, and they do not know about a problem.

It is now the job of the security officer to come up with a solution — not only to protect the mail, but to also be able to determine if and when that

system of communications is working properly. It is also the security officer's responsibility to be able to quickly identify when a system has been compromised and what it will take to return it to a protected state. This requires continuous monitoring and ongoing auditing of all related systems.

HOW E-MAIL WORKS

The basic principle behind e-mail and its functionality is to send an electronic piece of information from one place to another with as little interference as possible. Today's e-mail has to be implemented very carefully and utilize controls that are well-defined to meet the clients need and at the same time protect the communications efficiently.

Today, there are some general mail terms that one must understand when discussing e-mail. They are Multipurpose Internet Mail Extensions (MIME), which was standardized with RFC 822 that defines the mail header and type of mail content; and RFC 1521, which is designed to provide facilities to include multiple objects in a single message, to represent body text in character sets other than US-ASCII, to represent formatted multi-font text messages, to represent nontextual material such as images and audio fragments, and generally to facilitate later extensions defining new types of Internet mail for use by cooperating mail agents.

Then there is the Internet Message Access Protocol (IMAP) format of mail messages that is on the rise. This is a method of accessing electronic mail or bulletin board data. Finally, there is POP, which in some places means Point of Presence (when dealing with an Internet provider); but for the purpose of this book means Post Office Protocol.

IP Traffic Control

Before going any further with the explanation of e-mail and how to protect it, the reader needs to understand the TCP/IP protocol. Although to many this may seem like a simple concept, it may be new to others. In short, the TCP/IP protocol is broken into five layers (see Exhibit 3-3). Each layer of the stack has a specific purpose and performs a specific function in the movement of data. The layers the author is concerned about are layers three and above (the network layer). Layers one and two require physical access to the connections and therefore become more difficult to compromise.

TCP/IP Five-Layer Stack:

1. The e-mail program sends the e-mail document down the protocol stack to the transport layer.
2. The transport layer attaches its own header to the file and sends the document to the network layer.

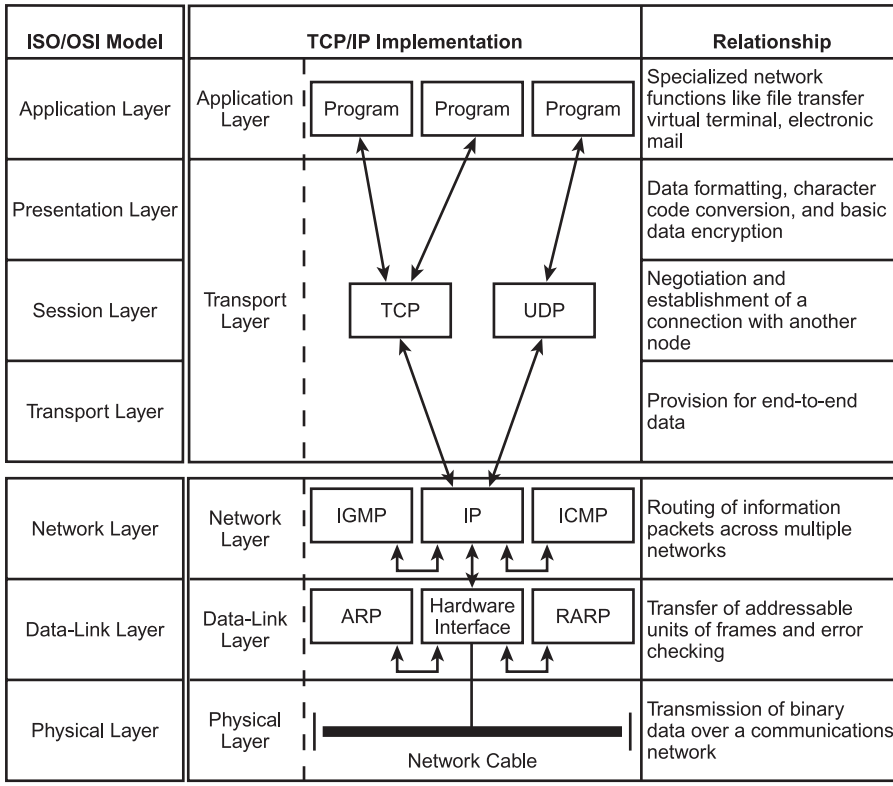


Exhibit 3-3. The five layers of the TCP/IP protocol.

3. The network layer breaks the data frames into packets, attaches additional header information to the packet, and sends the packets down to the data-link layer.
4. The data-link layer sends the packets to the physical layer.
5. The physical layer transmits the file across the network as a series of electrical bursts.
6. The electrical bursts pass through computers, routers, repeaters, and other network equipment between the transmitting computer and the receiving computer. Each computer checks the packet address and sends the packet onward to its destination.
7. At the destination computer, the physical layer passes the packets back to the data-link layer.
8. The data-link layer passes the information back to the network layer.
9. The network layer puts the physical information back together into a packet, verifies the address within the packet header, and verifies that the computer is the packet's destination. If the computer is the

packet's destination, the network layer passes the packet upward to the transport layer.
10. The transport layer, together with the network layer, puts together all the file's transmitted pieces and passes the information up to the application layer.
11. At the application layer, the e-mail program displays the data to the user.

The purpose of understanding how data is moved by the TCP/IP protocol is to understand all the different places that one's data can be copied, corrupted, or modified by an outsider. Due to the complexity of this potential for intrusion, critical data needs to be encrypted and or digitally signed. This is done so that the recipient knows who sent, and can validate the authenticity of, a message that they receive.

Encryption and digital signatures need to authenticate the mail from layer two (the data-link layer) up, at a minimum in this author's opinion. Below that level will require physical access; if one's physical security is good, this should not be an area of issue or concern.

Multipurpose Internet Mail Extensions (MIME)

Multipurpose Internet Mail Extensions (MIME) is usually one of the formats available for use with POP or e-mail clients (Pine, Eudora), Usenet News clients (WinVN, NewsWatcher), and WWW clients (Netscape, MS-IE). MIME extends the format of Internet mail.

STD 11, RFC 822, defines a message representation protocol specifying considerable detail about US-ASCII message headers, and leaves the message content, or message body, as flat US-ASCII text. This set of documents, collectively called the Multipurpose Internet Mail Extensions, or MIME, redefines the format of messages to allow:

- textual message bodies in character sets other than US-ASCII
- an extensible set of different formats for nontextual message bodies
- multi-part message bodies
- textual header information in character sets other than US-ASCII

These documents are based on earlier work documented in RFC 934, STD 11, and RFC 1049; however, it extends and revises them to be more inclusive. Because RFC 822 said so little about message bodies, these documents are largely not a revision of RFC 822 but are new requirements that allow mail to contain a broader type of data and data format.

The initial document specifies the various headers used to describe the structure of MIME messages. The second document, RFC 2046, defines the general structure of the MIME media typing system and also defines an initial

set of media types. The third document, RFC 2047, describes extensions to RFC 822 to allow non-US-ASCII text data in Internet mail header fields. The fourth document, RFC 2048, specifies various Internet Assigned Numbers Authority (IANA) registration procedures for MIME-related facilities. The fifth and final document, RFC 2049, describes the MIME conformance criteria as well as providing some illustrative examples of MIME message formats, acknowledgments, and the bibliography.

Since its publication in 1982, RFC 822 has defined the standard format of textual mail messages on the Internet. Its success has been such that the RFC 822 format has been adopted, wholly or partially, well beyond the confines of the Internet and the Internet SMTP transport defined by RFC 821. As the format has seen wider use, a number of limitations have been found to be increasingly restrictive for the user community.

RFC 822 was intended to specify a format for text messages. As such, nontextual messages, such as multimedia messages that might include audio or images, are simply not mentioned. Even in the case of text, however, RFC 822 is inadequate for the needs of mail users whose languages require the use of character sets with far greater size than US-ASCII. Because RFC 822 does not specify mechanisms for mail containing audio, video, Asian language text, or even text in most European and Middle Eastern languages, additional specifications were needed, thus forcing other RFCs to include the other types of data.

One of the notable limitations of RFC 821/822 based mail systems is the fact that they limit the contents of electronic mail messages to relatively short lines (i.e., 1000 characters or less) of seven-bit US-ASCII. This forces users to convert any nontextual data that they may wish to send into seven-bit bytes representable as printable US-ASCII characters before invoking a local mail user agent (UA). The UA is another name for the program with which people send and receive their individual mail.

The limitations of RFC 822 mail becomes even more apparent as gateways were being designed to allow for the exchange of mail messages between RFC 822 hosts and X.400 hosts. The X.400 requirement also specifies mechanisms for the inclusion of nontextual material within electronic mail messages. The current standards for the mapping of X.400 messages to RFC 822 messages specify either that X.400 nontextual material must be converted to (not encoded in) IA5Text format, or that they must be discarded from the mail message, notifying the RFC 822 user that discarding has occurred. This is clearly undesirable, as information that a user may wish to receive is then potentially lost if the original transmission is not recorded appropriately. Although a user agent may not have the capability of dealing with the nontextual material, the user might have some mechanism external to the UA that can extract useful information from the material after the message is received by the hosting computer.

There are several mechanisms that combine to solve some of these problems without introducing any serious incompatibilities with the existing world of RFC 822 mail, including:

- *A MIME-Version header field,* which uses a version number to declare a message to be in conformance with MIME. This field allows mail processing agents to distinguish between such messages and those generated by older or nonconforming software, which are presumed to lack such a field.
- *A Content-Type header field,* generalized from RFC 1049, which can be used to specify the media type and subtype of data in the body of a message and to fully specify the native representation (canonical form) of such data.
- *A Content-Transfer-Encoding header field,* which can be used to specify both the encoding transformations that were applied to the body and the domain of the result. Encoding transformations other than the identity transformation are usually applied to data to allow it to pass through mail transport mechanisms that may have data or character set limitations.
- Two additional header fields that can be used to further describe the data in a body include the *Content-ID* and *Content-Description header fields.*

All of the header fields defined are subject to the general syntactic rules for header fields specified in RFC 822. In particular, all these header fields except for Content-Disposition can include RFC 822 comments, which have no semantic contents and should be ignored during MIME processing.

Internet Message Access Protocol (IMAP)

IMAP is the acronym for Internet Message Access Protocol. This is a method of accessing electronic mail or bulletin board messages that are kept on a (possibly shared) mail server. In other words, it permits a "client" e-mail program to access remote message stores as if they were local. For example, e-mail stored on an IMAP server can be manipulated from a desktop computer at home, a workstation at the office, and a notebook computer while traveling to different physical locations using different equipment. This is done without the need to transfer messages or files back and forth between these computers.

The ability of IMAP to access messages (both new and saved) from more than one computer has become extremely important as reliance on electronic messaging and use of multiple computers increase. However, this functionality should not be taken for granted and can be a real security risk if the IMAP server is not appropriately secured.

The IMAP includes operations for creating, deleting, and renaming mailboxes; checking for new messages; permanently removing messages; setting and clearing flags; server-based RFC-822 and MIME, and searching; and selective fetching of message attributes, texts, and portions thereof.

IMAP was originally developed in 1986 at Stanford University. However, it did not command the attention of mainstream e-mail vendors until a decade later. It is still not as well-known as earlier and less-capable alternatives such as using POP mail. This is rapidly changing, as articles in the trade press and the implementation of the IMAP are becoming more and more commonplace in the business world.

Post Office Protocol (POP)

The Post Office Protocol, version 3 (POP-3) is used to pick up e-mail across a network. Not all computer systems that use e-mail are connected to the Internet 24 hours a day, 7 days a week. Some users dial into a service provider on an as-needed basis. Others may be connected to a LAN with a permanent connection but may not always be powered on (not logged into the network). Other systems may simply not have the available resources to run a full e-mail server. Mail servers may be shielded from direct connection to the Internet by a firewall security system, or it may be against organization policy to have mail delivered directly to user systems. In the case where e-mail must be directly mailed to users, the e-mail is sent to a central e-mail server where it is held for pickup when the user connects at a later time. POP-3 allows a user to logon to an e-mail post office system across the network and validates the user by ID and password. Then it will allow mail to be downloaded, and optionally allow the user to delete the mail from the server.

The widely used POP works best when one has only a single computer. POP e-mail was designed to support "offline" message access to increase network usability and efficiency. This means that messages can be downloaded and then deleted from the mail server if so configured. This mode of access is not compatible with access from multiple computers because it tends to sprinkle messages across all of the computers used for mail access. Thus, unless all of those machines share a common file system, the offline mode of access that is using POP effectively ties the user to one computer for message storage and manipulation. POP further complicates access by placing user-specific information in several locations as the data is stored as well.

The pop3d command is a POP-3 server and supports the POP-3 remote mail access protocol. Also, it accepts commands on its standard input and responds on its standard output. One normally invokes the pop3d command with the inetd daemon with those descriptors attached to a remote client connection.

The pop3d command works with the existing mail infrastructure consisting of sendmail and bellmail.

```
Net::POP3 - Post Office Protocol 3 Client class
          (RFC1081)
```

IMAP is a server for the POP and IMAP mail protocols. POP allows a "post office" machine to collect mail for users and have that mail downloaded to the user's local machine for reading. IMAP provides the functionality of POP, and allows a user to read mail on a remote machine without moving it to the user's local mailbox.

The popd server implements POP, as described in RFC1081 and RFC1082. Basically, the server listens on the TCP port named pop for connections. When it receives a connection request from a client, it performs the following functions:

- checks for client authentication by searching the POP password file in /usr/spool/pop
- sends the client any mail messages it is holding for the client (the server holds the messages in /usr/spool/pop)
- for historical reasons, the MH POP defaults to using the port named pop (port 109) instead of its newly assigned port named pop3 (port 110)

To determine which port MH POP, check the value of the POPSERVICE configuration option. One can display the POPSERVICE configuration option by issuing any MH command with the -help option. To find the port number, look in the /etc/services file for the service port name assigned to the POPSERVICE configuration option. The port number appears beside the service port name.

The POP database contains the following entry for each POP subscriber:

```
name::primary_file:encrypted_passwd::
          user@<client_address>::::0
```

The fields represent the following:

- name — the POP subscriber's username
- primary_file — the mail drop for the POP subscriber (relative to the POP directory)
- encrypted_passwd — the POP subscriber's password generated by popwrd(8)
- user@<client_address> — the remote user allowed to make remote POP (RPOP) connections

This database is an ASCII file and each field within each POP subscriber's entry is separated from the next by a colon. Each POP subscriber is separated from the next by a new line. If the password field is null, then no password is

valid; therefore, always check to see that a password is required to further enhance the security of your mail services.

To add a new POP subscriber, edit the file by adding a line such as the following:

```
bruce:: bruce:::::::0i
```

Then, use popwrd to set the password for the POP subscriber. To allow POP subscribers to access their maildrops without supplying a password (by using privileged ports), fill in the network address field, as in:

```
bruce:: bruce::: bruce@filteringisim.edu::::0
```

which permits "bruce@filteringisim.edu" to access the maildrop for the POP subscriber "bruce." Multiple network addresses can be specified by separating them with commas, as in:

```
bruce::bruce:9X5/m4yWHvhCc::bruce@filteringisim.edu,
          bruce@rsch.isim.edu::::
```

To disable a POP subscriber from receiving mail, set the primary file name to the empty string. To prevent a POP subscriber from picking up mail, set the encrypted password to "*" and set the network address to the empty string. This file resides in home directory of the login "pop." Because of the encrypted passwords, it can and does have general read permission.

Encryption and Authentication

Having determined what your e-mail needs are, one will have to determine how and when one will need to protect the information being sent. The "when" part is fairly straightforward, as this is set by corporate policy. If the security officer does not have the proper documentation and description of the controls that will need to be in place for electronic data transfer, then now is the time to put it together, as later will be too late. Suffice it to say that this author presumes that all the proper detail exists already. This needs to be done so the security officer will be able to determine the classification of the information that he or she will be working with for traffic to move successfully.

Encryption is a process whereby the sender and the receiver will share an encryption and decryption key that will protect the data from someone reading the data while it is in transit. This will also protect the data when it is backed up on tape or when it is temporarily stored on a mail server. This is not to say that encryption cannot be broken — it can, and has been done to several levels. What is being said is that the encryption used will protect the information long enough that the data is no longer of value to the person who intercepts it or has value to anyone else. This is important

to remember, to ensure that too much encryption is not used while, at the same time, enough is used to sufficiently protect the data.

Authentication is meant to verify the sender to the recipient. When the sender sends the message to the other party, they electronically sign the document that verifies to the person receiving the document the authenticity of it. It also verifies what the person sent is what the person received. It does not however protect the data while in transit, which is a distinct difference from encryption and is often a misconception on the part of the general user community.

Encryption

There are many books outlining encryption methodology and the tools that are available for this function. Therefore, this chapter does not go into great detail about the tools. However, the weaknesses as well as the strengths of such methods are discussed. All statements are those of the author and therefore are arguable; however, they are not conditional.

All mail can be seen at multiple points during its transmission. Whether it be from the sendmail server across the Internet, via a firewall to another corporation's firewall, or to their sendmail server, all mail will have multiple hops when it transits from sender to recipient. Every point in that transmission process is a point where the data can be intercepted, copied, modified, or deleted completely.

There are three basic types of encryption generally available today. They are private key (symmetric or single key) encryption, pretty good privacy (PGP) or public key encryption, and privacy enhanced mail (PEM). Each of these types of protection systems has strengths and flaws. However, fundamentally they all work the same way and if properly configured and used will sufficiently protect one's information (maybe).

Encryption takes the message that can be sent, turns it into unreadable text, and transmits it across a network where it is decrypted for the reader. This is a greatly simplified explanation of what occurs and does not contain nearly the detail needed to understand this functionality. Security professionals should understand the inner workings of encryption and how and when to best apply it to their environment. More importantly, they must understand the methods of encryption and decryption and the level at which encryption occurs.

Private key encryption is the least secure method of sending and receiving messages. This is due to a dependency on the preliminary setup that involves the sharing of keys between parties. It requires that these keys be transmitted either electronically or physically on a disk to the other party and that every person who communicates with this person potentially has a separate key. The person who supplies the encryption key must then

manage them so that two different recipients of data do not share keys and data is not improperly encrypted before transmission. With each new mail recipient the user has, there could potentially be two more encryption keys to manage.

This being the problem that it is, today there is public key encryption available. This type of encryption is better known as pretty good privacy (or PGP). The basic model for this system is to maintain a public key on a server that everyone has access. User 1, on the other hand, protects his private key so that he is the only one who can decrypt the message that is encrypted with his public key. The reverse is also true in that if a person has User 1's public key, and User 1 encrypts using his private key, then only a person with User 1's public key will be able to decrypt the message. The flaw here is that potentially anyone could have User 1's public key and could decrypt his message if they manage to intercept it.

With this method, the user can use the second party's public key to encrypt the private (single or symmetric) key and thereby transmit the key to the person in a secured fashion. Now users are using both the PGP technology and the private key technology. This is still a complicated method. To make it easy, everyone should have a public key that they maintain in a public place for anyone to pick up. Then they encrypt the message to the recipient and only the recipient can decrypt the message. The original recipient then gets the original sender's public key and uses that to send the reply.

As a user, PGP is the easiest form of encryption to use. User 1 simply stores a public key on a public server. This server can be accessed by anyone and if the key is ever changed, User 1's decryption will not work and User 1 will know that something is amiss. For the system administrator, it is merely a matter of maintaining the public key server and keeping it properly secured.

There are several different algorithms that can be applied to this type of technology. If the reader would like to know more about how to build the keys or development of these systems, there are several books available that thoroughly describe them.

Digital Certificates

Like a written signature, the purpose of a digital signature is to guarantee that the individual sending the message really is who he or she claims to be. Digital signatures are especially important for electronic commerce and are a key component of most authentication schemes. A digital signature is an attachment to an electronic message used for security purposes. The most common use of a digital certificate is to verify that a user sending a message is who he or she claims to be, and to provide the receiver with the means to encode a reply.

The actual signature is a quantity associated with a message that only someone with knowledge of an entity's private key could have generated, but which can be verified through knowledge of that entity's public key. In plain terms, this means that an e-mail message will have a verifiable number generated and attached to it that can be authenticated by the recipient.

Digital signatures perform three very important functions:

1. *Integrity:* A digital signature allows the recipient of a given file or message to detect whether that file or message has been modified.
2. *Authentication:* A digital signature makes it possible to verify cryptographically the identity of the person who signed a given message.
3. *Nonrepudiation:* A digital signature prevents the sender of a message from later claiming that they did not send the message.

The process of generating a digital signature for a particular document type involves two steps. First, the sender uses a one-way hash function to generate a message digest. This hash function can take a message of any length and return a fixed-length (e.g., 128 bits) number (the message digest). The characteristics that make this kind of function valuable are fairly obvious. With a given message, it is easy to compute the associated message digest. It is difficult to determine the message from the message digest, and it is difficult to find another message for which the function would produce the same message digest.

Second, the sender uses its private key to encrypt the message digest. Thus, to sign something, in this context, means to create a message digest and encrypt it with a private key.

The receiver of a message can verify that message via a comparable two-step process:

1. Apply the same one-way hash function that the sender used to the body of the received message. This will result in a message digest.
2. Use the sender's public key to decrypt the received message digest.

If the newly computed message digest matches the one that was transmitted, the message was not altered in transit, and the receiver can be certain that it came from the expected sender. If, on the other hand, the number does not match, then something is amiss and the recipient should be suspect of the message and its content.

The particular intent of a message digest, on the other hand, is to protect against human tampering by relying on functions that are computationally infeasible to spoof. A message digest should also be much longer than a simple checksum so that any given message may be assumed to result in a unique value. To be effective, digital signatures must be unforgeable; this means that the value cannot be easily replaced, modified, or copied.

A digital signature is formed by encrypting a message digest using the private key of a public key encryption pair. A later decryption using the corresponding public key guarantees that the signature could only have been generated by the holder of the private key. The message digest uniquely identifies the e-mail message that was signed. Support for digital signatures could be added to the Flexible Image Transport System, or FITS, by defining a FITS extension format to contain the digital signature certificates, or perhaps by simply embedding them in an appended FITS table extension.

There is a trade-off between the error detection capability of these algorithms and their speed. The overhead of a digital signature can be prohibitive for multi-megabyte files, but may be essential for certain purposes (e.g., archival storage) in the future. The checksum defined by this proposal provides a way to verify FITS data against likely random errors. On the other hand, a full digital signature may be required to protect the same data against systematic errors, especially human tampering.

An individual wishing to send a digitally signed message applies for a digital certificate from a certificate authority (CA). The CA issues an encrypted digital certificate containing the applicant's public key and a variety of other identification information. The CA makes its own public key readily available through print publicity or perhaps on the Internet.

The recipient of an encrypted digital certificate uses the CA's public key to decode the digital certificate attached to the message. Then they verify it as issued by the CA and obtain the sender's public key and identification information held within the certificate. With this information, the recipient can verify the owner of a public key.

A certificate authority is a trusted third-party organization or company that issues digital certificates used to verify the owner of a public key and create public-private key pairs. The role of the CA in this process is to guarantee that the individual granted the unique certificate is who he or she claims to be. Usually, this means that the CA has an arrangement with a financial institution, such as a credit card company, which provides it with information to confirm an individual's claimed identity. CAs are a critical component in data security and electronic commerce because they guarantee that the two parties exchanging information are really who they claim to be.

The most widely used standard for digital certificates is X.509. X.509 is actually an ITU Recommendation, which means that has not yet been officially defined or approved. As a result, companies have implemented the standard in different ways. For example, both Netscape and Microsoft use X.509 certificates to implement SSL in their Web servers and browsers. However, an X.509 certificate generated by Netscape may not be readable by Microsoft products, and vice versa.

Secure Sockets Layer (SSL)

Short for Secure Sockets Layer, SSL is a protocol developed by Netscape for transmitting private documents via the Internet. SSL works using a private key to encrypt data that is transferred over the SSL connection. Both Netscape Navigator and Internet Explorer support SSL, and many Web sites use the protocol to obtain confidential user information, such as credit card numbers. By convention, Web pages that require an SSL connection start with https: instead of http:.

The other protocol for transmitting data securely over the World Wide Web is Secure HTTP (S-HTTP). Whereas SSL creates a secure connection between a client and a server, over which any amount of data can be sent securely, S-HTTP is designed to securely transmit individual messages. SSL and S-HTTP, therefore, can be seen as complementary rather than competing technologies. Both protocols have been approved by the Internet Engineering Task Force (IETF) as a standard.

However, fully understanding what SSL is means that one must also understand HTTP (HyperText Transfer Protocol), the underlying protocol used by the World Wide Web (WWW). HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands. For example, when one enters a URL in the browser, this actually sends an HTTP command to the Web server directing it to fetch and transmit the requested Web page.

HTTP is called a stateless protocol because each command is executed independently, without any knowledge of the commands that came before it. This is the main reason why it is difficult to implement Web sites that react intelligently to user input. This shortcoming of HTTP is being addressed in a number of new technologies, including ActiveX, Java, JavaScript, and cookies.

S-HTTP is an extension to the HTTP protocol to support sending data securely over the World Wide Web. Not all Web browsers and servers support S-HTTP and, in the United States and other countries, there are laws controlling the exportation of encryption that can impact this functionality as well. Another technology for transmitting secure communications over the World Wide Web — Secure Sockets Layer (SSL) — is more prevalent. However, SSL and S-HTTP have very different designs and goals, so it is possible to use the two protocols together. Whereas SSL is designed to establish a secure connection between two computers, S-HTTP is designed to send individual messages securely.

The other main standard that controls how the World Wide Web works is HTML, which covers how Web pages are formatted and displayed.

Good Mail Scenario

Combining everything discussed thus far and a few practical principles involved in networking, one now has the ability to put together a much

more secure mail system. This will allow one to authenticate internal and external mail users. The internal requirements will only add one server and a router/filter outside the firewall, and the external requirements will require that there be a publicly available certificate authority (CA) for the world to access.

Now a system has been created that will allow users to segregate internally encrypted messages from externally. Each person will have two public keys to maintain:

- one that resides on the internally installed public key server
- one that resides on the external public key server

The private part of the public key pair will be a privately held key that the user will use to decrypt all incoming messages. Outside the firewall resides a server that will specifically handle all mail and will scan it for viruses and to be sure that all inbound mail is properly encrypted. If it is not, it will forward the message to a separate server that will authenticate the message to a specific user and will then scan and forward it after it has been properly accepted.

Now as we walk through the model of sending a message, no matter who intercepts it, or where it may be copied while in transit, the only place it can be understood will be at the final location of the keys. This method of PGP will not only secure the message, but it will act like a digital certificate in that the user will know limited information about the sender. If a digital signature is added to the model, then the recipient will know the source of the encryption session key. This will include the source of the digital signature and the senders' authentication information sufficiently enough to ensure that they are who they say they are.

There are many other components not discussed above that should be in place; these are outlined in the following steps. For more information, there are many books on router protocol and systems security that can be obtained at the local library.

Mail Sent Securely. The following steps break down the path with which a secure message can be sent (see Exhibit 3-4). This is a recommended method of securing all one's internal and external mail.

1. Before sending or receiving any messages, the author of the message gets a private encryption key from his private network.
2. Then the author places two public keys out on the networks. One is placed on his internal key ring and the second is placed on a public key ring. The purpose of this is to keep his internal mail private and still be able to use public-private key encryption of messages. This will also allow the author to separate mail traffic relevant to its origin.

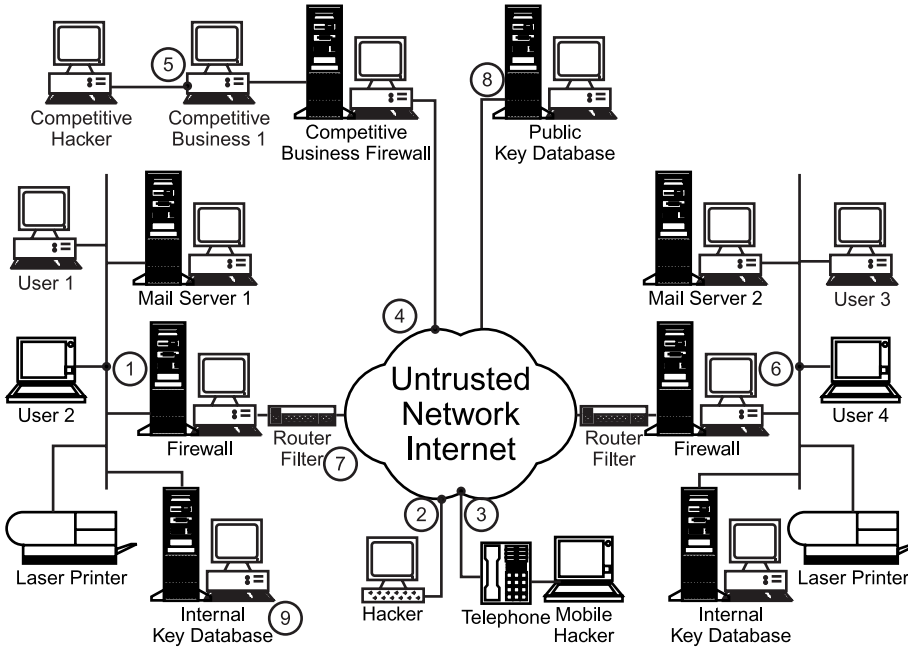


Exhibit 3-4. Secured network.

3. The author of an e-mail message logs on to his personal network and is also authenticated by the mail server via usage of a password to get ready to send electronic mail.
4. They the author composes the message using his personal mail utility that has been preconfigured with the following settings.
   a. all messages will be sent with a digital signature
   b. all messages will have receipt notice automatically sent
   c. private key encryption will be automatically utilized
5. The author signs and sends the document to a secure server. The message is encrypted and digitally signed before it leaves the author's machine.
6. The author's mail server is connected to the network with hardware-level encrypting routers that protect all internal communications. Note that the latency created by hardware-level encryption is nominal enough that most users will not notice a delay in transmission of data which is any different than already occurs.
7. The mail server determines whether the traffic is internal or external and forwards appropriately. This particular message is determined to be outbound and is therefore sent to the firewall and out to the Internet via an automated hardware encryption device.

8. In front of the firewall on the recipient's end is a hardware device that decrypts the traffic at layer three, but leaves it encrypted and signed as it was originally sent. Loss of this level of encryption is noted by the author. However, unless the outside recipient of this message has the proper hardware to decrypt the message, this level of protection will impede the communications and the recipient will not be able to read the message.
9. The message travels over the Internet. At this point, any interception that records the transmission will not assist another party in obtaining the information. To do so, they will have to:
   a. be in the line of traffic at the proper time to intercept the message
   b. have the decryption tools with which the message was encrypted
   c. have a copy or method of recreating the digital certificate if they want to modify the message and retransmit it
10. The message is then received by the recipient's firewall and allowed in based on the addressing of the message.
11. The firewall forwards the message to a mail server that quickly scans the message for viruses (this will slow down mail traffic considerably in a high traffic environment). To determine if this level of security is needed, one must determine the damage a virus or Trojan horse can do to the individual or systems to which the individual is connected.
12. The message is stored on the mail server until the recipient logs on to the network and authenticates himself to that particular server. The mail server is password protected and all data contained there will also be encrypted.
13. The mail recipient goes out to the appropriate public key server (internal for internal users and off the public key for external users) and retrieves the sender's public key before trying to open the sender's message.
14. The mail server then forwards the message to the individual user, who then opens the message after it is decrypted and verifies the signature based matching message digests.
15. Notification of receipt is automatically created and transmitted back to the original author via a reverse process that will include the recipient's signature.

The author recognizes that in a perfect world, the level of encryption that is used would not be breakable by brute force or other type of attack. The certificate and signature that are used cannot be copied or recreated. However, this is not true; it is believed that with 128-bit encryption, with an attached digital signature, the message's information will be secure enough that it will take longer to decrypt than the information would be viable or useful.

This methodology will slow down the communication of all e-mail. The return is the increased security that is placed on the message itself. There

are several layers of protection and validation that show that the message is authentic. The sender and the recipient both know who the message is from and to whom it is being sent, and both parties have confirmation of receipt.

If senders are not concerned about protecting the content of their individual messages, then the encryption part could be skipped, thereby speeding up the process of delivery. It is this author's opinion that digital signatures should always be used to authenticate any business-related or personal message to another party.

CONCLUSION

From the beginning of time, people have tried to communicate over long distances — efficiently and effectively. The biggest concern then and today is that the message sent is the message received and that the enemy (e.g., corporate competition) does not intercept a message.

From the time that the first electronic message was sent to today's megabit communications systems, people have been trying to figure out new ways to copy, intercept, or just disrupt that messaging system. The value of getting one's data is proportionately equal to the value that data has if private, and is far greater if in the corporate world.

Our challenge in today's world of computer communications — voice, video, and audio communications — is to protect it: to make sure that when it is transmitted from one specific medium to another it is received in a fashion that the recipient will be able to hear it, read it, or see it. Both the author and the recipient are confident enough that the communications are secure and reliable enough that they do not have to worry about the message not getting to where it should.

Setting up a system of checks and balances to verify transmission, to authenticate users, to authenticate messages and protect them from prying eyes becomes the task at hand for the systems administrator and the security officer. Effective implementation of encryption, digital certificates, and configuration of mail servers placed in the proper areas of a network are all components of making this happen efficiently enough that users will not try to bypass the controls.

The security officer is responsible for the information in the corporation, and becomes a security consultant by default when the architecture of a mail system is to be built. The security officer will be asked how to, when to, and where to implement security, all the while keeping in mind that one must inflict as little impact on the user community as possible. The security officer will be asked to come up with solutions to control access to e-mail and for authentication methods.

To be able to do this, the security officer needs to understand the protocols that drive e-mail, as well as the corporate standards for classification

and protecting information and the associated policies. If the policies do not exist, the security officer will need to write them. Then once they are written, one will need to get executive management to accept those polices and enforce them. The security officer will also need to make sure that all employees know and understand those standards and know how to follow them.

Most importantly, whenever something does not feel or look write, question it. Remember that even if something looks as if it is put together perfectly, one should verify it and test it. If everything tests out correctly and the messages are sent in a protected format, with a digital signature of some kind, and there is enough redundancy for high availability and disaster recovery, then all one has left to do is listen to the user community complain about the latency of the system and the complexity of successfully sending messages.

E-Mail Security

Clay Randall

THE FIRST E-MAIL APPLICATIONS WERE CREATED BEFORE ANY TYPE OF COMPUTER NETWORKS WERE IN ORDINARY USE, and thus were limited to communications between different users of a single multi-user computer system. E-mail was invented to fulfill a need for a standard, organized, and functional communications process and to prevent security problems.

Prior to e-mail applications, users would grant public access to a portion of their space so that other users could "drop off" messages and files. Users who lacked the necessary technical savvy created both non-operable conditions (insufficient privileges) and security problems (excessive privilege).

Because it was both widely desirable and applicable, some basic form of e-mail application was soon supplied as a standard component of nearly every multi-user computer operating system. As soon as computer networks became widely available, e-mail applications were adapted to be capable of exchanging e-mail between (like) systems.

The first commercially viable and widely deployed public computer networks were based on the ITU X.25 packet switching network standards. As more companies gained connectivity between sites, it quickly became useful for e-mail applications to have the ability to transport messages between computer systems over these networks — although initially the traffic was nearly exclusively intra-organizational. The early applications were not standardized to allow message transfer between different vendor's systems, and lacked appropriate management and security controls for multi-organizational environments.

The first international standard for e-mail systems using these networks was also an ITU creation: X.400 (1984, "Red Book"), and it became widely adopted among large commercial and governmental entities. (Although a smaller, more primitive form of the Internet was in existence, and had already developed e-mail standards that are still the foundation of Internet e-mail today, at the time, the public "Internet," NFSnet, specifically prohibited commercial use.)

The authors of the X.400 standard recognized the need for multiple layers of control, security, and operational organization and created a robust design hierarchically defined by country, ADMD (public operator), PRMD (private entity), organization, and organizational units. The authors of X.400 also recognized the need for minutely detailed standards to ensure interoperability between different software vendors, protocols for message format, communications between servers, communications between clients and servers, and additional features such as the ability to attach nontextual components (facsimile images, audio, video, etc.).

In addition to addressing certain weaknesses and flaws in the original version, the second iteration of the X.400 standard (1988, "Blue Book") added a vast array of optional features and a separate but related directory standard: X.500 (of which LDAP is basically a subset).

Before the newer ITU standards were widely deployed, the Internet "went commercial" and quickly overtook the established X.25 networks as the computer network of choice. For a few years, the two systems interoperated through gateway systems that translated between the two formats. Strangely, the relatively primitive and simple e-mail standards of the Internet quickly replaced the advanced and complex X.400 standard although the more sophisticated X.400 was capable of utilizing TCP/IP for transport.

The current Internet e-mail standards involve four primary areas.

SMTP (Simple Mail Transfer Protocol). Originally specified in RFC-821, and as extended with dozens of other RFCs, this protocol specifies the method for transferring messages between e-mail servers. Initially, it also specified some aspects of traffic routing, but the features of DNS as described below have replaced traffic routing. Of particular importance to security is the ASMTP (Authenticated SMTP) extension, RFC-2554, which provides a method to authenticate users submitting messages from client workstations.

"Standard for the Format of ARPA Internet Text Messages." Originally specified in RFC-822, and as extended and modified by dozens of other RFCs, this standard defines the format of the messages to be exchanged. Particularly important are the MIME (Multipurpose Internet Mail Extensions) that specify a standard method to encode multi-part message bodies, including nontextual information.

DNS (Domain Name System). The original purpose of DNS was to relate Internet IP addresses with computer names. This system was extended to aid SMTP e-mail routing. Currently, the second e-mail routing extension is in use over the Internet: MX (Mail eXchanger) records. These extensions have replaced the routing originally defined in SMTP.

S/MIME (Secure/MIME), PEM (Privacy Enhancement for Internet Electronic Mail). These standards allow for a variety of security features, including encryption and decryption of e-mail content, message integrity protection, and nonrepudiation of origin.

In addition, two standards were created to allow e-mail clients to retrieve mail from servers:

IMAP (Interactive Mail Access Protocol). This protocol defines a standard for client/server interaction between e-mail clients and servers. It is currently the *de facto* standard for open-standards e-mail systems but is also available as an alternate access method for many proprietary e-mail server systems. IMAP is designed to allow clients extensive control over the client's e-mail message store: retrieval, deletion, server-based searches, refiling messages between folders, message status, shared public (multi-user) folders, etc.

POP (Post Office Protocol). This protocol defines a standard for how e-mail clients can retrieve headers or messages from a server, and how it can request messages to be deleted from the server. While still in widespread use, it is currently relegated to minimal client and server implementations, and is being overtaken in robust systems by IMAP.

Importantly, neither of these protocols provides a method for submitting new messages for delivery. E-mail clients based on these standards utilize SMTP for submission of new messages.

GOALS AND NON-GOALS

It is important to consider what basic design goals are important to an effective e-mail system so that the security policies, plans, techniques, and devices do not unduly limit the functionality or prevent ease of use of the application.

Obviously, e-mail is intended to provide communication between users; and, as with any application, ease of use and reliability are important. From the very earliest, e-mail applications included three basic elements still found in all current e-mail applications:

- *Standard format*. A standard message format allows any user to exchange messages with any other user.
- *Organization*. All messages include fields such as originator (from), recipients (to, and possibly cc or bcc), submission date, and subject.
- *Security*. Users can only read their own mail, and messages they create are identified as originating from their accounts.

Current e-mail systems improve the three original elements in many ways, and have added only two new basic elements:

- *Interoperability*: The ability to exchange messages between networks of individual computer systems.
- *Transport of nontextual information*: The capability to include or attach computer data types such as audio, video, static images, databases, spreadsheets, executable files or scripts, etc.

Unfortunately, these last two goals are often in direct conflict with security.

To begin the list of security goals, there are common elements with most computer security areas:

- Control access to computer resources so that only legitimate users can access systems and services.
- Prevent loss of or damage to data.
- Prevent theft of data or services.
- Prevent inappropriate dissemination of data.
- Monitor for compliance with law or organizational policies.

RISKS AND PROBLEMS SPECIFIC TO E-MAIL COMMUNICATIONS

In general, e-mail systems need to allow the users in an organization to communicate with users in other organizations over the Internet. While this will ultimately require communications of e-mail messages between the Internet and the organization's e-mail servers, it does not require direct network connectivity between those e-mail servers and the Internet.

To limit network connectivity from the Internet to an organization's e-mail servers, one will have to have a standard "bastion" network between the Internet (or other insecure network) and the organization's internal network, and a mail relay device will need to be installed on the bastion network (see Exhibit 13-1).

While the exterior firewall will provide some protection to the e-mail relay system, it must allow some communications between the e-mail relay and external servers. Hackers will have the opportunity to attempt attacks through the e-mail channels provided. The protections provided by implementing the relay system in the bastion network include the following:

- Because it is the only system that can be directly attacked from the Internet, intrusion detection efforts can be focused on that system, while there may be multiple e-mail servers on the internal network.
- If compromised, the relay system contains only transient messages.
- Denial-of-service attacks launched against the relay may not prevent intra-organizational traffic from functioning normally.

In general, the attacker will only be able to do limited damage and disrupt service between internal users and external users. The hacker will need to have the ability to fully compromise the relay server, and will need

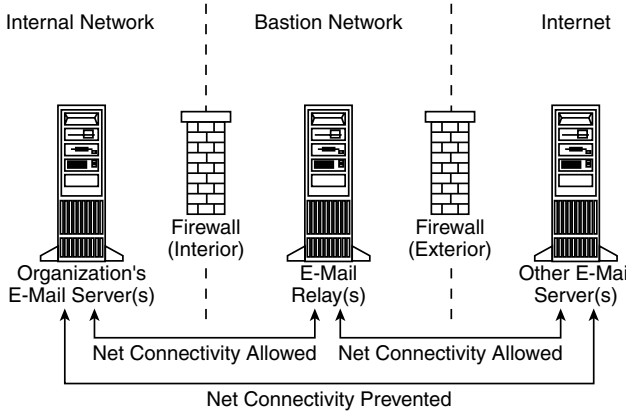


Exhibit 13-1. Limiting network connectivity from the Internet to e-mail servers.

to spend the time and effort to do so before being able to use it as a platform to directly attack the internal mail servers.

Some firewall vendors provide a similar functionality within a single firewall. When this capability is implemented, the firewall itself assumes the role of the e-mail relay. While not as robust a solution as a functionally separate relay system residing within a bastion network, it is quite superior to allowing direct network communications between the insecure network and the internal mail servers.

In many cases, e-mail messages traveling over the Internet will involve sensitive information that will need to be protected from third-party monitoring. With Internet connectivity, there is only one ready solution: encryption. Unfortunately, there exist multiple competing standards for e-mail encryption, and none of the standards are currently widely deployed. Depending on the amount of information and the distribution of affected users, there are several approaches to performing the encryption.

The greatest security can be achieved by utilizing encryption that occurs within each user's e-mail client software. As shown in Exhibit 13-2, each message is encrypted within the sender's system, and remains encrypted until it reaches the client software of the receiver's system. Unfortunately, there are many problems associated with this approach:

- Encryption only occurs when the sender remembers to activate the feature.
- The users of the different organizations must agree on utilizing the same encryption schemes (S/MIME, PGP, etc.).
- The user's and client software at each end must be able to exchange information about the encryption key(s) to use.

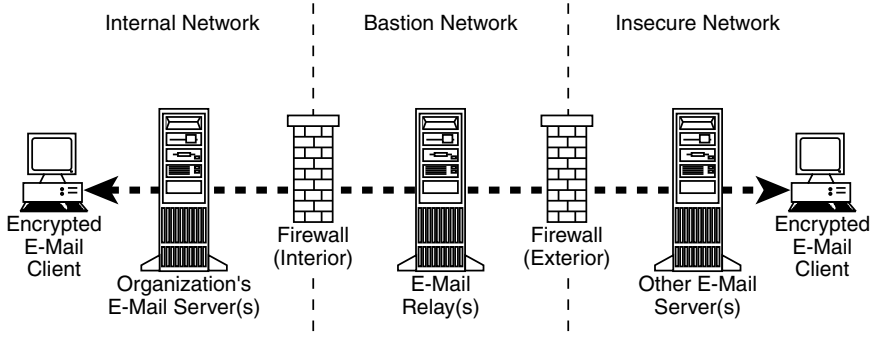


Exhibit 13-2. Encryption.

In cases where the Internet is used to provide network connectivity between geographically separate offices of the same or related organizations, an encrypted VPN can provide the protection necessary for intra-organizational traffic. It then simply becomes necessary to ensure that the routing of the e-mail traffic between the sites occurs through the VPN.

In cases where communications between two business partners' systems require the protection of encryption, but where for some reason it is impractical to implement a VPN, a mail encrypting appliance can be installed between the internal mail servers and the insecure networks, as shown in Exhibit 13-3. Once installed, the appliance is configured to encrypt/decrypt traffic exchanged with specific configured sites, while allowing traffic to pass through nonencrypted to nonconfigured sites.

In addition to the messages passing between the servers, the security of traffic passing between the servers and the users' workstations needs to be considered. Most e-mail application software systems have the ability to encrypt the communications channel between the client and server software. Because the use of encryption significantly increases the load on the server platforms, it is generally disabled by default. Some systems utilize proprietary encryption schemes, while others make use of existing encryption standards such as SSL/TLS.

Special attention should be given to the connectivity security for users accessing e-mail from home or while traveling. While some large organizations can economically provide private, secure remote access systems to internal network resources, organizations are increasingly utilizing the Internet as connectivity for remote users. All access methods in use (proprietary, SMTP, POP, IMAP, webmail, etc.) need to be considered when planning this encryption.

An alternative to encrypting e-mail client to server communications for remote users is the use of encryption-capable remote access servers (see

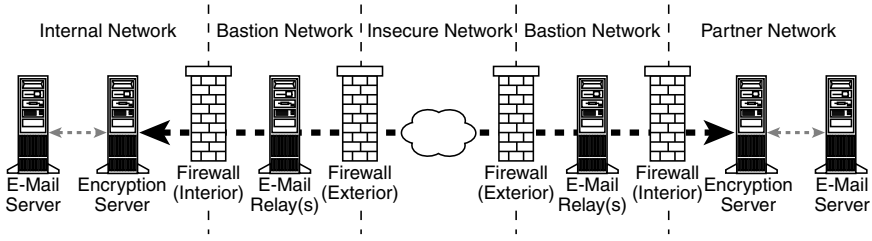


Exhibit 13-3. Installing a mail encrypting appliance.

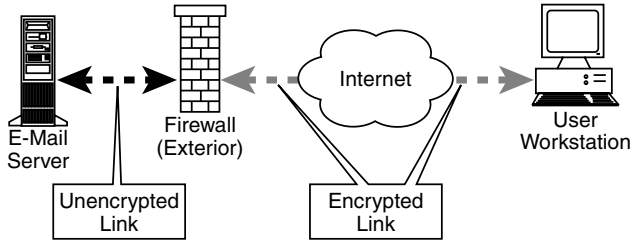


Exhibit 13-4. Encryption-capable remote access servers.

Exhibit 13-4). These devices are typically used to form encrypted tunnels directly to software installed on the user's workstation. Forming VPN tunnels to remote workstations requires the addition of remote access servers, installation and configuration of the VPN software on the clients' workstations, and the maintenance of the authentication schemes used for VPN tunnels setups. Once installed, however, it provides more than e-mail connectivity, as the remote users may then be granted access to any internal network resources.

The primary e-mail protocol utilized for exchange of traffic over the Internet is generally referred to as SMTP (Simple Mail Transfer Protocol). Originally developed in 1982, it was intended to be especially easy to implement, and was expected to be used on the (then) innocent Internet (mostly academic use). While numerous extensions and upgrades to the base protocol have become available over the years, the legacy of the original design created many security problems.

To begin with, SMTP has no mechanism for determining the validity of the originator of the message. In many systems, the indicated originator of the message is whatever the user has typed into their client software. If the user enters "George.Washington@whitehouse.gov," then that is what their client software places in the originator field. If the server receiving the message from the client system does not reject the orignator's identity,

it will probably not be rejected at any other point during the relay and delivery process. Server software should be configured to:

- *Require that the originating address of submitted messages match the identity authenticated during connection establishment from local users.* Sadly, most open systems do not do so by default, and some are completely incapable of doing so. The proprietary systems (Microsoft Exchange, Lotus Notes Domino, etc.) typically enforce the correct originating e-mail address for the user authenticated if utilizing the proprietary submission method (MAPI, etc.), but may not do so when allowing open-standards clients (SMTP/POP/IMAP) to connect.
- *Require session authentication if the originator's address of the message represents a local user.* Unless carefully configured to do so, most systems receiving messages via SMTP do not perform this check.

The answer to these specific problems it to require the use of ASMTP (Authenticated SMTP) and to verify that once enabled, the authenticated user may not submit messages with originating addresses that do not map to the user authenticated. (It is not safe to assume that one implies the other.)

Be forewarned that it may be necessary to disable the ASMTP requirement for specific IP addresses, typically for application servers that generate e-mail traffic without being able to authenticate. In these cases, the IP address of the server attempting to submit the traffic is considered to be "authentication."

While the use of ASMTP prevents the counterfeiting of messages within an organization's domain, it cannot be used to check the validity of messages being received from external organizations. In general, the most that can be done is to verify that the IP address or hostname of the server attempting to submit the traffic to one's server is "appropriate" for the domain of the message's originating address. Without prior arrangement and agreement between the organizations, any attempt to evaluate inbound message validity would be based on guesswork.

One of the best available solutions to this problem is to educate users of this situation, and to appropriately evaluate all e-mail messages received from outside domains. Even if all internal e-mail requires authentication, the credentials (typically a username/password combination) can be guessed, stolen, or otherwise compromised. (Proper user training in the use of e-mail is also discussed in several chapter sections below.)

It is also important to configure one's mail servers to prevent a condition known as "open relay." An e-mail server that functions as an open relay will accept messages with any originating address for delivery to

any recipient address. From the Internet's historical perspective, the open relay was a good samaritan that would route other organization's messages. In modern context, the open relay is an irresponsible bad citizen that allows "spammers" to operate. If allowed to continue unchecked, an organization's mail servers may become "blacklisted" and be unable to communicate to many other organizations. (More correctly known as UBE, Unsolicited Bulk E-mail, the subject of "spam" far exceeds the dimensions of this chapter.) From a security standpoint, there are three primary considerations:

- Spammers can utilize an organization's processing power and connectivity bandwidth without permission or compensation or simply "theft of service."
- If such usage remains unchecked, an organization's e-mail systems and bandwidth will be utilized to the point where the organization's use of e-mail will be degraded or become useless.
- Allowing one's systems to process the traffic causes others to hold one's organization in lower esteem and may cause public relations problems.
- To "close" one's e-mail servers as relays, they should be configured to accept traffic under only two basic conditions:
  - The system attempting to submit the message has properly authenticated as a user of one's system, and the originating address of the message matches the authenticated identity. (Unfortunately, the null originating address, "< >" is" valid for any authenticated identity as it is used for receipts, delivery notifications, etc.)
  - The system attempting to submit the message has not authenticated as a user, the originating address is from an outside domain, and all recipients of the message are inside the organization's domain.

It is not only important that the e-mail system be configured as advised by the manufacturer to close the relay capabilities, but also that the relay status of the server be tested after configuration to ensure that the recommended configuration actually closes the relay. (There are several major e-mail products currently marketed that are still partially open relays when configured as recommended. The best source of up-to-date information for the configuration necessary to close relays is available online from a variety of anti-spam organizations, such as MAPS, ORBS, and CAUCE.)

RISKS AND PROBLEMS SPECIFIC TO E-MAIL CONTENT

Certainly the most visible single e-mail security issue would be the transmission of viruses. Prior to the widespread use of e-mail, a computer

virus would often take weeks, months, or even years to cause widespread infestation. Thanks to certain automatic features of many recent e-mail clients and a little clever "social engineering" by some virus perpetrators, several recent viruses have spread worldwide in less than a day and taken down large-scale e-mail systems in minutes. Preventing the spread of viruses through e-mail systems clearly needs to be a high priority, carefully planned and implemented. (Included in this category are other programs and scripts that are not technically viruses such as Trojan horses.)

Traditionally, the approach to antivirus protection was through the installation of antivirus software on all workstations. While this is still very important, it cannot protect against all current forms of e-mail viruses. In particular, many e-mail client software packages have internal script processing and execution environments (JavaScript, VBS, etc.). Several virus varieties have exploited these capabilities within the e-mail realm in such a way as to prevent traditional antivirus software from intervening. The first large-scale example of such a virus was the "ILoveYou" virus. Through a combination of clever programming and social engineering, it triggered e-mail clients into sending copies of itself to every entry in the local user's address book. Indirectly, the volume of e-mail generated overwhelmed many e-mail servers to the level that it created a denial-of-service attack.

Another traditional approach to combating e-mail viruses has been to place a virus-scanning e-mail relay between the internal e-mail systems and the Internet. This is still an important step in a layered defense to protect e-mail systems from viral attack; it can be susceptible to the multi-server e-mail client. Most current e-mail clients, particularly those designed for open standards (SMTP/IMAP/POP) are designed to be able to interoperate with multiple accounts on multiple servers. Users often utilize this feature to cause their client to interact with both their organizational e-mail account and one or more personal accounts (home ISP, clubs, alumni, etc.). Because the client utilizes IMAP or POP to reach the remote servers, an SMTP relay with antivirus capability cannot protect these transactions. Once a virus-infected message reaches the user's workstation, it can then replicate freely inside the internal networks.

To circumvent this possibility, there are two approaches. First, the firewalls can be configured to prevent clients in the internal network from reaching external e-mail servers by blocking IMAP and POP TCP ports. This has limited effectiveness due to laptops and other portable computing devices. (A user takes his laptop home, and through his ISP connection reaches unprotected mail servers. If a virus-infected message is received, it can be triggered at a later time when the laptop is again brought into the office and reconnected to the internal network.)

The second approach is to install antivirus software that is designed to work directly with the e-mail server on every internal server. This software scans every message being submitted, preventing e-mail viruses from spreading, even after they reach the internal network. (This software scans for all virus types, not just e-mail-specific viruses.)

Because of the speed at which e-mail viruses are capable of spreading (ILoveYou spread worldwide in less than a day, and crippled some e-mail servers in minutes), it becomes necessary to find an antivirus package that can be updated with new antivirus definitions in near-real-time, preferably automatically. In evaluating antivirus solutions, it is most important to ensure that the systems intended for the e-mail servers and relays are capable of being upgraded quickly. If possible, try to select a server antivirus solution that is automatically upgraded, where the antivirus solution vendor transmits the new virus definitions to the servers. (Trying to download updates from vendor sites immediately after a major new virus strike can be problematic.)

The best approach to providing virus protection involves a layered approach:

- Workstations should have antivirus software installed and properly upgraded. While these packages may not protect against some e-mail-specific viruses, they do protect against other virus propagation methods (portable media, transmission through shared storage, etc.), and prevent some major forms of damage (formatting disks, deletion of files, etc.).
- E-mail servers within the internal network should have antivirus software designed to scan all e-mail messages (including all attachments) to protect these servers from viruses that enter into the internal networks due to portable computing and clients accessing remote e-mail accounts. It is important that this software be kept up to date with the latest virus definitions in near-real-time.
- E-mail relays passing traffic between the internal network and the Internet (or any untrusted network) should have antivirus systems to control virus transmission between internal systems and untrusted external systems. It is critically important that this system be kept updated with the latest antivirus definitions in near-real-time. In instances where there is a "storm" of virus transmission on the Internet, this system will prevent the internal e-mail servers from becoming overwhelmed with the handling (rejecting or disposing of) inbound virus-infected messages. (The relay system may become bogged down, but intra-organizational e-mail remains operable.)
- All e-mail client software used by internal users should be configured to maximal security settings to prevent autonomous virus transmission and be kept up to date with the latest security patches available

from the vendor. In many cases, the default configuration at installation is highly insecure.
- E-mail users should be trained in the various forms of e-mail viruses, and the precautions they need to employ when working with e-mail. This is particularly important in preventing the transmission of Trojan horse programs. (Never open attachments of e-mails from unknown or untrusted sources. Be suspicious of unexpected e-mail attachments from known sources. Know how to obtain assistance if a suspicious message is received.)

There is another nontechnological form of virus: the denial-of-service (hoax) virus. Typically, these consist of a detailed message describing a brand-new virus that is spread through e-mail. While the virus described may be entirely fictitious, the message is carefully crafted to strike fear in the recipient, and typically requests the reader to spread the word. Several of these hoax virus warnings have been so convincing that large numbers of users, in the interest of helping their associates, forward the message to large numbers of recipients. In effect, e-mail systems may become overloaded with the volume of these messages such that an effective denial-of-service attack is created. It is important that users know how they should react in these situations:

- Do not propagate such a virus warning to multiple recipients. If desired, instruct users to forward a single copy to a responsible party within the organization who will then evaluate the threat. (If it is a real threat, that person or organization becomes responsible for notification.)
- The user should expect to receive any real virus threat warning from a specific address within the organization, and trust only those messages.

In addition to the handling of virus-infected messages and virus warnings, users should be trained in several other aspects of e-mail. In the previous chapter section, recommendations were made to help prevent the counterfeiting of e-mail messages (also commonly known as spoofing). Typical users are usually unaware of how simple it can be to counterfeit e-mail. Users should be trained not to implicitly trust everything they receive. If they receive an e-mail that is in any way outside normal practices and procedures, the content needs to be confirmed through other channels. The implicit trust some users have otherwise placed in the validity of e-mail has caused a range of problems. Some real examples are startling.

- A cruel joke perpetrated by a fellow employee. An employee receives an e-mail addressed from their manager saying that "You're fired. Have your personal belongings removed from your desk and office and be out of the building by 5 p.m." In another case, a spoofed e-mail is sent to the corporate security department indicating that

an employee appears to be stealing office supplies, apparently originating from the employee's manager.
- A disgruntled employee creating trouble. The order fulfillment manager receives an e-mail addressed from a VP in finance indicating that all orders being shipped to a major customer should be halted until further notice due to lack of payment.
- A dishonest person (outside the company) fakes an e-mail to appear to be from a high-level marketing executive. The e-mail instructs another employee to ship 100 samples of a product to an address in another country for an upcoming trade show to be given out as samples to prospective clients. Due to an oversight, the samples need to be shipped immediately, with the usual paperwork following thereafter.

Would a manager really fire an employee via e-mail? Are these instructions/orders normally communicated with e-mail? Users need to be instructed as to which matters are routinely handled through e-mail, and when and how they should question or confirm such messages. If the receiver of the message implicitly trusts these types of messages, disastrous situations can result.

Because e-mail messages often travel in near-real-time, and most e-mail systems have very limited archiving and logging capabilities, systems and procedures will be needed for the ongoing or after-the-fact investigations. The previous chapter section gave samples of unpleasant actions involving jokes, sabotage, and theft. There are also issues of corporate espionage, sexual harassment, threats, contraband, etc. enacted through e-mail. While most major corporations have active compliance programs in place to keep users trained in appropriate behavior and usage, those problems that still occur will need to be investigated.

The reader should be aware that the following information needs to be considered in the context of possible laws regarding privacy, data retention, and encryption, which are discussed in a later chapter section.

At the very minimum, policy and procedure should be established to retain the logs of the e-mail systems for an "appropriate" time period. Any e-mail system will normally log the originator, recipients, size, and time of receipt and delivery of each message. Many e-mail servers have optional logging configurations that control the amount and types of information recorded. If applicable, the following settings should be examined and set to record the most critical information to allow investigation.

- Indication of the actual, original source of any message (independent of the "From" field). If possible, include the authenticated user. The name or network address of the workstation or server that submitted the message is important as an alternate indication or as data

corroborating originator authentication. If the submitting computer is multi-user, what account was used to submit the message?
- Subject or other headers from the message can be crucial in identifying individual messages and what route they took through various e-mail servers and systems.
- Content types and names for attachments ("customers.doc" — application/ms-word, "nudie.jpg" — image/jpeg, etc.) may be helpful.

In addition, some e-mail systems allow for archival copies of messages to be made. Where applicable, these copies retain the complete content. Some systems archive all messages transferred through a system, while others have controls that indicate which messages to archive by originator/recipient address or domain, priority, size, or other factors. If archiving features are available, they may impart a large additional storage requirement on the e-mail systems, so it will be necessary to determine organizational needs, priorities, and budgets and balance them with potential security requirements.

Where archival features are not available, systems may allow copies of messages to be automatically created and sent to (unintended) recipients. (Typically, these are BCC or auto-forward features.) These features are not typically efficient enough to be left enabled for all users, so they are generally only useful for new investigations.

If investigation is required, whether searching server logs or archived messages, it may be impractical to sift through the collected information without effective search or reporting tools. It is highly advisable to acquire and test the necessary tools ahead of time. The functionality of the tools should be verified and the resultant familiarity with the tools will save valuable time.

Within some organizations, personal use of e-mail is considered nothing more than an employee perk. If the organizational policy indicates that the use of e-mail is limited to official business, then personal use may be considered theft of service. Where this is the case, a need is created to be able to detect users' usage patterns (correspondence with non-business partners; receiving messages from entertainment, sports, or joke lists; etc.) or content types (multimedia — images, videos, audio, games, etc.). Unfortunately, e-mail server systems are typically designed for functionality and flexibility and are not designed to limit content. The ability to observe usage patterns is typically limited to post-processing of server log files. The ability to observe, filter, log, or archive traffic by content type is not available.

E-mail relay products and services designed to provide these features are available. Generally described as e-mail firewalls, their processing occurs at the application level as opposed to the network level of ordinary

firewalls. (The various features they typically supply are described in a later chapter section.) Where used, these e-mail firewalls can be effective in several areas, but are generally limited to messages passing between servers and are typically installed at the boundary between the internal e-mail systems and the Internet. Messages passing between different users of the same server are typically processed entirely internally and cannot be investigated by these devices.

WIRELESS SECURITY

Among the latest new developments in e-mail connectivity is wireless data communications. While the term "wireless" is often discussed as a single category, the various devices operate in different ways on different types of data networks, with multiple technologies for interconnecting to the Internet. The primary security concern is that the e-mail data will be intercepted either over the wireless link or over an Internet link during transfer. With the exception of pagers, most wireless data devices have some form of encryption over the wireless link. Due to the large variety of services and the quickly changing market, it will be necessary to check the specific service in question. All of these devices use the Internet for some portion of the message routing, and that portion of the routing does not inherently support encryption.

Among the current crop of wireless devices are cell phones, Internet modems, PDAs, LAN cards, and pagers.

Digital Cell Phones. While there are variations in the specifics of the network technology, these devices all utilize the networks originally designed for carrying digitally encoded two-way voice telephone calls. For data network use, once the signal reaches the cell tower, various methods are used to interconnect with the Internet. E-mail service is provided on the phone by two different methods:

- First, the cell phone can access e-mail through either HTML or WAP Webmail. In this access mode, it is functionally the same as ordinary Internet access, with the exception that it is likely that this Web access will not support SSL encryption for the connection.
- Second, some cellular service providers supply an e-mail account with the phone. In this case, the cellular provider operates the e-mail server. Normally, this is independent of an organization's e-mail security, except for the likelihood that users with these devices will want to set up auto-forwarding of some or all of their e-mail from their local account to their cell phone account. Bear in mind that the auto-forwarded messages are routed across the Internet unencrypted.

Wireless Internet Modems for Laptops and PDAs (WAN), PDAs with Built-In Wireless Modems, and Digital Cell Phones with Integral PDAs. These devices use either an independent wireless network or a digital cell phone network. Once the data passes through the wireless network onto the Internet, the data is not encrypted. Laptops can overcome this problem by utilizing SSL encryption for the connection, but the client software for PDAs is often not capable of SSL.

Wireless LAN Cards. There are a variety of these devices that use a number of different technologies. The vendor's documentation will be required to determine whether or not the transmissions are encrypted and whether or not they must be configured to enable or enforce encryption on links.

Wireless E-Mail-Specific Devices. While some other functions are typically included, their primary function is to be able to send and receive e-mail messages, and may use encrypted wireless data networks or unencrypted pager networks. All of these devices use proprietary protocols to communicate between the wireless device and the service provider. They operate in two modes:

- First, the service provider may provide a gateway that translates between the proprietary protocol of the device and open-standard protocols (SMTP/POP/IMAP) to a preconfigured Internet e-mail host. The service may or may not support SSL for the Internet connection, and may or may not have the ability to submit messages via ASMTP.
- Second, the service provider may supply a separate e-mail account for the device on an e-mail server operated by the service provider. Again, users within an organization may wish to auto-forward some or all of their e-mail to this account. The auto-forwarded mail would then travel unencrypted over the Internet to the server for this account.

Alphanumeric Pagers and Two-Way Pagers. For these devices, the paging service provider provides an e-mail address associated with the pager. When the service provider's server receives e-mail for that address, the message is typically stripped down to a minimal textual form and then transmitted to the pager. The wireless communication is typically not encrypted. Two-way pagers usually have a method to send simple reply messages.

For those devices with access methods that require that an e-mail account be forwarded to the service provider's e-mail server, care should be taken not to allow sensitive information to be auto-forwarded because it will be sent unencrypted over the Internet. If auto-forwarding cannot be configured to be selective enough, it may be necessary to disable auto-forwarding to the Internet.

Where the wireless device accesses the organization's e-mail server through the Internet and where the protection of SSL session encryption is not available, it becomes necessary to decide whether to prevent the access or to trust the user not to remotely access the e-mail server from the Internet.

E-MAIL SECURITY TOOLS

At the time of this writing (early 2001), there are primarily three categories of tools directly applicable to e-mail security:

E-Mail Encryption Systems. These devices are generally available in the form of e-mail relay devices that encrypt and decrypt traffic between configured e-mail servers. Some utilize proprietary encryption schemes, although most now utilize one of a variety of competing e-mail encryption standards (primarily S/MIME and PGP).

Due to the lack of wide-scale acceptance, the lack of a clear single standard for e-mail encryption, and various problems with the PKI infrastructure, these devices are generally only usable between systems under common control or between cooperating organizations.

Antivirus Systems Designed to Interoperate with E-Mail Servers. The vendors of e-mail servers have recognized the need for antivirus protection, but are generally not proficient in the antivirus arena. In most cases, the makers of the server software provide a published API for message processing between the acceptance and delivery phases of message processing, which one or more antivirus vendors utilize to provide server-specific products. There are also some e-mail antivirus server products designed for inline placement with the message acceptance protocol of open-standard (SMTP) systems.

E-Mail Firewall Products and Services. Both products and services may include any of a number of combinations of functions, including antivirus, anti-spam, content filtering (content search), content type filtering (attachment name or type detection), message archiving, usage pattern reporting, disclaimer notices, load limiting for defense against denial-of-service attacks, encryption, anti-counterfeiting, anti-spoofing, and user monitoring.

When planning to utilize either an encryption or firewall product or service, it will also be necessary to evaluate how it will be positioned between the affected e-mail servers. For organizations with only one server, it can be positioned between the e-mail server and the Internet as a relay. If the organization has multiple servers to be protected, then it will be necessary to determine how it can be positioned to provide the services for multiple servers while not interfering with e-mail routing.

KEEPING UP-TO-DATE

The electronic messaging environment generally changes rapidly. Every few months, vendors release new server and client software with new features that can affect security. Hackers and security experts regularly find new exploits and weaknesses of existing products, and vendors produce patches and new versions to close the gaps. Untold miscreants are actively working on the next new virus. Those individuals trying to keep up with these developments need to regularly update their knowledge in the field.

The best source of information about the most recent security and virus issues can generally be found on the Internet in the form of Web sites and mailing lists. (Of course, it is important for the surfer to evaluate the sources.) Some of the most important are:

- CERT (Computer Emergency Response Team); for general computer security issues, CERT regularly issues bulletins that include those related to e-mail and viruses.
- The vendors of e-mail client and server software are important sources of information about security issues.
- The rootshell Web site regularly has important information about hacks that can be perpetrated against services.

The vendors of antivirus software are good sources of timely information about new viruses. Be sure to check with your vendor, as many have limited access portions on their Web sites or mailing lists restricted to their customers.

SUMMARY

This chapter has focused on providing a general overview of e-mail and the security challenges it brings when used in a corporate environment. Beginning with a historical profile of electronic communications, the text investigated numerous enterprise e-mail risks, detailed the technical and operational concepts behind them, and revealed the tools and applications IT organizations can use to combat them. The text has also provided strategies for dealing with wireless e-mail security in the enterprise.

GLOSSARY

ARPAnet *Advanced Research Projects Agency NETwork* is the research network funded by the U.S. Advanced Research Projects Agency (ARPA). The precursor to today's Internet.

DNS *Domain Name System* Name resolution software that lets users locate computers on a UNIX network or the Internet (TCP/IP network) by domain name.

IMAP *Internet Messaging Access Protocol* A standard mail server expected to be widely used on the Internet. It provides a message store that holds incoming e-mail until users log on and download it. IMAP4 is the latest version.

MAPS *Mail Abuse Prevention System* A California-based, nonprofit organization dedicated to eliminating spamming by maintaining the RBL (Real-time Blackhole List). The RBL contains the IP addresses of spammers, and companies and ISPs can use the list to reject incoming mail.

ORBS *Open Relay Behavior modification System* A database for tracking SMTP servers that have been confirmed to permit third-party (open) relay of bulk e-mail messages. ORBS is a competitor of MAPS.

PEM *Privacy Enhanced Mail* A standard for secure e-mail on the Internet. It supports encryption, digital signatures, and digital certificates, as well as both private and public key methods.

POP3 *Post Office Protocol 3* A standard mail server commonly used on the Internet. It provides a message store that holds incoming e-mail until users log on and download it. POP3 is a simple system with little selectivity. All pending messages and attachments are downloaded at the same time. POP3 uses the SMTP messaging protocol.

S/MIME *Secure Multipurpose Internet Mail Extensions* A common method for transmitting non-text files via Internet e-mail, which was originally designed for ASCII text. S/MIME is a version of MIME that adds RSA encryption for secure transmission.

SMTP *Simple Mail Transfer Protocol* The standard e-mail protocol on the Internet, it is a TCP/IP protocol that defines the message format and the message transfer agent (MTA), which stores and forwards the mail.

SSL *Secure Sockets Layer* The leading security protocol on the Internet. When an SSL session is started, the server sends its public key to the browser, which the browser uses to send a randomly generated secret key back to the server in order to have a secret key exchange for that session.

TLS *Transport Layer Security* A security protocol from the IETF that is a merger of SSL and other protocols. It is expected to become a major security standard on the Internet, eventually superseding SSL. TLS is backward compatible with SSL and uses Triple DES encryption.

UBE *Unsolicited Bulk E-mail* E-mail sent to a large number of recipients without their solicitation or permission; otherwise known as spam.

VPN *Virtual Private Network* A private network that is configured within a public network that enjoys the security of a private network via access

control and encryption, while taking advantage of the economies of scale and built-in management facilities of large public networks.

X.25 The first international standard packet switching network developed in the early 1970s and published in 1976 by the CCITT (now ITU). X.25 was designed to become a worldwide public data network similar to the global telephone system for voice, but it never came to pass due to incompatibilities and the lack of interest within the United States.

X.400 An OSI and ITU standard messaging protocol that is an application layer protocol (layer 7 in the OSI model). X.400 has been defined to run over various network transports, including Ethernet, X.25, TCP/IP, and dialup lines.

PROTECTING AGAINST DIAL-IN HAZARDS: E-MAIL AND DATA COMMUNICATIONS

Leo A. Wrobel

INSIDE

The Telecommunications Privacy Policy, Tailgating, Dial-Back Modems, Securing the Mainframe, Vendor Solutions, Internet Security, Firewalls, Backup T1s

PROBLEMS ADDRESSED

With the advent of nomadic and home office environments, remote access security is once again taking its place at the forefront of security planning activities. Everyone wants an Internet presence and Internet access. Telecommuting is gaining in popularity. Sales agents armed with laptops roam the countryside.

Opening up systems to casual access by nomadic and home office workers requires the implementation of security procedures before the systems become mission critical and revenue producing. This article presents an overview of considerations to be addressed regarding dial-in and Internet access systems. Tips on how to ensure that standards for both physical equipment and privacy policies for today's mobile data world are also included. For information on protecting against dial-in hazards involving voice systems, see article 5-04-41.

PAYOFF IDEA

As more workers use E-mail and data communications, the importance of security grows, and should be firmly established before the systems are used to generate revenue. Beginning with a sound telecommunications privacy policy, organizations should implement protective measures ranging from paging systems, dialback modems, and comprehensive after-market equipment to test firewalls, fully redundant configurations, and backup T1s.

THE TELECOMMUNICATIONS PRIVACY POLICY

What happens if you read someone else's confidential E-mail? Can the company read yours? Does an employee have an absolute right to privacy? Many individuals and companies have no idea how to answer these questions.

For example, it is a violation of federal law to listen to a telephone conversation without the knowledge of the participants. We all know from television shows that there is a rigid process to secure a wire tap on a phone line. Do similar protections exist for E-mail?

Generally, a company's employee policy on E-mail privacy, usually in a telecommunications privacy document, sets the standard. Unfortunately, many organizations do not have such a document.

Every so often, a story in the paper underscores the vulnerability of E-mail far better than thousands of words by experts. The following is one example.

An office romance was blooming between two employees of a major service company. The company depended heavily on electronic mail in the conduct of daily business, and employees had every reason to believe this E-mail was secure. The young lady involved apparently thought it would be romantic to send a graphic E-mail letter, with an attached photograph, to her suitor. This would have been well and good if she had not clicked on the "All Users" button when sending the message. Suffice it to say this made for good office gossip and sent a clear message to everyone about the use of E-mail systems.

Notwithstanding such human errors, are E-mail systems really secure? Can an employer read E-mail? Do employees have a right to privacy? Article 5-04-41 discussed other forms of communication such as fax transmissions. Is a person breaking the law when he or she receives and reads a fax or E-mail intended for someone else? The answers may surprise you, and could call for a thorough review of security procedures for these systems.

A policy on telecommunications privacy should be broad enough in scope to cover not only E-mail, but voice mail and other mediums. Policies generally fit in between the following two ends of the spectrum:

- "Employees work for the company, and it owns the system. The company will listen to or monitor whatever we feel like monitoring or listening to," or
- "ABC Company is committed to absolute privacy of communications and each employee has the right to not have their communications monitored."

Which approach is right? That depends on your company. We usually opt for the latter, with a caveat, as follows:

- "ABC Company is committed to absolute privacy of communications, and each employee has the right to not have their communications monitored. However, if in the course of normal maintenance activity we inadvertently discover illegal activity, we reserve the right to report this activity to the responsible authorities."

Once again, it is wise to contact legal counsel when writing these policies. I was purposely casual in these illustrations to illustrate the range of options, but also because failure to contact legal counsel can leave organizations exposed to risk. An example of this is an employee who ran an illegal bookmaking operation out of a company system. He was fired but then reinstated because the company had no policy on privacy on which to base the dismissal. It is important to contact the corporate legal department, outside counsel, or an internal audit department for further details.

In addition to the establishment of the privacy policy, an evaluation of protective measures for dial-in lines should begin with an overview of their hazards. Any proposed solutions must address the types of intrusion discussed in the following sections if they are to ensure even a minimum level of protection.

HACKERS

Hackers are unauthorized users, often juveniles, who attempt to break into a system for kicks. They may or may not be lethal, but some rudimentary precautions can prevent these break-ins. Because these individuals often use demon dialers, which dial every number in a prefix to find modem lines (e.g., 555-0000, 555-0001, and so on), it is often not difficult for them to find numbers, especially if they are front ended with an identifying script. Therefore, security precautions must be evaluated to prevent this occurrence. These include:

- Modems that dial back the user.
- Modems that screen the CALLER ID of the calling party.
- Modems or equipment that answer initially with silence, rather than with a modem tone.
- Equipment that does not paint an initial screen, such as "Welcome to ABC Widget Company," which can serve to further encourage an unauthorized user.
- Equipment that logs and tracks unsuccessful log-in attempts.
- Equipment that requires a special hardware key to allow access.

Although none of these provides a definitive solution, several or all of these methods can provide a nearly impenetrable defense against unauthorized access.

SABOTEURS

The most unsettling types of attacks come from those who are knowledgeable of the environment. Disgruntled employees, for example, can cause more damage than anyone else, because they know exactly what attack can be the most damaging. Many organizations have a high level of employee trust, and have an established policy of allowing employees

a high degree of system access. This is commendable, but care should be taken because even the most close-knit firms can never be sure when an employee will destroy a critical system because of a personal gripe.

Recommended minimum precautions include:

- A simple process for eliminating log in access when an employee leaves the company.
- A mandatory process for eliminating log in access when any employee is terminated.

TAILGATING

Tailgating is an old ploy used to gain access to a system. It goes like this:

1. A super user or system administrator dials into a remote system.
2. The hacker dials the number (obtained through a demon dialer) and gets a busy signal.
3. The hacker dials 0 and asks the local telephone operator to verify the line.
4. The operator interrupts the line, which usually drops the authorized super user.
5. The hacker is meanwhile dialing out at the same time on another line. If timed perfectly, the modem sees the drop of carrier as a temporary line hit and reestablishes the session with the hacker's modem.
6. The hacker is online with super user access; the super user in turn oftentimes does not even know he has been dropped and instead thinks the system has simply locked up.
7. Working quickly, the hacker grabs the password files and compromises the system for his next attempt later, before the super user realizes anything was amiss. When security logs are checked, only the super user is logged because his session was never terminated.

Sounds rather ingenious? Actually, compared with some of the other tricks, this one is elementary. It underscores that any additional security precautions implemented provide greater peace of mind and protection to organizational assets. Remember, security is a major concern when dozens or hundreds of employees are accessing mission-critical systems through the public telephone network. Careful planning can avoid major difficulties later.

PREVENTIVE MEASURES

Inbound Call Accounting Systems

Each proposed solution should provide an accounting record of all call attempts to make a paper trail of dial-in access. Strength in screening, reporting, and presentation of this information must be a principal selec-

tion criterion in any protective system. A system showing 350 unsuccessful log-in attempts one night is sending a clear signal.

Paging Systems

Some systems that require a high degree of security provide automatic pager notification. When a user logs in, a system administrator's pager goes off. These can be combined with procedures for reporting mysterious login attempts that cannot otherwise be accounted for. They are not terribly expensive considering that a system administrator is instantly notified of anomalies.

Hardware Keys

Hardware devices such as hardware keys should be included in any security recommendations for mission-critical systems. Ease of use, such as plugging into a parallel port, and low cost should be both overriding criteria in the use of these devices.

The keys are a hardware device that usually plugs into a parallel port of a laptop computer. In conjunction with the attendant software, they provide a fairly bulletproof solution because an intruder would have to have both the encryption software, and the hardware key, to get even close to accessing a system.

Caller ID

Caller ID is available in many cities. Even in telephone wire centers where it is available, there are limitations. Caller ID is useful for more than just identification of annoying calls during dinnertime. Properly used, it can identify unauthorized users by their telephone number and often by name. Even nicer, caller ID is a built-in feature for many modems and ISDN (integrated services digital network) terminal adapters. The numbers can be logged on a call-by-call basis as part of the dial-in log described earlier.

Owing to the nonavailability of caller ID service in many areas, modems or other equipment that use this service as the sole underlying basis of a protective system may not be considered. Even if caller ID is available, there are still security concerns, namely:

- Caller ID data may not always be passed by interexchange carriers like AT&T, MCI, and Sprint. Your company would in a sense be vulnerable to long distance callers using carriers who do not pass this data. (This is rapidly changing as carriers comply with FCC regulations to pass caller ID data whenever possible.)
- Even if interexchange carriers were equipped to pass this data, the distant local central office might not be. A company would still be open to intrusion unless other methods were employed.

- Many local central offices in parts of the country are not caller ID capable for either local or long-distance calls.

Therefore, at least a few calls will still slip through with the "out of area" disclaimer on the modem or display device. Caller ID alternatives should be carefully considered as an exclusive security precaution until the service becomes more ubiquitous. Even after universal deployment, it is recommended that this service is used only to augment existing security measures and not as a solution. Even where caller ID is available, the user can in many cases dial the override code to block it. This demands another level of protection on a modem: rejection of users where the incoming data indicates that the caller ID information was deliberately blocked.

Dial-Back Modems

Many dial-back modems are available on the market today. These devices require that users login, and then hang up and call the incoming caller back at a predetermined number. These are fairly foolproof, but inconvenient. A nomadic user in a hotel will not have an authorized number and will not be able to dial into a call-back modem bank. Nonetheless, special modem banks and numbers can be set up for this purpose with special emphasis and screening for potential intruders.

Securing the Mainframe

Many users are stuck trying to protect legacy mainframe environments where security options for dial-in are marginal at best. While IBM has no graceful and simple solution for the mainframe, it can provide an additional level of security by front ending the protocol converter with a dial-in server. The IBM 8235 dial-in server is one candidate. It provides the necessary accounting, dial-back capability, and with an eight port maximum capacity, it seems sized correctly for any future growth. However, it is somewhat expensive.

More common are solutions where distributed devices are hung off the mainframe through the use of bridges and LAN switches. A PC-based system with appropriate protocol conversion software will often suffice in a pinch as a secure dial-in medium for the mainframe.

Software-Based Solutions

Because transparency for dial-in users is an issue (different departments often use a variety of software packages when dialing in), you may not want to consider a wholesale change of dial-in software emulation packages. This might prove disruptive to your present operating environment.

Software alternatives that augment or enhance the current hardware package in use are most preferable because the need for training on new packages is minimal.

After-Market Equipment

Often, the only way to provide acceptable security across a broad range of installed equipment and large cross-section of users is to adapt some sort of outboard solution. Naturally, the potential exists to black box a company to death by over-broadening the range of installed equipment. It pays to evaluate carefully. Following are several effective alternatives:

- A line of equipment distributed by CDI Incorporated of Clifton, NJ. This equipment seems to most adequately reflect pressing security concerns presented by most users. Although I have not had direct experience with this equipment, on paper it certainly seems to provide a most comprehensive solution to the dial-in security issue and should be carefully considered.
- Another cost-effective solution is brokered by LeeMah Data Comm Security Corp. of Hayward CA. It also appears to meet criteria for transparency and accommodation of diverse remote users.

When evaluating these or another product, look for the following features:

1. The unit should serve as security device and modem manager. Anyone who has ever repeatedly hit a "ring-no-answer" when dialing a modem pool can appreciate this feature. Make sure the system can automatically busy these lines out, then alert you to the problem.
2. The unit should provide response time information by modem, by phone line, and by port. For example, it should interface to a personal computer for effective performance management. This makes a good source of information to a help desk for when users call in to report trouble connecting.
3. The product should offer effective upgradeability. For additional security, the product should offer token hardware devices that interfaces to a user's parallel port. Software token should also be available. DOS or Windows software both should be supported.
4. Software and hardware keys. Because transparency of equipment for users is usually an issue, try not to consider major changes in hardware used by remote users. This might prove too disruptive to the present operating environment. This may cost more later in maintenance and training.

An unbiased opinion makes LeeMah the favorite in terms of flexibility and cost-effectiveness. Some of the features offered provide effective evaluation criteria for whatever system you decide to acquire. These include:

- The unit is a multiple port challenge-response unit.
- It supports up to 32 modems (Traq-Net 2032).

- The unit installs between the phone line and modem, allowing for use of present modems.
- The product allows for use of (optional) proprietary LeeMah Security Modems for additional protection.
- The product employs either a hardware or software token at user request.
- It provides a full audit trail.
- The product meets DES security standard.
- The product operates transparently, allowing for use of all present emulation software.
- It offers reasonably priced software (Infodisk).
- The product supports, for example, Procomm, Qmodem, Crosstalk, PCAnywhere, and Smartcom.

LeeMah DataComm provides a standards-based, virtually impenetrable, flexible, and configurable, security solution for the protection of remote access to telecommunications and data communications network information and resources. LeeMah's remote access security solutions consist of three elements: access control systems, personal authentication devices, and security administration software.

The LeeMah system represents one of the most adaptable and feature-rich solutions to protect dial-in services over a wide variety of equipment types from mainframes to local area networks (LANs). However, it is still wise to evaluate several vendors and base a decision on each unique environment. An Internet search will probably uncover numerous other choices with similar capabilities.

INTERNET SECURITY RESPONSIBILITIES

No discussion of unauthorized data access is complete without mentioning the Internet. The Internet is a relatively new phenomenon for many companies, at least as a revenue generating system, and many companies have unresolved organizational issues about security responsibilities. Who maintains the equipment used for Internet access? Historically, these types of operations often have fallen under a special unit in the IS department, such as midrange computer services. However, today many companies have a separate group of technologists responsible for the actual operation of the Internet firewall and other components. There is not always a clear business unit responsible for Internet security.

Many clients have reported minor snafus (i.e., holes or vulnerabilities left temporarily exposed in the system) due to lack of a clear policy outlining who is responsible for which system and under what circumstances. Although this responsibility will ultimately gravitate to an IT security group (much like the LAN and mainframe services of today), vulnerabilities will

continue in the immediate term, while the technology is in the "tweaking and tinkering" stage.

Another issue includes staffing and resource allocation. Many companies should consider a nominal increase in manpower to avoid creating too small a pool of specialists and provide better depth. When Internet access is established and any possible security breeches or holes are closed, organization changes may be readdressed. If a company has one person who is readily identified as the Internet guru, take note. These folks are in high demand and could leave you holding the bag if they accept other employment. Besides, outgunned and undermanned staffs have little time to probe for security violations.

Installation of Test Firewall

Many companies do not have firewall platform exclusively earmarked for testing and backup. For all intents and purposes, the present technology is single threaded in almost every way. This is not a major concern yet, but will be when the firewall goes into full operation, and the system becomes revenue producing.

Just as in mainframes and local area networks, it is important to establish a protocol and procedure that does not directly introduce new applications into a production environment. This lesson became apparent during 25 years of mainframe operations, and even the most renegade LAN managers have learned to adopt it as a gospel of prudent operation. Like many new technologies, these protocols have yet to catch up in the Internet arena for many firms.

A test firewall also can double as a backup in the event of a major equipment failure in the primary configuration. This will be important again when the system becomes fully revenue generating.

Because the Internet is a relatively new technology for many firms, staff should be encouraged to dabble. Although it is not prudent to experiment on a production platform, the backup firewall configuration can provide a practical option. The backup can be justified further by encouraging the staff to experiment, improve, and refine without jeopardizing operation of the enterprise. In summary, the extra expense of a backup firewall capability can be justified for the resiliency it provides the network and because it shortens the educational curve when principal technologists are encouraged to try new processes.

Upgrading to a Fully Redundant Configuration

The issue of redundant physical componentry of the Internet firewall raises several items of concern. Again, these will not be major concerns until the Internet and firewalls go into full production and become revenue-generating systems, but they will demand increased attention in the

future. The first is in the area of general fault tolerance on the physical components.

Many routers in use, such as the CISCO 4000 series, have no redundancy. The CISCO 5000 series has redundant power and a redundant CPU, which will be required later. Every other component in other systems generally has a redundant backplane, power supply, CPU, and other common logic. As usual in the world of technology, the newer systems play catch-up for a couple of years with regard to redundancy. CISCO appears to have responded commendably to user demands for such backup systems, as have other vendors. Organizations should explore these options and use them as soon as Internet access is about to become revenue generating or otherwise mission critical.

Backup T1s

Another issue to consider is that most large users install only one T1 to the Internet Service Provider (ISP), which creates a point of vulnerability. A wiser approach is to consider adding a second T1 along with "Round Robin DNS" for greater resiliency on the wide area network connectivity to the ISP. Many local telephone companies offer services designed to diversify T1 access as well. In Southwestern Bell territory, the service is called SecureNet™, which offers a completely diverse T1 circuit at a significantly reduced rate. Other components, such as CSUs and DSUs, are single threaded without redundancy. Spares should be kept or depot arrangements should be made with vendors to ensure that failed components can be replaced quickly, minimizing the impact on the business.

As the Internet becomes more and more of an integral part of a company's operations (as defined by impact on revenue or other valid measurement), storing of spare components, including hard drives, redundant controller cards, spare tape drives, and power supplies should be considered.

In summary, to ensure a system up to par with revenue-generating applications, companies should upgrade to series 5000 or 6000 routers (or equivalent), combined with dual connections to the ISP and "Round Robin DNS" at the same juncture. Depot arrangements should be established for spare parts, and services such as Southwestern Bell SecureNet and other methods for diversifying T1 access should be considered. Such precautions will provide cheap insurance for what will fast become a revenue-producing system.

RECOMMENDED COURSE OF ACTION

Although revenue-generating dial-in systems may seem to be far away for many companies, experience shows that systems like these have a way of catching on. Insurance companies love the idea of roving claims adjusters with dial-in laptop computers. Everyone wants to work at home.

Commerce is blossoming on the Internet. Waiting until there is a revenue impact after a failure resigns an organization to be almost perpetually in the reactive mode of trying to keep up with the protection of a potentially business debilitating system. The alternative is to start now, while these systems are still relatively immature and design the protective systems in before the Internet becomes a fully functional business system.

Leo A. Wrobel is president and CEO of Premiere Network Services, Inc., in DeSoto, TX. An active author, national and international lecturer, and technical futurist, he has published 10 books and over 100 trade articles on a variety of technical subjects, including *Writing Disaster Recovery Plans for Telecommunications and LANS* (Artech House, 1993) and *Business Resumption Planning* (Auerbach Publications, 1997). His experience of nearly two decades includes assignments at AT&T, a major mortgage banking company, and a host of other firms engaged in banking, brokerage, heavy manufacturing, telecommunications services and government, as well as the design and regulatory approval of a LATA-wide OC-12/ATM network for a $10 billion manufacturing giant, the first of its kind. A three-term city councilman and previous mayor, Leo Wrobel is a knowledgeable and effective communicator known for his entertaining presentation style on a wide variety of technical topics. For more information, contact his web site at http://www.dallas.net/~premiere or phone at (972) 228-8881.

E-mail Security Using Pretty Good Privacy

William Stallings

Payoff

Many users are unaware that their E-mail messages are completely public and can be monitored by someone else. This article describes Pretty Good Privacy, an E-mail security package that allows users to send messages that are secure from eavesdropping and guaranteed to be authentic.

Introduction

Users who rely on electronic mail for business or personal communications should beware. Messages sent over a network are subject to eavesdropping.If the messages are stored in a file, they are subject to perusal months or even years later. There is also the threat of impersonation and that a message may not be from the party it claims to be from. Protection is available in the form of Pretty Good Privacy (PGP), an E-mail security package developed by Phil Zimmermann that combines confidentiality and digital signature capabilities to provide a powerful, virtually unbreakable, and easy-to-use package.

PGP Defined

The most notable features of this E-mail security program are that it:

- Enables people to send E-mail messages that are secure from eavesdropping. Only the intended recipient can read a Pretty Good Privacy message.
- Enables people to send E-mail messages that are guaranteed authentic. The recipient is ensured that the PGP message was created by the person who claims to have created it and that no one has altered the message since it was created.
- Is available as freeware on the Internet, many electronic bulletin boards, and most commercial services such as CompuServe.
- Is available in versions for Disk Operating System, Macintosh, UNIX, Amiga, OS/2 , VMS, and other operating systems.
- Works with any E-mail package to create secure E-mail messages.

E-Mail Risks

PGP provides protection from the threat of eavesdropping. A message sent over the Internet can pass through a handful of mail forwarders and dozens of packet-switching nodes. A systems administrator or someone who has gained privileged access to any of these transfer points is in a position to read those messages.

Although E-mail users may feel they have nothing to hide, they may someday want to correspond with their lawyers or accountants using the Internet, or they may work for companies that want to send proprietary information over the Internet. Many people already use the Internet for sending highly personal or sensitive messages.

There is also a civil liberties issue to be concerned about. The police, intelligence, and other security forces of the government can easily monitor digital and computerized E-

mail messages, looking for key words, names, and patterns of exchanges. Any user could be innocently caught up in such a net.

Authenticity of messages poses another potential risk. It is not difficult to spoof the network into sending a message with an incorrect return address, enabling impersonation. It is also relatively easy to trap a message along its path, alter the contents, and then send it on its way.

For example, if a user is on a shared system, such as a UNIX system, that hooks into the Internet, then the impersonator could be someone with “superuser” privileges on the system. Such a person could divert all incoming and outgoing traffic from an unsuspecting mailbox to a special file. The impersonator could also have access to a router, mail bridge, or other type of gateway through which all traffic between the user and a correspondent must pass. Such impersonators could use their privileged status on the gateway to intercept mail and to create and send mail with a fraudulent return address.

PGP's History: Privacy At Issue

PGP is a legitimate tool that can be used for legitimate reasons by ordinary citizens, although some users consider it slightly suspect.

Phil Zimmerman began working on Pretty Good Privacy in the 1980s and released the first version in 1991. One of the key motivating factors for PGP's development was an effort by the FBI to secure passage of a law that would ban certain forms of security algorithms and force computer manufacturers to implement security features for E-mail that could be bypassed by government agencies. Zimmerman saw this as a threat to privacy and freedom. Thus, PGP was conceived as a package that could be used by the average person on a small system to provide E-mail privacy and authenticity. Zimmerman accomplished this by:

- Selecting the best available security algorithms as building blocks.
- Integrating these algorithms into a general-purpose application that is independent of the operating system and processor and that is based on a small set of easy-to-use commands.
- Making the package and its documentation, including the source code, free and widely available.

Because PGP uses encryption algorithm, it was subject to export controls. An encryption algorithms lets users scramble a message in such a way that allows only the intended recipient to unscramble it.

Encryption algorithms are classified by the US government as armaments and fall under the International Trafficking in Armaments Regulations (ITAR). ITAR requires that users get an export license from the State Department to export armaments. In practice, the State Department will not grant any such license for strong encryption algorithms, and PGP uses two of the strongest.

This problem does not need to concern the average user because there is no law against using PGP in the US. There is also no law outside the US to prevent use of a product that was illegally exported from the US. Furthermore, some of the more recent versions of PGP actually originated outside the US, eliminating the problem altogether.

A second problem has to do with patents. One of the two encryption algorithms in PGP is known as Rivest-Shamir-Adleman (RSA). Anyone using PGP inside the US was, for a time, potentially subject to a lawsuit for Rivest_Shamir-Adleman patent infringement.

A new release of PGP, known as version 2.6, which was developed at MIT with the supervision of Phil Zimmermann, has patent approval from the RSA patent holders. Like the original PGP, this version has also made its way onto bulletin boards and Internet sites outside the US. In addition, a compatible non-US version 2.6was created outside the US. As long as a user chooses any of the flavors of version 2.6, there is no infringement on any patents.

Conventional Encryption

PGP exploits two powerful security functions: conventional encryption and public-key encryption. Conventional encryption is the classic approach to secret codes that dates back to ancient Rome and even earlier. A conventional encryption scheme (see Exhibit 1) includes the following five ingredients:

- **Plaintext.** This is the readable message or data that is fed into the algorithm as input.
- **Encryption algorithm.** The encryption algorithm performs various substitutions and transformations on the plaintext.
- **Secret key.** The secret key is also input to the algorithm. The exact substitutions and transformations performed by the algorithm depend on the key.
- **Ciphertext.** This is the scrambled message produced as output. It depends on the plaintext and the secret key.
- **Decryption algorithm.** This is essentially the encryption algorithms run in reverse. It takes the ciphertext and the same secret key and produces the original plaintext.

Conventional Encryption

The Caesar cipher, used by Julius Caesar, is a simple example of encryption. The Caesar cipher replaces each letter of the alphabet with the letter standing three places further down the alphabet, for example:

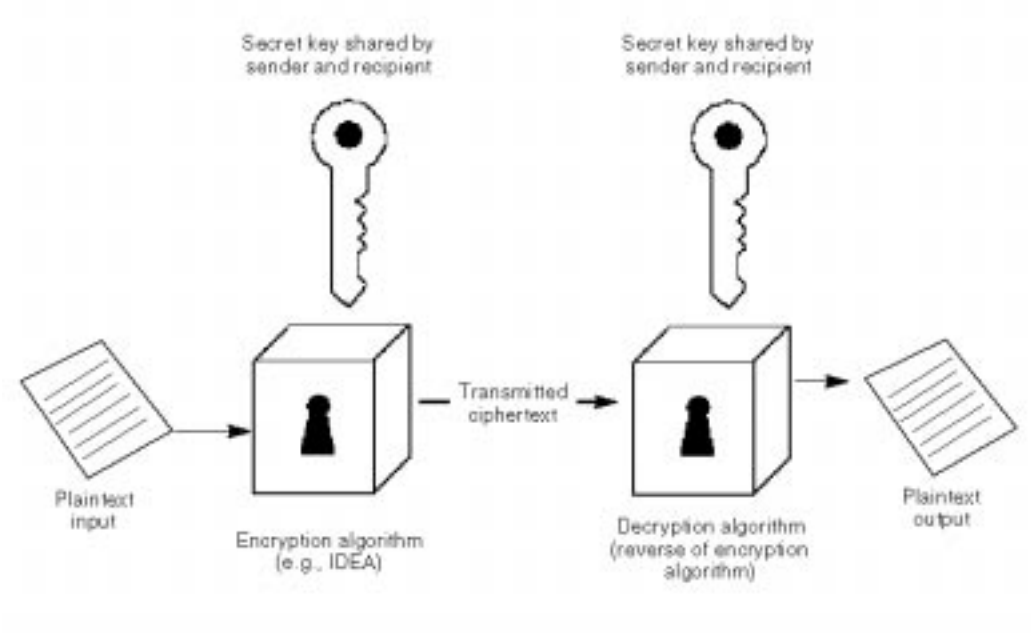
```
plain:  meet me after the toga party
cipher: phhw ph diwhu wkh wrjd sduwb
```

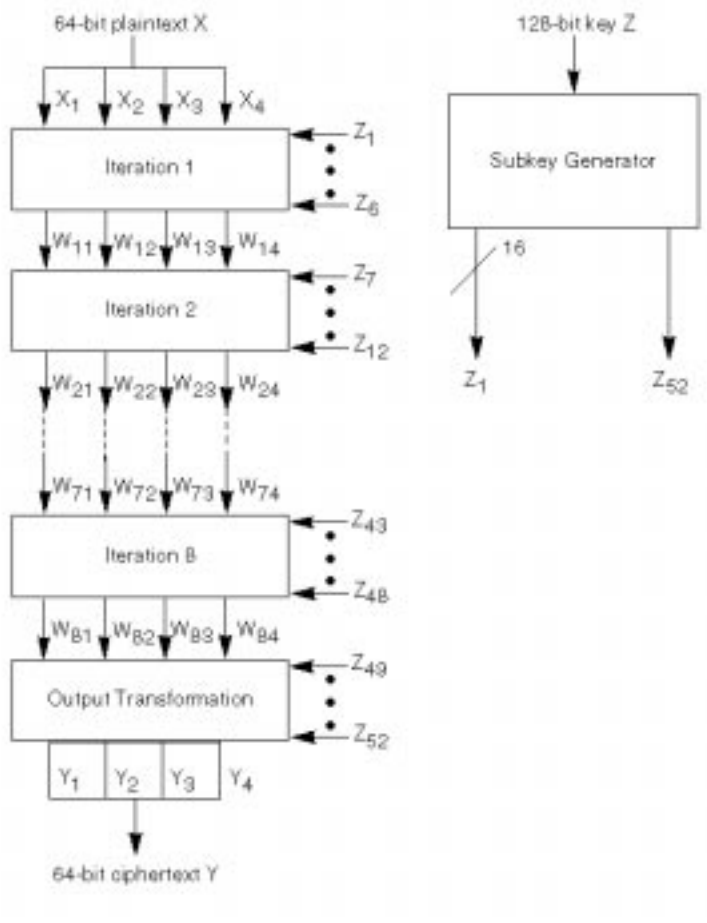
The alphabet is wrapped around so that the letter following Z is A. The decryption algorithm simply takes the ciphertext and replaces each letter with the letter standing three places earlier on in the alphabet. A general Caesar cipher involves a shift of *k* letters, where *k* ranges from 1 through 25. In this case, *k* is the secret key to the algorithm.

The Caesar cipher is not very secure. Anyone who wanted to decipher the code could simply try every possible shift from 1 to 25. Pretty Good Privacy uses a much stronger algorithm known as the International Data Encryption Algorithm, or Interactive Data Extraction and Analysis.

The International Data Encryption Algorithm

IDEA is a block-oriented conventional encryption algorithms developed in 1990 by Xuejia Lai and James Massey of the Swiss Federal Institute of Technology. The overall scheme for IDEA encryption is illustrated in Exhibit 2. IDEA uses a 128-bit key to encrypt data in blocks of 64 bits.





Overall IDEA Structure

The IDEA algorithm consists of eight rounds, or iterations, followed by a final transformation function. The algorithm breaks the input into four 16-bit subblocks. Each of the iteration rounds takes four 16-bit subblocks as input and produces four 16-bit output blocks. The final transformation also produces four 16-bit blocks, which are concatenated to form the 64-bit ciphertext. Each of the iterations also uses six 16-bit subkeys, whereas the final transformation uses four subkeys, for a total of 52 subkeys. The right-hand portion of the exhibit indicates that these 52 subkeys are all generated from the original 128-bit key.

Each iteration of IDEA uses three different mathematical operations. Each operation is performed on two 16-bit inputs to produce a single 16-bit output. The operations are:

- Bit-by-bit exclusive-OR, denoted as ∞.
- Addition of integers modulo 2^{16} (modulo 65536), with input and output treated as unsigned 16-bit integers. This operation is denoted as §
- Multiplication of integers modulo 216 + 1 (modulo 65537), with input and output treated as unsigned 16-bit integers, except that a block of all zeros is treated as representing 2^{16}. This operation is denoted as [Theta].

For example,

0000000000000000 [Theta]1000000000000000 = 1000000000000001

because

$$2^{16} * 2^{15} \bmod (2^{16} + 1) = 2^{15} + 1$$

These three operations are incompatible because no pair of the three operations satisfies a distributive law. For example:

a§b[Theta]c≠(a§b)[Theta](a§c)

They are also incompatible because no pair of the three operations satisfies an associative law. For example:

a§(b∞c)≠a§b)∞c

The use of these three separate operations in combination provides for a complex transformation of the input, making cryptanalysis very difficult.

Exhibit 3 illustrates the algorithm for a single iteration. In fact, this exhibit shows the first iteration. Subsequent iterations have the same structure, but with different subkey and plaintext-derived input. The iteration begins with a transformation that combines the four input subblocks with four subkeys, using the addition and multiplication operations. This transformation is highlighted as the upper shaded rectangle. The four output blocks of this transformation are then combined using the XOR operation to form two 16-bit blocks that are input to the lower shaded rectangle, which also takes two subkeys as input and combines these inputs to produce two 16-bit outputs.

Single Iteration of IDEA (First Iteration)

Finally, the four output blocks from the upper transformation are combined with the two output blocks of the MA structure using XOR to produce the four output blocks for this iteration. The two outputs that are partially generated by the second and third inputs (X_2 and X_3) are interchanged to produce the second and third outputs (W_{12} and W_{13}), thus increasing the mixing of the bits being processed and making the algorithm more resistant to cryptanalysis.

The ninth stage of the algorithm, labeled the output transformation stage in Exhibit 2, has the same structure as the upper shaded portion of the preceding iterations (see Exhibit 3). The only difference is that the second and third inputs are interchanged before being applied to the operational units. This has the effect of undoing the interchange at the end of the eighth iteration. This extra interchange is done so that decryption has the same structure as encryption. This ninth stage requires only four subkey inputs, compared to six subkey inputs for each of the first eight stages. The subkeys for each iteration are generated by a series of shifts on the original 128-bit key.

IDEA has advantages over older conventional encryption techniques. The key length of 128 bits makes it resistant to brute-force key search attacks. IDEA is also highly resistant to cryptanalysis and was designed to facilitate both software and hardware implementations.

Public-Key Encryption

One essential characteristic of Interactive Data Extraction and Analysis and all conventional encryption algorithm is the need for the two parties to share a secret key that is not known to anyone else. This is a tremendous limitation, especially for an E-mail application.

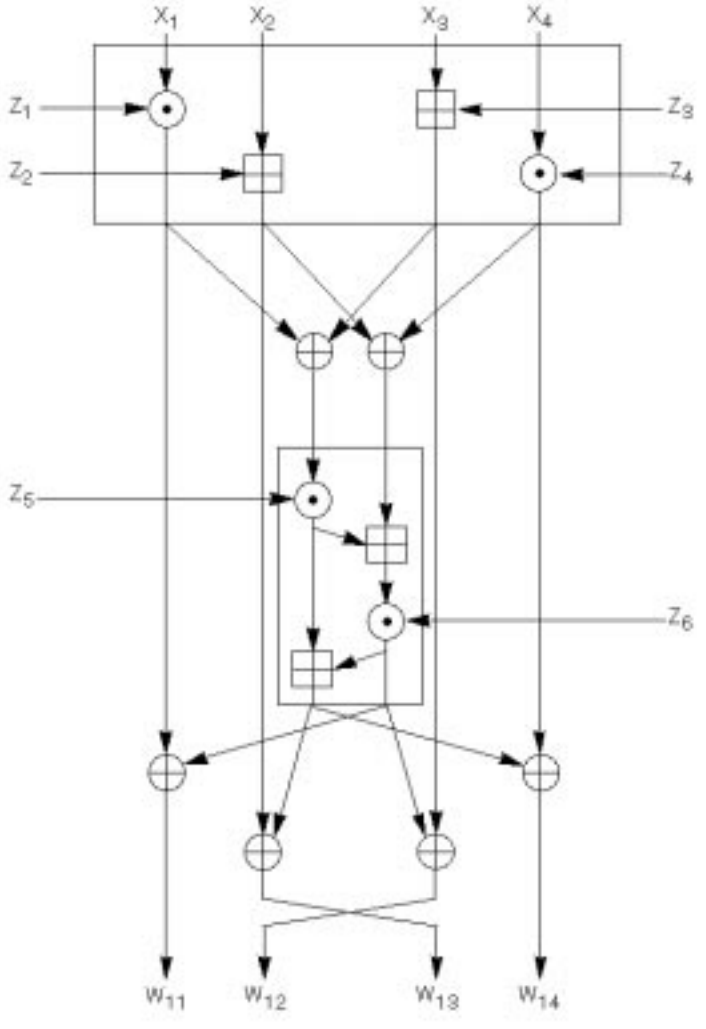
If Pretty Good Privacy depended solely on the use of IDEA, before a user could correspond with anyone, that user would somehow have to arrange to share a secret 128-bit number with the message recipient. If there is no way to communicate securely, it becomes difficult to send the key.

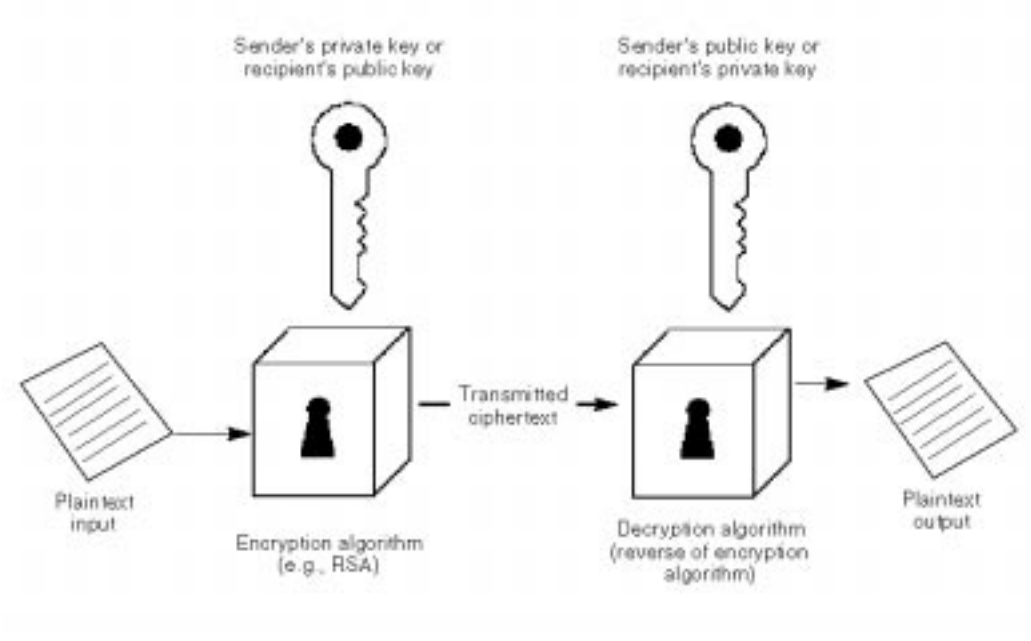
A new approach to encryption known as public-key encryption offers a solution to this problem. With this method, developed in 1976 by Whitfield Diffie, there is no need to convey a secret key. Instead, each person has a private key and a matching public key. Encryption is done with one of these two keys and decryption uses the other. The private key is kept secret, known only to its holder. The matching public key is just that—public. The private key holder can broadcast the matching public key.

Public-key encryption can be used to ensure privacy in much the same way as IDEA (see Exhibit 4). Users put plaintext and the intended recipient's public key in the encryption algorithms. The algorithm uses the plaintext and the public key to produce ciphertext. At the receiving end, the decryption algorithm, which is the reverse of the encryption algorithms, is used. In this case, the input is the ciphertext and the receiver's private key. This message is secure from eavesdropping because only the receiver has the private key necessary for decryption. Anyone who has a copy of the recipient's public key can create a message that can be read only by this recipient.

Public-Key Encryption

Authentication can also be performed by putting plaintext and the sender's private key in the encryption algorithms. The algorithm uses the plaintext and the private key to produce ciphertext. At the receiving end, the decryption algorithm, which is the reverse of the





encryption algorithms, is used. In this case, the input is the ciphertext and the sender's public key.

This message is guaranteed to be authentic because only the sender has the private key necessary for encryption. Anyone who has a copy of the sender's public key can read the message and verify that it must have come from the alleged sender.

The public-key scheme used for PGP is the Rivest_Shamir-Adleman algorithm. RSA takes variable-length keys. Typically, the key size for both the private and public keys is 512 bits.

The RSA Algorithm

One of the first public-key schemes was developed in 1977 by Ron Rivest, Adi Shamir, and Len Adleman at MIT and first published in 1978. Named for its creators, the Rivest-Shamir-Adleman (RSA) scheme has since reigned as the only widely accepted and implemented approach to public-key encryption. RSA is a block cipher in which the plaintext and ciphertext are integers between 0 and n - 1 for some n. Encryption and decryption take the following form for some plaintext block M and ciphertext block C:

$C = M^e \bmod n$

$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$

Both sender and receiver must know the value of n. The sender knows the value of e, and only the receiver knows the value of d. Thus, this is a public-key encryption algorithms with a public key of KU = {e, n} and a private key of KR = {d, n}. For this algorithm to be satisfactory for public-key encryption, the following requirements must be met:

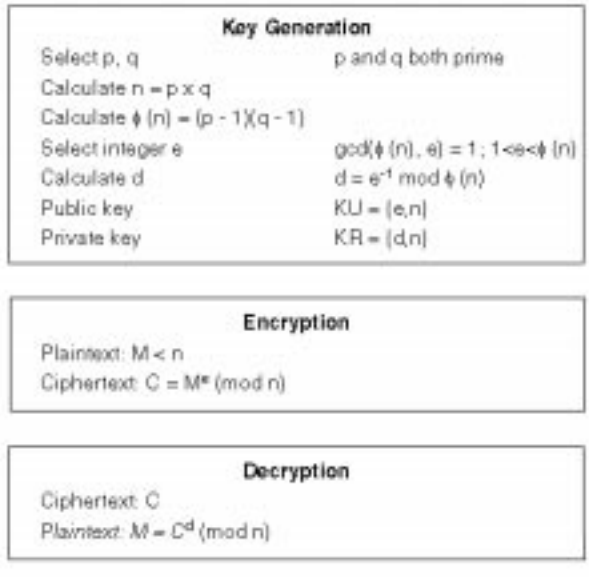
- It should be possible to find values of e, d, n such that $M^{ed} = M \bmod n$ for all $M < n$.
- It should be relatively easy to calculate M^e and C^d for all values of $M < n$.
- It should be infeasible to determine d given e and n.

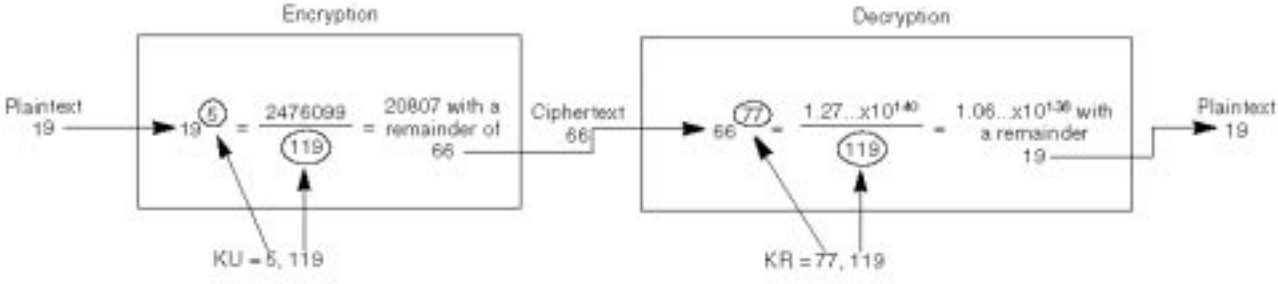
Exhibit 5 summarizes the RSA algorithm. To understand the algorithm, users should begin by selecting two prime numbers, p and q, and calculating their product n, which is the modulus for encryption and decryption. Next, the quantity $\phi(n)$, which is referred to as the Euler totient of n, which is the number of positive integers less than n and relatively prime to n should be determined. Then an integer d, that is relatively prime to f (n), (i.e., the greatest common divisor of d and $\phi(n)$ is 1), should be selected. Finally, $e$should be calculated as the multiplicative inverse of d, modulo $\phi(n)$. It can be shown that d and e have the desired properties.

The private key consists of {d, n} and the public key consists of {e, n}. Suppose that user A has published its public key and that user B wishes to send the message M to A. Then, B calculates $C = M^e \pmod n$ and transmits C. On receipt of this ciphertext, user A decrypts by calculating $M = C^d \pmod n$.

An example is shown in Exhibit 6. For this example, the keys are generated as follows:

- Two prime numbers, p = 7 and q = 17, are selected.
- Calculate $n = pq = 7 \times 17 = 119$.





Key:
KU public key
KR private key

- Calculate f (n) = (p-1)(q-1) = 96.
- Select e such that e is relatively prime to f (n) = 96 and less than f (n);in this case, e = 5.
- Determine d such that de =1 mod 96 and d < 96. The correct value is d = 77, because 77 x 5 = 385 = 4 x 96 + 1.

The resulting keys are public key KU = {5, 119} and private key KR = {77,119}. The example shows the use of these keys for a plaintext input of M = 19.For encryption, 19 is raised to the fifth power, yielding 2,476,099. Upon division by 119, the remainder is determined to be 66. Therefore, 195 [equiv]66 mod 119, and the ciphertext is 66. For decryption, it is determined that 66^{77} [equiv]19 mod 119.

How Hard Is It to Break the Code?

There are two possible approaches to defeating the RSA algorithm. The first is the brute-force approach: trying all possible private keys. Thus the larger the number of bits in $e$and d, the more secure the algorithm. However, because the calculations involved, both in key generation and in encryption/decryption, are complex, the larger the size of the key, the slower the system will run.

Most discussions of the cryptanalysis of RSA have focused on the task of factoring p into its two prime factors. Until recently, this was considered infeasible for numbers in the range of 100 decimal digits, which is about 300 or more bits. To demonstrate the strength of Rivest-Shamir-Adleman, its three developers, issued a challenge to decrypt a message that was encrypted using a 129-decimal-digit number as their public modulus. The authors predicted that it would take 40 quadrillion years with current technology to crack the code. Recently, the code was cracked by a worldwide team cooperating over the Internet and using more than 1,600 computers after only eight months of work. This result does not invalidate the use of RSA; it simply means that larger key sizes must be used. Currently, a 1,024-bit key size (about 300 decimal digits), is considered strong enough for virtually all applications.

How PGP Works

Digital Signature

It may seem that Rivest-Shamir-Adleman is all that is needed for a secure E-mail facility. Everyone who wants to use Pretty Good Privacy can create a matching pair of keys (PGP will do the necessary calculation) and then distribute the public key. To send a message, it must first be encrypted with the private key to guarantee its authenticity. Next, the result of step one must be encrypted with the recipient's public key to guarantee that no one else can read the message.

This scheme is technically valid but impractical. The problem is that RSA, and all other public-key schemes, are very slow. To double-encrypt messages of arbitrary length is far too time-consuming. Users could experience delays of minutes or even hours waiting for their PCs to do the number-crunching.

Instead, PGP exploits the strengths of conventional and public-key encryption. When a message is sent, it goes through two security-related stages of processing: digital signature and encryption.

The digital signature is one of the most clever innovations to come out of the work on public-key encryption. To use digital signature, users take the message that they want to send and map it into a fixed-length code of 128 bits. The algorithm for doing this is called MD5 (message digest version 5). The 128-bit message digest is unique for this message. It would be virtually impossible for someone to alter this message or substitute another message and still come up with the same digest.

PGP then encrypts the digest using RSA and the sender's private key. The result is the digital signature, which is attached to the message. Anyone who gets this message can re-compute the message digest and then decrypt the signature using RSA and the sender's public key. If the message digest in the signature matches the message digest that was calculated, then the signature is valid. Because this operation only involves encrypting and decrypting a 128-bit block, it takes little time.

For the encryption stage, PGP randomly generates a 128-bit secret key and uses Interactive Data Extraction and Analysis to encrypt the message plus the attached signature. The recipient can discover the secret key by using RSA. PGP takes the secret key as input to RSA, using the receiver's public key, and produces an encrypted secret key that is attached to the message. On the receiving end, PGP uses the receiver's private key to recover the secret key and then uses the secret key and IDEA to recover the plaintext message plus signature.

Getting Public Keys

Public-key encryption techniques make use of two keys for each user: a private key that is known only to one user, and a corresponding public key that is made known to all users. With these two keys, it is possible to create digital signatures that guarantee the authenticity of a message and to support the encryption of a message in such a way that only the intended recipient can read it.

There is, however, a common misconception that each user simply keeps his or her private key private and publishes the corresponding public key. Unfortunately, this is not a simple solution. An impostor can generate a public- and private-key pair and disseminate the public key as if it were someone else's. For example, suppose that user A wishes to send a secure message to user B. Meanwhile, user C has generated a public- and private-key pair, attached user B's name and an E-mail address that user C can access, and published this key widely. User A has picked this key up, uses the key to prepare her message for user B, and uses the attached E-mail address to send the message. Result: user C receives and can decrypt the message; user B either never receives the message or cannot read it without holding the required private key.

One way around this problem is to insist on the secure exchange of public keys. For example, if user B and user A know each other personally and live near each other, they could physically exchange keys on diskettes. But for PGP to be useful as a general-purpose E-mail security utility, it must be possible for people in widely distributed sites to exchange keys with others that they have never met and may not even know.

Public-Key Certificates and Distributed Security

The basic tool that permits widespread use of PGP is the public-key certificate. The essential elements of a public-key certificate are:

- The public key itself.

- A user ID consisting of the name and E-mail address of the owner of the key.

- One or more digital signatures for the public key and user ID.

The signer testifies that the user ID associated with this public key is valid. The digital signature is formed using the private key of the signer. Anyone in possession of the corresponding public key can verify that the signature is valid. If any change is made, either to the public key or the user ID, the signature will no longer compute as valid.
Public-key certificates are used in several security applications that require public-key cryptography. In fact, it is the public-key certificate that makes distributed security applications using public keys practical.

One approach that might be taken to use public-key certificates is to create a central certifying authority. This is the approach recommended for use with the privacy-enhanced mail (PEM) scheme. Each user must register with the central authority and engage in a secure exchange that includes independent techniques for verifying user identity. Once the central authority is convinced of the identity of a key holder, it signs that key. If everyone who uses this scheme trusts the central authority, then a key signed by the authority is automatically accepted as valid.

There is nothing inherent in the PGP formats or protocols to prevent the use of a centralized certifying authority. However, PGP is intended as an E-mail security scheme for the masses. It can be used in a variety of informal and formal environments. Accordingly, Pretty Good Privacy is designed to support a so-called web of trust, in which individuals sign each other's keys and create an interconnected community of public-key users.

If user B has physically passed a public key to user A, then user A knows that this key belongs to user B and signs it. User A keeps a copy of the signed key and also returns a copy to user B. Later, user B wishes to communicate with user D and sends this person the public key, with user A's signature attached. User D is in possession of user A's public key and also trusts user A to certify the keys of others. User D verifies user A's signature on user B's key and accepts user B's key as valid.

Computing Trust

Although Pretty Good Privacy does not include any specification for establishing certifying authorities or for establishing trust, it does provide a convenient means of using trust, associating trust with public keys, and exploiting trust information.

Each user can collect a number of signed keys and store them in a PGP file known as a public-key ring. Associated with each entry is a key legitimacy field that indicates the extent to which PGP will trust that this is a valid public key for this user; the higher the level of trust, the stronger is the binding of this user ID to this key. This field is computed by Pretty Good Privacy. Also associated with the entry are zero or more signatures that the key ring owner has collected that sign this certificate. In turn, each signature has associated with it a signature trust field that indicates the degree to which this PGP user trusts the signer to certify public keys. The key legitimacy field is derived from the collection of signature trust fields in the entry. Finally, each entry defines a public key associated with a particular owner, and an owner trust field is included that indicates the degree to which this public key is trusted to sign other public-key certificates; this level of trust is assigned by the user. The signature trust fields can be thought of as cached copies of the owner trust field from another entry.

Trust Processing

If user A inserts a new public key on the public-key ring, PGP must assign a value to the trust flag that is associated with the owner of this public key. If the owner is in fact A, and this public key also appears in the private-key ring, then a value of ultimate trust is automatically assigned to the trust field. Otherwise, PGP asks user A for an assessment of the trust to be assigned to the owner of this key, and user A must enter the desired level. The user can specify that this owner is unknown, untrusted, marginally trusted, or completely trusted.

When the new public key is entered, one or more signatures may be attached to it. More signatures may be added later on. When a signature is inserted into the entry, PGP searches the public-key ring to see if the author of this signature is among the known public-key owners. If so, the OWNERTRUST value for this owner is assigned to the SIGTRUST field for this signature. If not, an unknown user value is assigned.

The value of the key legitimacy field is calculated on the basis of the signature trust fields present in this entry. If at least one signature has a signature trust value of ultimate, then the key legitimacy value is set to complete. Otherwise, PGP computes a weighted sum of the trust values. A weight of 1/X is given to signatures that are always trusted and 1/Y to signatures that are usually trusted, where* and Y are user-configurable parameters. When the total of weights of the introducers of a key/user ID combination reaches 1, the binding is considered to be trustworthy, and the key legitimacy value is set to complete. Thus, in the absence of ultimate trust, at least X signatures that are always trusted or Y signatures that are usually trusted or some combination, is needed.

Signature Trust and Key Legitimacy

Periodically, PGP processes the public-key ring to achieve consistency. In essence, this is a top-down process. For each OWNERTRUST field, PGP scans the ring for all signatures authored by that owner and updates the SIGTRUST field to equal the OWNERTRUST field. This process starts with keys for which there is ultimate trust. Then, all KEYLEGIT fields are computed on the basis of the attached signatures.

Exhibit 7 provides an example of the way in which signature trust and key legitimacy are related. The exhibit shows the structure of a public-key ring. The user has acquired a number of public keys, some directly from their owners and some from a third party such as a key server.

PGP Trust Model Example

The node labeled “You” refers to the entry in the public-key ring corresponding to this user. This key is valid and the OWNERTRUST value is ultimate trust. Each other node in the key ring has an OWNERTRUST value of undefined unless some other value is assigned by the user. In this example, the user has specified that it always trusts users D, E, F, and L to sign other keys. This user also partially trusts users A and B to sign other keys.

The shading, or lack thereof, of the nodes in Exhibit 7 indicates the level of trust assigned by this user. The tree structure indicates which keys have been signed by which other users. If a key is signed by a user whose key is also in this key ring, the arrow joins the signed key to the signer. If the key is signed by a user whose key is not present in this key ring, the arrow joins the signed key to a question mark, indicating that the signer is unknown to the user.

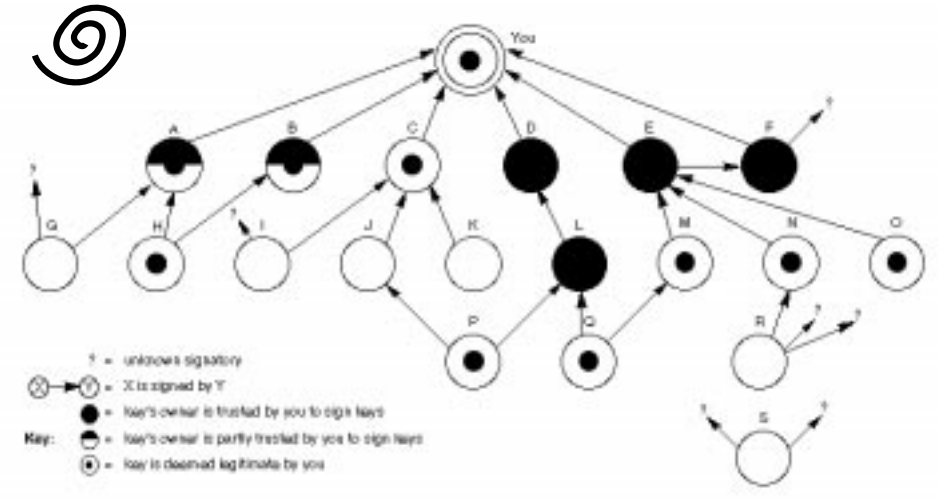


Exhibit 7 illustrates that all keys whose owners are fully or partially trusted by the user have been signed by this user, with the exception of node L. Such a user signature is not always necessary, as the presence of node L indicates, but in practice most users are likely to sign the keys for most owners that they trust. So, for example, even though E's key is already signed by trusted introducer F, the user chose to sign E's key directly.
It can be assumed that two partially trusted signatures are sufficient to certify a key. Hence, the key for user H is deemed valid by PGP because it is signed by A and B, both of whom are partially trusted.

A key may be determined to be valid because it is signed by one fully trusted or two partially trusted signers, but its user may not be trusted to sign other keys. For example, N's key is valid because it is signed by E, whom this user trusts, but N is not trusted to sign other keys because this user has not assigned N that trust value. Therefore, although R's key is signed by N, PGP does not consider R's key valid. This situation makes perfect sense. If a user wants to send a secret message to an individual, it is not necessary that the user trust that individual in any respect. It is only necessary to ensure use of the correct public key for that individual.

Exhibit 7 also shows a detached orphan node S, with two unknown signatures. Such a key may have been acquired from a key server. PGP cannot assume that this key is valid simply because it came from a reputable server. The user must declare the key valid by signing it or by telling PGP that it is willing to fully trust one of the key's signers.
It is the PGP web of trust that makes it practical as a universal E-mail security utility. Any group, however informal and however dispersed, can build up the web of trust needed for secure communications.

Conclusion

PGP is already widely used. Pretty Good Privacy has become essential to those struggling for freedom in former Communist countries. Ordinary people throughout the world are active participants in the alt.security.PGP USENET newsgroup. Because PGP fills a widespread need, and because there is no reasonable alternative, its future is secure. One of the best lists of locations for obtaining PGP, with the file name getpgp.asc, is maintained at two File Transfer Protocol sites on the Internet: ftp.csn.net/mpj and ftp.netcom.com/pub/mp/mpj.

Author Biographies

William Stallings
William Stallings is an independent consultant and president of Comp-Comm Consulting of Brewster MA. He is the author of 14 books on data communications and computer networking. This article is based on material in the author's latest book, *Protect Your Privacy: A Guide for PGP Users*(Englewood Cliffs NJ: Prentice-Hall, 1995).

PROTECTING AGAINST DIAL-IN HAZARDS: VOICE SYSTEMS

Leo A. Wrobel

INSIDE

Voice-Mail Breaches and Preventive Measures, Facsimile Breaches and Preventive Measures, Cellular Phone Security and Preventive Measures, Cracking Voice Systems, Precautions for Telephone Fraud

INTRODUCTION

"Hello, this is Carl. I can't take your call right now, but if you leave your name and number, I'll return your call as soon as I can. If your call is urgent, press zero for the operator."

Sounds familiar enough, doesn't it? Just suppose, this time, it is not a customer or employee, but rather a hacker attempting to compromise your system. This is not difficult to do. In fact, it is as easy in many cases as dialing one number: zero. In this case, the hacker exercises the "press-zero) option, and when the operator answers, he proceeds to jump down her throat:

Hacker: "Where the HELL is Carl! I've been trying him all day and he's supposed to be at this desk! That lowlife is in big trouble! Oh, never mind, just give me an outside line so I can track him down!"

The hacker's objective is to intimidate and frazzle the receptionist, so the abusive language is common. The receptionist has one more hostile caller, who appears to be calling from inside the building, after all, who's number is lighting up on her switchboard? Why Carl's, of course (remember, the hacker transferred from Carl's line). So what happens next? More often than not, the receptionist will give the caller an outside line. Once connected to the local Bell operator, and the receptionist has disconnected and moved on to the next caller, the hacker continues:

PAYOFF IDEA

Dial-in voice technologies, including voice mail, facsimile machines, and cellular phones, provide fertile ground for fraud that is easily accomplished but difficult to recover. The targeted suggestions presented for each of these technologies provide practical and cost-effective solutions that can save organizations tens of thousands of dollars and a career or two along the way.

Hacker: "Operator, I would like to place a call to Bombay, India, please."

Operator: "How would you like to pay for that call?"

Hacker: "Oh, just bill it to this number."

This example shows one of the easiest ways a hacker can compromise a network. Voice mail systems are common targets, but compared to some of the more sophisticated tricks, this scenario is the technological equivalent of stone knives and bear claws. Thousands of companies fall victim each year, and when it happens to them, they get stuck with the bill. One feature that can help protect against these operator "splash-backs" is called line screening. It is generally used to tip the operator off that the call is coming from a hotel or pay phone. It is typically used on hotel and pay phone trunks to avoid the situation just described. However, most businesses have never heard of it, let alone ordered it. Absent this precaution, procedures prohibiting any transfer (except 911) and mandatory operator training are a must.

EXAMPLES OF VOICE MAIL SECURITY BREACHES

An astute hacker can tell from listening to the automated voice on a company's voice mail the type of system the company uses. This can be disastrous, because hackers will also usually know what the factory default codes are for the system, which will allow them to dial into the operating system. This could mean they could make outgoing international calls. It also makes for some other interesting situations, as the following example indicates.

One user I met at a seminar said his voice mail system was used for several months to run a call girl operation. The hackers were able to break in and set up special voice mailboxes (using the authorized users boxes) after hours. They were always careful to delete all of these messages before the employees came in at 8 a.m. One day, however, the call girl operation was either raided, moved, or just lost interest in the voice mail system. All the messages to the "clients" were still there when the shifts came in on Monday morning! The content of these messages was not elaborated on by the telecom manager, but I am told many of the surprising messages found that morning are still legendary within the company. This is comical, especially because there was no real cost to the company other than to raise the office gossip a notch or two.

What was unsettling for everyone involved, however, was that it went undetected for so long. Does your company change these codes frequently? Does it monitor for unsuccessful attempts? Would it know if it were being hacked right now?

A second company was not so lucky. This company had a formal policy to block all operator transfers from voice mail to outside lines, except for 911. The operators were trained and signed off on the new proce-

dure. One Thanksgiving weekend, when the operators were off and the phones rolled to the security guard (who was not trained), a friendly caller identified himself as "Bill at AT&T" asked for an outside transfer to test the phones. "Bill" spent all Thanksgiving Day and the next three days, hacking the company. Each time, the guard made an entry in the log "Assissted AT&T." On Monday morning, a security supervisor saw dozens of entries, got suspicious, and called the telecom manager. The result? A five-digit telephone bill to Pakistan! Would you know if this was happening right now in your organization? Has everyone been trained?

PREVENTATIVE MEASURES FOR VOICE MAIL

What can a user do to protect against abuse on these systems? The following activities provide a prevention checklist:

1. Implement policy that prohibits transfers to outside lines in all cases, except for the possible exception of a 911 emergency transfer.
2. Disconnect DISA (direct inward system access). Hackers refer to it as Dial In/Steal Away. It is used by nomadic or homebound workers to access PBX dial tone from a remote location. The caller dials in, gets a dial tone back, enters an access code, then completes a call on the company switch. These systems literally cry out to the world, "Please hack me!" If this system is necessary, use long, complicated access codes, and consider having the system answer with silence, not a dial tone, until after the caller enters the code.
3. Monitor systems for suspicious activity. If 300 unsuccessful attempts were made on DISA or dial-in modem ports last night, would the company know it?
4. Are the dial-in ports to multiplexers, routers, and PBX monitored for suspicious activity? Are the original factory codes changed? Besides vendors, who may be dialing in right now?
5. Watch for dumpster divers. Many people go through the company's trash looking for credit card receipts and long distance access codes!
6. Hire bonded maintenance workers. Do you know the cleaning crews? Are users instructed to not leave sensitive access codes out on desks where crews can find them? One client of mine actually had a high-end fax machine stolen from a 17th floor office! It would found in a nearby pawn shop a few days later.
7. Does the company have a telecommunications privacy policy? This should be implemented and be broad enough in scope to cover E-mail, voice mail, and other mediums.

For example:

"ABC Company is committed to absolute privacy of communications, and each employee has the right to not have their communications mon-

itored. However, if in the course of normal maintenance activity we inadvertently discover illegal activity, we reserve the right to report this activity to the responsible authorities."

This would give some recourse if a situation required monitoring. One manufacturing company's employees came in on a Monday morning after a holiday (hackers love long weekends) to find three T1's worth of traffic into U.S. Sprint, all filled with people speaking Spanish: 72 channels of people speaking a foreign tongue to a faraway land on the company's nickel. The company did not have any Spanish-speaking clients, employees, or overseas branches. Most of the people on the calls probably did not know it was illegal. Scam artists constantly work the immigrant communities, and lines of new immigrants at pay phones waiting for "discount calls" from disreputable thieves is a common sight in many cities.

FACSIMILE SECURITY BREACHES

What about other internal compromises of security, such as the fax machine? Because the security of information is paramount in business, and unauthorized access of information is becoming more prevalent, facsimile transmissions should not be neglected. The fax machine is widely accepts and heavily used, yet security precautions for these devices often do not exist.

Fax security concerns not only fax machines but fax boards in workstations. A company must know how much proprietary information might be leaving your organization this very instant from a $79 fax board. This has a direct bearing on LAN (local area network) standards, which every company must have. What is the company's liability if it receives a fax intended for another company, or even more ominously, if an employee or employees use the information learned from the fax for personal gain? It could be significant.

Even more disturbing, however, are people who are deliberately trying to obtain proprietary information from a company. For example, is the wastebasket next to the fax machine routinely shredded, or are these materials simply thrown away? Is the machine in a visible area? Is it checked frequently for incoming messages to avoid confidential correspondence from being in open view for extended periods of time? Is the fax machine used at any time for truly proprietary data? If so, additional precautions are necessary. All too often, the major cause of facsimile interception (or any type of confidential information) is from internal sources. Companies often ignore the threat from inquisitive or disgruntled employees. These employees may read fax traffic. The negative effect of unauthorized access to payroll, force reduction, or financial information through unsecured fax machines is self evident.

PREVENTIVE MEASURES FOR FAX SECURITY

To prevent unwanted access to fax machines, a company can take several proactive steps. Rather than name specific makes and models of equipment (which become quickly outdated), the specific features desirable for securing a fax machine, the operating environment surrounding the machine, or both, are described in the following sections.

Use a High-End Fax Machine

These machines receive fax transmissions into memory and store them on a hard drive in a confidential mail box. The addressing scheme should require the use of extended dialing by the sender to store the fax in the confidential mailbox. A user with an appropriate security code must then enter this code into the machine in order to retrieve the fax from the hard drive. The fax will then be printed and the file simultaneously deleted from the hard drive.

Use a Low-End Desktop Machine

For users who consistently send or receive confidential or sensitive information, use of a low-end desk top machine may be advisable. This can be controlled in the area where needed. As a point of need machine, it can be used by a specific department for sending confidential information. For routine administrative traffic, the normal centralized fax machine can be used. This separates the traffic, providing an additional level of security. It is also inexpensive because the cost of these desktop machines is often under $400, depending on features required, plus telephone line costs.

Store-and-Forward Fax and Fax Servers on LANs

Here, a computer sends from memory and receives into memory. The use of confidential long-on procedures to send and receive messages is similar to that of high-end fax machines. Local area networks now employ fax servers on the networks. These are actually computers that receive faxes from users and store them until transmission can take place. Original documents must be scanned into the system, which adds a level of complexity. Further, inbound traffic must be capable of being directed to a confidential mail box, similar to high-end stand-alone fax machines. If this is not implemented, security is violated because a systems administrator would have to read the fax, then direct it to the intended individual either on the network or in paper form.

Front-End with a Voice Mail System

Similar to some of the schemes just described, a voice mail system can be often modified to provide storage and password protection for incom-

ing faxes. The system would be programmed to answer with a message such as: "You have two messages and one fax." The user would then retrieve the voice messages from any phone but would be forced to dial from a fax machine to retrieve the fax. The alternative to moving to a fax machine is to enter a code and have the voice mail system dial out to a common fax machine; however, this is considered more risky. An employee could enter his code, then immediately receive another incoming call or have something else distract his attention. In the meantime, the fax comes in, unprotected, to an unattended machine. Built-in safeguards and procedures that force employees to be at the machine to retrieve a fax are preferable for this reason.

Because interception and monitoring is possible on almost any kind of phone line, lines serving especially sensitive fax machines can be made secure through the use of an encryption device. These devices, which can be expensive, scramble the data before it is sent. Some work for both fax machines and conversations. This means that both ends have comparable equipment for encryption/decryption of the information. Otherwise, fax traffic must be shipped in an unsecured mode.

Educating Users

Users with lap-top computers or fax cards in personal computers must be educated about the risks associated with the transmission and reception of faxes across these systems. Appropriate audit and security controls will help to avoid confidential files and information from being faxed out of the company directly from a PC without being saved on paper.

Hotel Faxes

The best approach is to avoid sending confidential information to or through a hotel fax machine. Hotel clerks usually make copies of the fax traffic in case there is an inquiry about the status. It also is not uncommon for hotel staff to deliver a fax to the wrong party.

CELLULAR PHONE SECURITY

Another frequently abused armament of the road warrior is the cellular phone. A multi-billion manufacturing company was arranging financing for the company. Each day, the CEO would discuss the most intimate details of the transaction on his cellular phone. Only as the deal neared a close did he question whether someone could monitor his conversations. Luckily, things went smoothly. Nonetheless, the CEO was shocked to learn that anyone with a $200 bear scanner could have monitored his entire conversation with ease.

CLONING CELL PHONES

Thieves also have become adept at cloning bogus authorization numbers into cellular phones. In the past, cellular providers have been hit hard by hackers cloning phones, then calling overseas. Now, most block calls of this type unless the provider is certain the user is authorized. However, this often can mean inconvenience for authorized users who happen to be roaming in another service area, where their identity cannot always be guaranteed with certainty.

This problem can also involve domestic calls. I was recently in an area of New York City (which has the dubious distinction of being the telephone fraud capital of the world) where I could not roam with my cell phone at all. All calls were directed to an operator, who would only complete a telephone credit card call. I found this preposterous because hundreds of hackers were probably standing by using their $200 bear scanners, just waiting for me to read my credit card number to the operator so they could steal it and beat it to death calling overseas locales. I instead chose to find a pay phone.

Precautions for Cellular Phones

The following list is designed to help secure cellular voice systems.

1. Never give a credit card or telephone credit card number over a cellular phone.
2. Never say anything over any wireless phone that you would not mind the whole world knowing.
3. Try to dial 800 numbers whenever possible when on the road rather than making credit card calls. An astute hacker can capture your touch tone digits when you make an automated credit card call, even if the number is not read to an operator.
4. Monitor your cellular bill closely and report any unusual calling activity (indicating your number may be been cloned) to your cellular provider immediately.
5. Consider upgrading to digital cellular service. Although not foolproof, digital technologies require more sophisticated equipment to monitor and are thus more difficult to intercept by hackers.

CRACKING VOICE SYSTEMS

It is easy for a potential hacker to download a "War Games" auto dialer from the Internet. The system need not be one that caters to hackers, because these programs are commonly available from lots of sites. These are designed specifically to find DISA and modem lines. The programs sequentially dial every number in a NXX (within a given three-digit pre-

fix) logging every number that answers with a carrier (modem) tone or dial tone. This is done while the hacker sleeps. For example, if the hacker knows your company has a 755 prefix for its telephone numbers, the program is set up to dial sequentially, 775-0000, 775-0001, 775-0002 … up to 775-9999. At some point they will hit, and log, every one of the numbers that answers with a dial tone or carrier. After a good night's rest, the hacker has a list of numbers to use.

Once the hacker has a list of possible candidates, he first screens them with a regular modem program to see what they are. Some are naturally metering circuits, credit card authorization lines, or other numbers only of marginal interest to the hacker.

Recommended Telephone Fraud Precautions

Organizations should take several precautions to prevent this type of abuse. These include:

- Disconnecting DISA lines if at all possible. This is the only guaranteed solution. If disconnection is not possible, consider the remaining suggestions.
- Using longer access codes. Do not use a three-digit access code; rather, use a 7, 8, or 9 digit code. The odds increase exponentially for each number you add to the code. If your system is too difficult to crack, the hacker is likely to move on to an easier mark.
- Optioning a DISA to answer with silence. The dial tone that most DISA lines put out when answering is a dead give away and an open invitation to hackers and the curious. By answering the line with silence, the automated equipment used by the hacker will see the line as a voice call, and will not flag it as useful.
- Monitoring traffic. Several warning signs of impending disaster include:
  1. Increased traffic volume on 1-800 lines
  2. Increased traffic volume on outgoing trunks
  3. Increased access to 950

  By monitoring this traffic, it may be possible to identify fraud before it is too late. Similarly, carriers can aid in this pursuit. Most now offer fraud insurance at a monthly fee, which provides traffic monitoring and reporting of any unusual circumstances or traffic loads.
- Blocking 011, 1-900, 976 access. If the company does not have Caribbean offices, disable the 809 area code, along with all the recently assigned, new Caribbean area codes. These are a high source of fraudulent calls and are a source of anguish and exposure for many a telecom manager.

Remember, if a company is defrauded, the FCC has ruled that carriers are not responsible for toll fraud and are entitled to collect the full amount owed regardless of whether the calls were due to theft of services. Also keep in mind, although federal authorities are supposed to investigate instances of toll fraud, cases of less than $100,000 in loss get little or no attention, owing to the huge workload generated by thousands of hackers. While toll fraud in itself does not necessarily represent a disaster, a $100,000 telephone bill can be disastrous to a career.

CONCLUSION

The areas in which a company can pay a significant financial penalty for lack of vigilance are too numerous to be covered in just one article. It pays to keep an eye on even the most mundane systems, to ensure that these minor conveniences of the office do not become a major strain on another important system — your cardiovascular system. Article 5-04-42 will discuss the other side of the coin, namely, securing your data carrier from the unauthorized intruder.

Leo A. Wrobel is president and CEO of Premiere Network Services, Inc., in DeSoto, TX. An active author, national and international lecturer, and technical futurist, he has published 10 books and over 100 trade articles on a variety of technical subjects, including *Writing Disaster Recovery Plans for Telecommunications and LANS* (Artech House Books, 1993) and *Business Resumption Planning* (Auerbach Publications, 1997). His experience of nearly two decades including assignments at AT&T, a major mortgage banking company, and a host of other firms engaged in baking, brokerage, heavy manufacturing, telecommunications services, and government, including the design and regulatory approval of a LATA-wide OC-12/ATM network for a $10 billion manufacturing giant, the first of its kind. A three-term city councilman and previous mayor, Leo Wrobel is a knowledgeable and effective communicator known for his entertaining presentation style on a wide variety of technical topics. For more information, contact his web site a http://www.dallas.net/~premiere or phone at (972) 228-8881.

51

Voice Security

Chris Hare, CISSP, CISA

Most security professionals in today's enterprise spend much of their time working to secure access to corporate electronic information. However, voice and telecommunications fraud still costs the corporate business communities millions of dollars each year. Most losses in the telecommunications arena stem from toll fraud, which is perpetrated by many different methods.

Millions of people rely upon the telecommunication infrastructure for their voice and data needs on a daily basis. This dependence has resulted in the telecommunications system being classed as a critical infrastructure component. Without the telephone, many of our daily activities would be more difficult, if not almost impossible.

When many security professionals think of voice security, they automatically think of encrypted telephones, fax machines, and the like. However, voice security can be much simpler and start right at the device to which your telephone is connected. This chapter looks at how the telephone system works, toll fraud, voice communications security concerns, and applicable techniques for any enterprise to protect its telecommunication infrastructure. Explanations of commonly used telephony terms are found throughout the chapter.

POTS: Plain Old Telephone Service

Most people refer to it as "the phone." They pick up the receiver, hear the dial tone, and make their calls. They use it to call their families, conduct business, purchase goods, and get help or emergency assistance. And they expect it to work all the time.

The telephone service we use on a daily basis in our homes is known in the telephony industry as POTS, or plain old telephone service. POTS is delivered to the subscriber through several components (see Exhibit 51.1):

- The telephone handset
- Cabling
- A line card
- A switching device

The telephone handset, or station, is the component with which the public is most familiar. When the customer picks up the handset, the circuit is closed and established to the switch. The line card signals to the processor in the switch that the phone is off the hook, and a dial tone is generated.

The switch collects the digits dialed by the subscriber, whether the subscriber is using a pulse phone or Touch-Tone®. A pulse phone alters the voltage on the phone line, which opens and closes a relay at the switch. This is the cause of the clicks or pulses heard on the line. With Touch-Tone dialing, a tone generator at the switch creates the tones for dialing the call.

The processor in the switch accepts the digits and determines the best way to route the call to the receiving subscriber. The receiving telephone set may be attached to the same switch, or connected to another halfway around the world. Regardless, the routing of the call happens in a heartbeat due to a very complex network of switches, signaling, and routing.

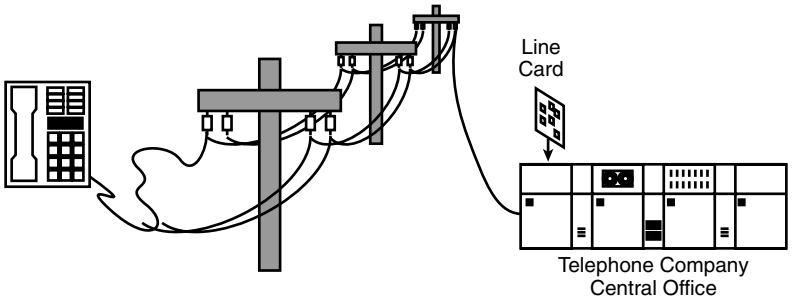


EXHIBIT 51.1 Components of POTS.

However, the process of connecting the telephone to the switching device, or to connect switching devices together to increase calling capabilities, uses lines and trunks.

Connecting Things Together

The problem with most areas of technology is with terminology. The telephony industry is no different. Trunks and lines both refer to the same thing — the circuitry and wiring used to deliver the signal to the subscriber. The fundamental difference between them is where they are used.

Both trunks and lines can be digital or analog. The line is primarily associated with the wiring from the telephone switch to the subscriber (see Exhibit 51.2). This can be either the residential or business subscriber, connected directly to the telephone company's switch, or to a PBX. Essentially, the line typically is associated with carrying the communications of a single subscriber to the switch.

The trunk, on the other hand, is generally the connection from the PBX to the telephone carrier's switch, or from one switch to another. A trunk performs the same function as the line. The only difference is the amount of calls or traffic the two can carry. Because the trunk is used to connect switches together, the trunk can carry much more traffic and calls than the line. The term *circuit* is often used to describe the connection from one device to the other, without attention to the type of connection, analog or digital, or the devices on either end (station or device).

Analog versus Digital

Both the trunk and the line can carry either analog or digital signals. That is to say, they can only carry one type at a time. Conceptually, the connection from origin to destination is called a circuit, and there are two principal circuit types.

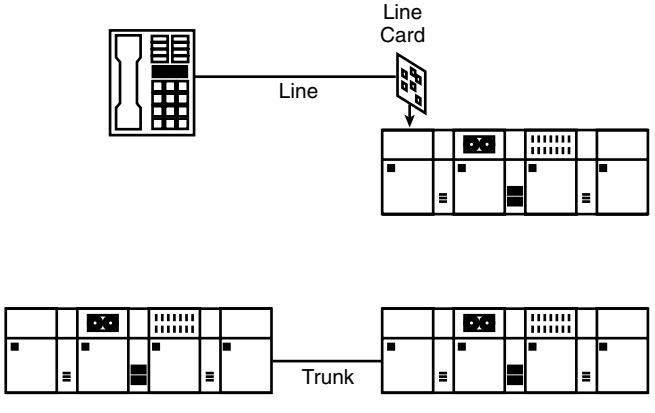


EXHIBIT 51.2 Trunks and lines.

Analog circuits are used to carry voice traffic and digital signals after conversion to sounds. While analog is traditionally associated with voice circuits, many voice calls are made and processed through digital equipment. However, the process of analog/digital conversion is an intense technical discussion and is not described here.

An analog circuit uses the variations in amplitude (volume) and frequency to transmit the information from one caller to the other. The circuit has an available bandwidth of 64K, although 8K of the available bandwidth is used for signaling between the handset and the switch, leaving 56K for the actual voice or data signals.

Think about connecting a computer modem to a phone line. The maximum available speed the modem can function at is 56K. The rationale for the 56K modem should be obvious now. However, most people know a modem connection is rarely made at 56K due to the quality of the circuit, line noise, and the distance from the subscriber to the telephone carrier's switch. Modems are discussed again later in the chapter.

Because analog lines carry the actual voice signals for the conversation, they can be easily intercepted. Anyone with more than one phone in his or her house has experienced the problem with eavesdropping. Anyone who can access the phone circuit can listen to the conversation. A phone tap is not really required — only knowledge of which wires to attach to and a telephone handset.

However, despite the problem associated with eavesdropping, many people do not concern themselves too much with the possibility someone may be listening to their phone call.

The alternative to analog is digital. While the analog line uses sound to transmit information, the digital circuit uses digital signals to represent data. Consequently, the digital circuit technologies are capable of carrying significantly higher speeds as the bandwidth increases on the circuit.

Digital circuits offer a number of advantages. They can carry higher amounts of data traffic and more simultaneous telephone calls than an analog circuit. They offer better protection from eavesdropping and wiretapping due to their design. However, despite the digital signal, any telephone station sharing the same circuit can still eavesdrop on the conversation without difficulty.

The circuits are not the principal cause of security problems. Rather, the concern for most enterprises and individuals arises from the unauthorized and inappropriate use of those circuits.

Lines and trunks can be used in many different ways and configurations to provide the required level of service. Typically, the line connected to a station offers both incoming and outgoing calls. However, this does not have to be the case in all situations.

Direct Inward Dial (DID)

If an outside caller must be connected with an operator before reaching his party in the enterprise, the system is generally called a key switch PBX. However, many PBX systems offer direct inward dial, or DID, where each telephone station is assigned a telephone number that connects the external caller directly to the call recipient.

Direct inward dial makes reaching the intended recipient easier because no operator is involved. However, DID also has disadvantages. Modems connected to DID services can be reached by authorized and unauthorized persons alike. It also makes it easier for individuals to call and solicit information from the workforce, without being screened through a central operator or attendant.

Direct Outward Dial (DOD)

Direct outward dial is exactly the opposite of DID. Some PBX installations require the user to select a free line on his phone or access an operator to place an outside call. With DOD, the caller picks up the phone, dials an access code, such as the digit 9, and then the external phone number. The call is routed to the telephone carrier and connected to the receiving person.

The telephone carrier assembles the components described here to provide service to its subscribers. The telephone carriers then interconnect their systems through gateways to provide the public switched telephone network.

The Public Switched Telephone Network (PSTN)

The pubic switched telephone network is a collection of telephone systems maintained by telephone carriers to provide a global communications infrastructure. It is called the public switched network because it is accessible to the general public and it uses circuit-switching technology to connect the caller to the recipient.

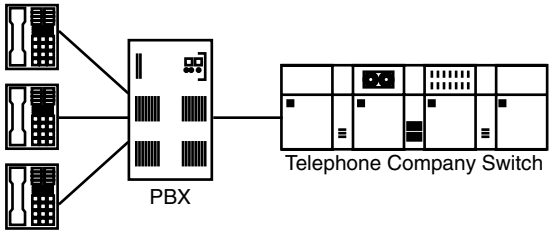


EXHIBIT 51.3 PBX connection.

The goal of the PSTN is to connect the two parties as quickly as possible, using the shortest possible route. However, because the PSTN is dynamic, it can often configure and route the call over a more complex path to achieve the call connection on the first attempt.

While this is extremely complex on a national and global scale, enterprises use a smaller version of the telephone carrier switch called a PBX (or private branch exchange).

The Private Area Branch Exchange (PABX)

The private area branch exchange, or PABX, is also commonly referred to as a PBX. Consequently, you will see the terms used interchangeably. The PBX is effectively a telephone switch for an enterprise; and, like the enterprise, it comes in different sizes. The PBX provides the line card, call processor, and some basic routing. The principal difference is how the PBX connects to the telephone carrier's network. If we compare the PBX to a router in a data network connecting to the Internet, both devices know only one route to send information, or telephone calls, to points outside the network (see Exhibit 51.3).

The PBX has many telephone stations connected to it, like the telephone carrier's switch. The PBX knows how to route calls to the stations connected directly to the same PBX. A call for an external telephone number is routed to the carrier's switch, which then processes the call and routes it to the receiving station.

Both devices have similar security issues, although the telephone carrier has specific concerns: the telephone communications network is recognized as a critical infrastructure element, and there is liability associated with failing to provide service. The enterprise rarely has to deal with these issues; however, the enterprise that fails to provide sufficient controls to prevent the compromise of its PBX may also face specific liabilities.

Network Class of Service (NCOS)

Each station on the phone PBX can be configured with a network class of service, or NCOS. The NCOS defines the type of calls the station can make. Exhibit 51.4 illustrates different NCOS levels.

When examining Exhibit 51.4, we can see that each different class of service offers new abilities for the user at the phone station. Typically, class of service is assigned to the station and not the individual, because few phone systems require user authentication before placing the call.

EXHIBIT 51.4 Network Class-of-Service Levels

Level	Internal	Local Seven-Digit Dialing	Local Ten-Digit Dialing	Domestic Long Distance	International Long Distance
1	X				
2	X	X	X		
3	X	X	X	X	
4	X	X	X	X	X

NOTE: Blocking specific phone numbers or area codes, such as 976, 900, or 809, is not done at the NCOS level but through other call-blocking methods available in the switch.

Through assigning NCOS to various phones, some potential security problems can be avoided. For example, if your enterprise has a phone in the lobby, it should be configured with a class of service low enough to allow calls to internal extensions or local calls only. Long distance should not be permitted from any open-area phone due to the cost associated with those calls.

In some situations, it may be desirable to limit the ability of a phone station to receive calls, while still allowing outgoing calls. This can be defined as another network class of service, without affecting the capabilities of the other stations.

However, not all PBX systems have this feature. If your enterprise systems have it, it should be configured to allow the employees only the ability to make the calls that are required for their specific job responsibilities.

Voicemail

Voicemail is ubiquitous with communications today. However, voicemail is often used as the path to the telephone system and free phone calls for the attacker — and toll fraud for the system owner.

Voicemail is used for recording telephone messages for users who are not available to answer their phones. Users access messages by entering an identifier, which is typically their phone extension number, and a password.

Voicemail problems typically revolve around password management. Because voicemail must work with the phone, the password can only contain digits. This means attacking the password is relatively trivial from the attacker's perspective. Consequently, the traditional password and account management issues exist here as in other systems:

- Passwords the same as the account name
- No password complexity rules
- No password aging or expiry
- No account lockout
- Other voicemail configuration issues

A common configuration problem is through-dialing. With through-dialing, the system accepts a phone number and places the call. The feature can be restricted to allow only internal or local numbers, or to disable it. If through-dialing is allowed and not properly configured, the enterprise now pays the bills for the long-distance or other toll calls made.

Attackers use stale mailboxes — those that have not been accessed in a while — to attempt to gain access to the mailbox. If the mailbox password is obtained, and the voicemail system is configured to allow through-dialing, the attackers are now making free calls. The attacker first changes the greeting on the mailbox to a simple "yes." Now, any collect call made through an automated system expecting the word response "yes" is automatically accepted. The enterprise pays the cost of the call.

The attacker enters the account identifier, typically the phone extension for the mailbox, and the password. Once authenticated by the voicemail system, the attacker then enters the appropriate code and phone number for the external through-call. If there are no restrictions on the digits available, the attacker can dial any phone number anywhere in the world.

The scenario depicted here can be avoided using simple techniques applicable to most systems:

- Change the administrator and attendant passwords.
- Do not use the extension number as the initial password.
- Disable through-dialing.
- Configure voicemail to use a minimum of six digits for the password.
- Enable password history options if available.
- Enable password expiration if available.
- Remove stale mailboxes.

Properly configured, voicemail is a powerful tool for the enterprise, as is the data network and voice conferencing.

Voice Conferencing

Many enterprises use conference calls to regularly conduct business. In the current economic climate, many enterprises use conference calls as the cost-efficient alternative to travel for meetings across disparate locations.

The conference call uses a "bridge," which accepts the calls and determines which conference the caller is to be routed to based upon the phone number and the conference call password.

The security options available to the conference call bridge are technology dependent. Regardless, participants on the conference call should be reminded not to discuss enterprise-sensitive information because anyone who acquires or guesses the conference call information could join the call. Consequently, conference call participant information should be protected to limit participation.

Conference bridges are used for single-time, repetitive, and ad hoc calls using various technologies. Some conference call vendors provide services allowing anyone in the enterprise to have an on-demand conference bridge. These conference bridges use a "host" or chairperson who must be present to start the conference call. The chairperson has a second passcode, used to initiate the call. Any user who learns the host or chairperson code can use the bridge at any time.

Security issues regarding conference bridges include:

- Loss of the chairperson code
- Unauthorized use of the bridge
- Inappropriate access to the bridge
- Loss of sensitive information on the bridge

All of these issues are addressed through proper user awareness — which is fortunate because few enterprises actually operate their own conference bridge, relying instead upon the telephone carrier to maintain the configurations.

If possible, the conference bridge should be configured with the following settings and capabilities:

- The conference call cannot start until the chairperson is present.
- All participants should be disconnected when the chairperson disconnects from the bridge.
- The chairperson should have the option of specifying a second security access code to enter the bridge.
- The chairperson should have commands available to manipulate the bridge, including counting the number of ports in use, muting or un-muting the callers, locking the bridge, and reaching the conference operator.

The chairperson's commands are important for the security of the conference call. Once all participants have joined, the chairperson should verify everyone is there and then lock the bridge. This prevents anyone from joining the conference call.

Security Issues

Throughout the chapter, we have discussed technologies and security issues. However, regardless of the specific configuration of the phone system your enterprise is using, there are some specific security concerns you should be knowledgeable of.

Toll Fraud

Toll fraud is a major concern for enterprises, individuals, and the telephone carriers. Toll fraud occurs when toll-based or chargeable telephone calls are fraudulently made. There are several methods of toll fraud, including inappropriate use by authorized users, theft of services, calling cards, and direct inward dialing to the enterprise's communications system.

According to a 1998 *Consumer News* report, about $4 billion are lost to toll fraud annually. The report is available online at the URL http://www.fcc. gov/Bureaus/Common_Carrier/Factsheets/ttf&you.pdf.

The cost of the fraud is eventually passed on to the businesses and consumers through higher communications costs. In some cases, the telephone carrier holds the subscriber responsible for the charges, which can be devastating. Consequently, enterprises can pay for toll fraud insurance, which pays the telephone carrier

after the enterprise pays the deductible. While toll fraud insurance sounds appealing, it is expensive and the deductibles are generally very high.

It is not impossible to identify toll fraud within your organization. If you have a small enterprise, simply monitoring the phone usage for the various people should be enough to identify calling patterns. For larger organizations, it may be necessary to get calling information from the PBX for analysis. For example, if you can capture the call records from each telephone call, it is possible to assign a cost for each telephone call.

Inappropriate Use of Authorized Access

Every employee in an enterprise typically has a phone on the desk, or access to a company-provided telephone. Most employees have the ability to make long-distance toll calls from their desks. While most employees make long-distance calls on a daily basis as part of their jobs, many will not think twice to make personal long-distance calls at the enterprise's expense.

Monitoring this type of usage and preventing it is difficult for the enterprise. Calling patterns, frequently called *number analysis*, and advising employees of their monthly telecommunications costs are a few ways to combat this problem. Additionally, corporate policies regarding the use of corporate telephone services and penalties for inappropriate use should be established if your enterprise does not have them already. Finally, many organizations use billing or authorization codes when making long-distance phone calls to track the usage and bill the charges to specific departments or clients.

However, if your enterprise has its own PBX with conditional toll deny (CTD) as a feature, you should considering enabling this on phone stations where long-distance or toll calls are not permitted. For example, users should not be able to call specific phone numbers or area codes. Alternatively, a phone station may be denied toll-call privileges altogether.

However, in Europe, implementing CTD is more difficult because it is not uncommon to call many different countries in a single day. Consequently, management of the CTD parameters becomes very difficult. CTD can be configured as a specific option in an NCOS definition, as discussed earlier in the chapter.

Calling Cards

Calling cards are the most common form of toll fraud. Calling-card numbers are stolen and sold on a daily basis around the world. Calling-card theft typically occurs when an individual observes the subscriber entering the number into a public phone. The card number is then recorded by the thief and sold to make other calls.

Calling-card theft is a major problem for telephone carriers, who often have specific fraud units for tracking thieves, and calling software, which monitors the calling patterns and alerts the fraud investigators to unusual calling patterns.

In some cases, hotels will print the calling-card number on the invoices provided to their guests, making the numbers available to a variety of people. Additionally, if the PBX is not configured correctly, the calling-card information is shown on the telephone display, making it easy for anyone nearby to see the digits and use the number.

Other PBX-based problems include last number redial. If the PBX supports last number redial, any employee can recall the last number dialed and obtain the access and calling-card numbers.

Employees should be aware of the problems and costs associated with the illegitimate use of calling cards. Proper protection while using a calling card includes:

- Shielding the number with your hands when entering it
- Memorizing the number so you do not have a card visible when making the call
- Ensuring your company PBX does not store the digits for last number redial
- Ensuring your enterprise PBX does not display the digits on the phone for an extended period of time

Calling cards provide a method for enterprise employees to call any number from any location. However, some enterprises may decide this is not appropriate for their employees. Consequently, they may offer DISA access to the enterprise phone network as an alternative.

DISA

Direct inward system access, or DISA, is a service available on many PBX systems. DISA allows a user to dial an access number, enter an authorization code, and appear to the PBX as an extension. This allows callers to make calls as if they were in the office building, whether the calls are internal to the PBX or external to the enterprise.

DISA offers some distinct advantages. For example, it removes the need to provide calling cards for employees because they can call a number and be part of the enterprise voice network. Additionally, long-distance calls placed through DISA services are billed at the corporate rate because the telephone carrier sees the calls as originating from the enterprise.

DISA's advantages also represent problems. If the DISA access number becomes known, unauthorized users only need to try random numbers to form an authorization code. Given enough time, they will eventually find one and start making what are free calls from their perspective. However, your enterprise pays the bill.

DISA authorization codes, which must be considered passwords, are numeric only because there is no way to enter alphabetic letters on the telephone keypad. Consequently, even an eight-number authorization code is easily defeated.

If your organization does use DISA, there are some things you can do to assist in preventing fraudulent access of the service:

- Frequent analysis of calling patterns
- Monthly "invoices" to the DISA subscribers to keep them aware of the service they are using
- Using a minimum of eight-digit authorization codes
- Forcing changes of the authorization codes every 30 days
- Disabling inactive DISA authorization codes if they are not used for a prescribed period of time or a usage limit is reached
- Enabling authorization code alarms to indicate attempts to defeat or guess DISA authorization codes

The methods discussed are often used by attackers to gain access to the phone system and make unauthorized telephone calls. However, technical aspects aside, some of the more skillful events occur through social engineering techniques.

Social Engineering

The most common ploy from a social engineering perspective is to call an unsuspecting person, indicate the attacker is from the phone company, and request an outside line. The attacker then makes the phone call to the desired location, talks for as long as required, and hangs up. As long as the attacker can find numbers to dial and does not have to go through a central operator, this can go on for months.

Another social engineering attack occurs when a caller claims to be a technical support person. The attacker will solicit confidential information, such as passwords, access numbers, or ID information, all under the guise of providing support or maintenance support to ensure the user's service is not disrupted. In actuality, the attacker is gathering sensitive information for better understanding of the enterprise environment and enabling him to perform an attack.

Other Voice Services

There are other voice services that also create issues for the enterprise, including modems, fax, and wireless services.

Modems

Modems are connected to the enterprise through traditional technologies using the public switched telephone network. Modems provide a method of connectivity through the PSTN to the enterprise data network. When installed on a DID circuit, the modem answers the phone when an incoming call is received. Attackers have regularly looked for these modems using war-dialing techniques.

If your enterprise must provide modems to connect to the enterprise data network, these incoming lines should be outside the enterprise's normal dialing range. This makes it more difficult for the attacker to find.

However, because many end stations are analog, the user could connect the modem to the desktop phone without anyone's knowledge.

This is another advantage of digital circuits. While digital-to-analog converters exist to connect a modem to a digital circuit, this is not infallible technology. Should your enterprise use digital circuits to the desktop, you should implement a program to document and approve all incoming analog circuits and their purpose. This is very important for modems due to their connectivity to the data network.

Fax

The fax machine is still used in many enterprises to send information not easily communicated through other means. The fax transmission sends information such as scanned documents to the remote fax system. The principal concern with fax is the lack of control over the document at the receiving end.

For example, if a document is sent to me using a fax in a shared area, anyone who checks the fax machine can read the message. If the information in the fax is sensitive, private, or otherwise classified, control of the information should not be considered lost.

A second common problem is misdirected faxes. That is, the fax is successfully transmitted, but to the wrong telephone number. Consequently, the intended recipient does not receive the fax.

However, fax can be controlled through various means such as dedicated fax machines in controlled areas. For example,

- Contact the receiver prior to sending the fax.
- Use a dedicated and physically secure fax machine if the information requires it.
- Use a cover page asking for immediate delivery to the recipient.
- Use a cover page asking for notification if the fax is misdirected.

Fax requires the use of analog lines because it uses a modem to establish the connection. Consequently, the inherent risks of the analog line are applicable here. If an attacker can monitor the line, he may be able to intercept the modem tones from the fax machine and read the fax. Addressing this problem is achieved through encrypted fax if document confidentiality is an ultimate concern.

Encrypted fax requires a common or shared key between the two fax machines. Once the connection is established, the document is sent using the shared encryption key and subsequently decoded and printed on the receiving fax machine. If the receiving fax machine does not have the shared key, it cannot decode the fax. Given the higher cost of the encrypted fax machine, it is only a requirement for the most highly classified documents.

Cellular and Wireless Access

Cellular and wireless access to the enterprise is also a problem due to the issues associated with cellular. Wireless access in this case does not refer to wireless access to the data network, but rather wireless access to the voice network.

However, this type of access should concern the security professional because the phone user will employ services such as calling cards and DISA to access the enterprise's voice network. Because cellular and wireless access technologies are often subject to eavesdropping, the DISA access codes or calling card could potentially be retrieved from the wireless caller.

The same is true for conversations — if the conversation between the wireless caller and the enterprise user is of a sensitive nature, it should not be conducted over wireless. Additionally, the chairperson for a conference call should find out if there is anyone on the call who is on a cell phone and determine if that level of access is appropriate for the topic to be discussed.

Voice-over-IP: The Future

The next set of security challenges for the telecommunications industry is Voice-over-IP. The basis for the technology is to convert the voice signals to packets, which are then routed over the IP network. Unlike the traditional circuit-switched voice network, Voice-over-IP is a packet-switched network. Consequently, the same types of problems found in a data network are found in the Voice-over-IP technology.

There are a series of problems in the Voice-over-IP technologies, on which the various vendors are collaborating to establish the appropriate standards to protect the privacy of the Voice-over-IP telephone call. Some of those issues include:

- No authentication of the person making the call
- No encryption of the voice data, allowing anyone who can intercept the packet to reassemble it and hear the voice data
- Quality of service, because the data network has not been traditionally designed to provide the quality-of-service levels associated with the voice network

The complexities in the Voice-over-IP arena for both the technology and related security issues will continue to develop and resolve themselves over the next few years.

Summary

This chapter introduced the basics of telephone systems and security issues. The interconnection of the telephone carriers to establish the public switched telephone network is a complex process. Everyone demands a dial tone when they pick up the handset. Such is the nature of this critical infrastructure.

However, enterprises often consider the telephone their critical infrastructure as well, whether they get their service directly from the telephone carrier or use a PBX to provide internal services, which is connected to the public network.

The exact configurations and security issues are generally very specific to the technology in use. This chapter has presented some of the risks and prevention methods associated with traditional voice security. The telephone is the easiest way to obtain information from a company and the fastest method of moving information around in a nondigital form. Aside from implementing the appropriate configurations for your technologies, the best defense is ensuring your users understand their role in limiting financial and information losses through the telephone network.

Acknowledgments

The author wishes to thank Beth Key, a telecommunications security and fraud investigator from Nortel Networks' voice service department. Ms. Key provided valuable expertise and support during the development of this chapter.

Mignona Cote of Nortel Networks' security vulnerabilities team provided her experiences as an auditor in a major U.S. telecommunications carrier prior to joining Nortel Networks.

The assistance of both these remarkable women contributed to the content of this chapter, and they are examples of the quality and capabilities of the women in our national telecommunications industry.

References

PBX Vulnerability Analysis, Finding Holes in Your PBX before Someone Else Does, U.S. Department of Commerce, NIST Special Pub. 800-24, http://csrc.nist.gov/publications/nistpubs/800-24/sp800-24pbx.pdf.

Security for Private Branch Exchange Systems, http://csrc.nist.gov/publications/nistbul/ itl00-08.txt.

52

Secure Voice Communications (VoI)

Valene Skerpac, CISSP

Voice communication is in the midst of an evolution toward network convergence. Over the past several decades, the coalescence of voice and data through the circuit-based, voice-centric public switched telephone network (PSTN) has been limited. Interconnected networks exist today, each maintaining its own set of devices, services, service levels, skill sets, and security standards. These networks anticipate the inevitable and ongoing convergence onto packet- or cell-based, data-centric networks primarily built for the Internet. Recent deregulation changes and cost savings, as well as the potential for new media applications and services, are now driving a progressive move toward voice over some combination of ATM, IP, and MPLS. This new-generation network aims to include novel types of telephony services that utilize packet-switching technology to receive transmission efficiencies while also allowing voice to be packaged in more standard data applications. New security models that include encryption and security services are necessary in telecommunication devices and networks.

This chapter reviews architectures, protocols, features, quality-of-service (QoS), and security issues associated with traditional circuit-based landline and wireless voice communication. The chapter then examines convergence architectures, the effects of evolving standards-based protocols, new quality-of-service methods, and related security issues and solutions.

Circuit-Based PSTN Voice Network

The PSTN has existed in some form for over 100 years. It includes telephones, local and interexchange trunks, transport equipment, and exchanges; and it represents the whole traditional public telephone system. The foundation for the PSTN is dedicated 64 kbps circuits. Two kinds of 64 kbps pulse code modulation techniques are used to encode human analog voice signals into digital streams of 0s and 1s (mu-law, the North American standard; and a-law, the European standard).

The PSTN consists of the local loop that physically connects buildings via landline copper wires to an end-office switch called the central office or Class 5 switch. Communication between central offices connected via trunks is performed through a hierarchy of switches related to call patterns. Many signaling techniques are utilized to perform call control functions. For example, analog connections to the central office use dual-tone multifrequency (DTMF) signaling, an in-band signaling technique transmitted over the voice path. Central office connections through a T1/E1 or T3/E3 use in-band signaling techniques such as MF or robbed bit.

After World War II, the PSTN experienced high demand for greater capacity and increased function. This initiated new standards efforts, which eventually led to the organization in 1956 of the CCITT, the Comité Consultatif International de Télephonie et de Télégraphie, also known as the ITU-T, International Telecommunication Union Telecommunication Standardization Sector. Recommendations known as Signaling System 7 (SS7) were created, and in 1980 a version was completed for implementation. SS7 is a means of sending messages between switches for basic call control and for custom local area signaling services (CLASS). The move to SS7 represented a change to common-channel signaling versus its predecessor, per-trunk signaling.

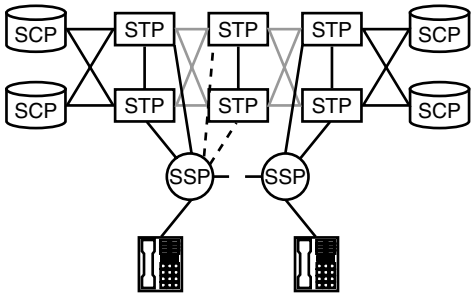


EXHIBIT 52.1 Diagram of SS7 key components and links.

SS7 is fundamental to today's networks. Essential architectural aspects of SS7 include a packet data network that controls and operates on top of the underlying voice networks. Second, a completely different transmission path is utilized for signaling information of voice and data traffic. The signaling system is a packet network optimized to speedily manage many signaling messages over one channel; it supports required functions such as call establishment, billing, and routing. Architecturally, the SS7 network consists of three components, as shown in Exhibit 52.1: service switch points (SSPs), service control points (SCPs), and signal transfer points (STPs). SSP switches originate and terminate calls communicating with customer premise equipment (CPE) to process calls for the user. SCPs are centralized nodes that interface with the other components through the STP to perform functions such as digit translation, call routing, and verification of credit cards. ScPs manage the network configuration and call-completion database to perform the required service logic. STPs translate and route SS7 messages to the appropriate network nodes and databases. In addition to the SS7 signaling data link, there are a number of other SS7 links between the SS7 components whereby certain links help to ensure a reliable SS7 network.

Functional benefits of SS7 networks include reduced post-dialing delay, increased call completion, and connection to the intelligent network (IN). SS7 supports shared databases among switches, providing the groundwork for IN network-based services such as 800 services and advanced intelligent networks (AINs). SS7 enables interconnection and enhanced services, making the whole next generation and conversion possible.

The PSTN assigns a unique number to each telephone line. There are two numbering plans: the North American numbering plan (NANP) and the ITU-T international numbering plan. NANP is an 11-digit or 1+10 dialing plan, whereas the ITU-T is no more than 15 digits, depending on the needs of the country.

Commonly available PSTN features are call waiting, call forwarding, and three-way calling. With SS7 end to end, CLASS features such as ANI, call blocking, calling line ID blocking, automatic callback, and call return (*69) are ready for use. Interexchange carriers (IXCs) sell business features including circuit-switched long distance, calling cards, 800/888/877 numbers, VPNs (where the telephone company manages a private dialing plan), private leased lines, and virtual circuits (Frame Relay or ATM). Security features may include line restrictions, employee authorization codes, virtual access to private networks, and detailed call records to track unusual activity. The PSTN is mandated to perform emergency services. The basic U.S. 911 relays the calling party's telephone number to public safety answering points (PSAPs). Enhanced 911 requirements include the location of the calling party, with some mandates as stringent as location within 50 meters of the handset.

The traditional enterprise private branch exchange (PBX) is crucial to the delivery of high availability, quality voice, and associated features to the end user. It is a sophisticated proprietary computer-based switch that operates as a small, in-house phone company with many features and external access and control. The PBX architecture separates switching and administrative functions, is designed for 99.999 percent reliability, and often integrates with a proprietary voicemail system. Documented PBX threats and baseline security methods are well known and can be referenced in the document *PBX Vulnerability Analysis* by NIST, special publication 800-24. Threats to the PBX include toll fraud theft, eavesdropping on conversations, unauthorized access to routing and address data, data alteration of billing information and system tables to gain additional services, unauthorized access, denial-of-service attacks, and a passive traffic analysis attack. Voice messages are also prone to threats of eavesdropping and accidental or purposeful forwarding. Baseline security policies and controls methods, which to a certain extent depend on the proprietary equipment, need to be implemented. Control methods include manual assurance of database integrity, physical security, operations security, management-initiated controls, PBX system control, and PBX system terminal access control such as password

control. Many telephone and system configuration practices need to be developed and adhered to. These include blocking well-known non-call areas or numbers, restart procedures, software update protection using strong error detection based on cryptography, proper routing through the PBX, disabling open ports, and configuration of each of the many PBX features. User quality-of-service (QoS) expectations of basic voice service are quite high in the area of availability. When people pick up the telephone, they expect a dial tone. Entire businesses are dependant on basic phone service, making availability of service critical. Human voice interaction requires delays of no more that 250 milliseconds.

Carriers experienced fraud prior to the proliferation of SS7 out-of-band signaling utilized for the communication of call establishment and billing information between switches. Thieves attached a box that generated the appropriate signaling tones, permitting a perpetrator to take control of signaling between switches and defeat billing. SS7 enhanced security and prevented unauthorized use.

Within reasonable limitations, PSTN carriers have maintained *closed* circuit-based networks that are not open to public protocols except under legal agreements with specified companies. In the past, central offices depended on physical security, passwords system access, a relatively small set of trained individuals working with controlled network information, network redundancy, and deliberate change control. U.S. telephone carriers are subject to the Communications Assistance for Law Enforcement Act (CALEA) and need to provide access points and certain information when a warrant has been issued for authorized wiretapping.

The network architecture and central office controls described above minimized security exposures, ensuring that high availability and QoS expectations were essentially met. While it is not affordable to secure the entire PSTN, such are the requirements of certain government and commercial users. Encryption of the words spoken into a telephone and decryption of them as they come out of the other telephone is the singular method to implement a secure path between two telephones at arbitrary locations. Such a secure path has never broadly manifested itself cost-effectively for commercial users.

Historically, PSTN voice scramblers have existed since the 1930s but equipment was large, complicated, and costly. By the 1960s, the KY-3 came to market as one of the first practical voice encryption devices. The secure telephone unit, first generation (STU-1) was introduced in 1970, followed in 1975 by the STU-II used by approximately 10,000 users. In 1987, the U.S. National Security Agency (NSA) approved STU-III and made secure telephone service available to defense contractors where multiple vendors such as AT&T, GE, and Motorola offered user-friendly deskset telephones for less than U.S.$2000. During the 1990s, systems came to market such as an ISDN version of STU called STE, offered by L3 Communications, AT&T Clipper phone, Australian Speakeasy, and British Brent telephone. Also available today are commercial security telephones or devices inserted between the handset and telephone that provide encryption at costs ranging from U.S.$100 to $2000, depending on overall capability.

Wireless Voice Communication Networks

Wireless technology in radio form is more than 100 years old. Radio transmission is the induction of an electrical current at a remote location, intended to communicate information whereby the current is produced via the propagation of an electromagnetic wave through space. The wireless spectrum is a space that the world shares, and there are several methods for efficient spectrum reuse. First, the space is partitioned into smaller coverage areas or cells for the purpose of reuse. Second, a multiple access technique is used to allow the sharing of the spectrum among many users. After the space has been specified and multiple users can share a channel, spread spectrum, duplexing, and compression techniques to utilize the bandwidth with even better efficiency are applied.

In digital cellular systems, time division multiplexing (TDMA) and code division multiple (CDMA) access techniques exist. TDMA first splits the frequency spectrum into a number of channels and then applies time division multiplexing to operate multiple users interleaved in time. TDMA standards include Global System for Mobile Communications (GSM), Universal Wireless Communications (UWC), and Japanese Digital Cellular (JDC). CDMA employs universal frequency reuse, whereby everybody utilizes the same frequency at the same time and each conversation is uniquely encoded, providing greater capacity over other techniques. First-generation CDMA standards and second-generation wideband CDMA (WCDMA) both use a unique code for each conversation and a spread spectrum method. WCDMA uses bigger channels, providing for greater call capacity and longer encoding strings than CDMA, increasing security and performance.

Multiple generations of wireless WANs have evolved in a relatively short period of time. The first-generation network used analog transmission and was launched in Japan in 1979. By 1992, second-generation (2G) digital

networks were operational at speeds primarily up to 19.2 kbps. Cellular networks are categorized as analog and digital cellular, whereas PCS, a shorter-range, low-power technology, was digital from its inception. Today, cellular networks have evolved to the 2.5G intermediate-generation network, which provides for enhanced data services on present 2G digital platforms. The third-generation (3G) network includes digital transmission. It also provides for an always-on per-user and terminal connection that supports multimedia broadband applications and data speeds of 144 kbps to 384 kbps, potentially up to 2 Mbps in certain cases. The 3G standards are being developed in Europe and Asia, but worldwide deployment has been slow due to large licensing and build costs. There are many competing cellular standards that are impeding the overall proliferation and interoperability of cellular networks.

Digital cellular architecture, illustrated in Exhibit 52.2, resembles the quickly disappearing analog cellular network yet is expanded to provide for greater capacity, improved security, and roaming capability. A base transceiver station (BTS), which services each cell, is the tower that transmits signals to and from the mobile unit. Given the large number of cells required to address today's capacity needs, a base station controller (BSC) is used to control a set of base station transceivers. The base station controllers provide information to the mobile switching center (MSC), which accesses databases that enable roaming, billing, and interconnection. The mobile switching center interfaces with a gateway mobile switching center that interconnects with the PSTN.

The databases that make roaming and security possible consist of a home location register, visitor location register, authentication center, and equipment identity register. The home location register maintains subscriber information, with more extensive management required for those registered to that mobile switching center area. The visitor location register logs and periodically forwards information about calls made by roaming subscribers for billing and other purposes. The authentication center is associated with the home location register; it protects the subscriber from unauthorized access, delivering security features including encryption, customer identification, etc. The equipment identity register manages a database of equipment, also keeping track of stolen or blacklisted equipment.

Prior to digital cellular security techniques, there was a high amount of toll fraud. Thieves stood on busy street corners, intercepted electronic identification numbers and phone numbers, and then cloned chips. The digitization of identification information allowed for its encryption and enhanced security. Policies and control methods are required to further protect against cellular phone theft. Methods include the use of an encrypted PIN code to telephone access and blocking areas or numbers. Privacy across the air space is improved using digital cellular compression and encoding techniques; CDMA encoding offers the greatest protection of the techniques discussed.

Despite security improvements in the commercial cellular networks, end-to-end security remains a challenge. Pioneering efforts for many of the digital communication, measurement, and data techniques available today

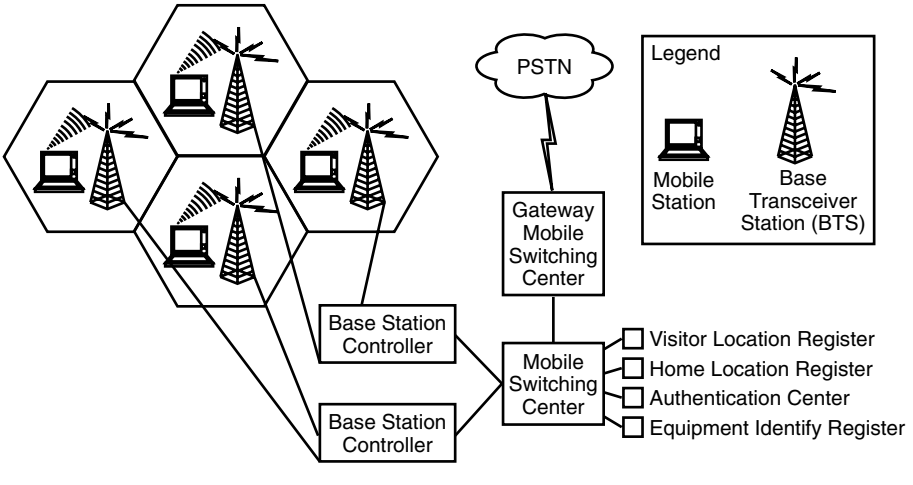


EXHIBIT 52.2 Digital cellular architecture.

were performed in a successful attempt to secure voice communication using FSK–FDM radio transmission during World War II. The SIGSALY system was first deployed in 1943 by Bell Telephone Laboratories, who began the investigation of encoding techniques in 1936 to change voice signals into digital signals and then reconstruct the signals into intelligible voice. The effort was spurred on by U.K. and U.S. allies who needed a solution to replace the vulnerable transatlantic high-frequency radio analog voice communications system called A-3. SIGSALY was a twelve-channel system; ten channels each measured the power of the voice signal in a portion of the whole voice frequency spectrum between 250 and 3000 Hz, and two channels provided information regarding the pitch of the speech and presence of unvoiced (hiss) energy. Encryption keys were generated from thermal noise information (output of mercury-vapor rectifier vacuum tubes) sampled every 20 milliseconds and quantized into six levels of equal probability. The level information was converted into channels of a frequency-shift-keyed audio tone signal, which represented the encryption key, and was then recorded on three hard vinyl phonograph records. The physical transportation and distribution of the records provided key distribution.

In the 1970s, U.S. Government wireless analog solutions for high-grade end-to-end crypto and authentication became available, though still at a high cost compared to commercial offerings. Secure telephone solutions included STU-III compatible, Motorola, and CipherTac2K. STU-III experienced compatibility problems with 2G and 3G networks. This led to the future narrow-band digital terminal (FNBDT) — a digital secure voice protocol operating at the transport layer and above for most data/voice network configurations across multiple media — and mixed excitation linear prediction vocoder (MELP) — an interoperable 2400-bps vocoder specification. Most U.S. Government personnel utilize commercial off-the-shelf solutions for sensitive but unclassified methods that rely on the commercial wireless cellular infrastructure.

Network Convergence

Architecture

Large cost-saving potentials and the promise of future capabilities and services drive the move to voice over a next-generation network. New SS7 switching gateways are required to support legacy services and signaling features and to handle a variety of traffic over a data-centric infrastructure. In addition to performing popular IP services, the next-generation gateway switch needs to support interoperability between PSTN circuits and packet-switching networks such as IP backbones, ATM networks, Frame Relay networks, and emerging Multi-Protocol Label Switching (MPLS) networks. A number of overlapping multimedia standards exist, including H.323, Session Initiation Protocol (SIP), and Media Gateway Control Protocol (MGCP). In addition to the telephony-signaling protocols encompassed within these standards, network elements that facilitate VoIP include VoIP gateways, the Internet telephony directory, media gateways, and softswitches. An evolution and blending of protocols, and gateway and switch functions continues in response to vendors' competitive searches for market dominance.

Take an example of a standard voice call initiated by a user located in a building connected to the central office. The central office links to an SS7 media gateway switch that can utilize the intelligence within the SS7 network to add information required to place the requested call. The call then continues on a packet basis through switches or routers until it reaches a destination media gateway switch, where the voice is unpackaged, undigitalized, and sent to the phone called.

Voice-over-IP (VoIP) changes voice into packets for transmission over a TCP/IP network. VoIP gateways connect the PSTN and the packet-switched Internet and manage the addressing across networks so that PCs and phones can talk to each other. Exhibit 52.3 illustrates major VoIP network components. The VoIP gateway performs packetization and compression of the voice, enhancement of the voice through voice techniques, DTMF signaling capability, voice packet routing, user authentication, and call detail recording for billing purposes. Many solutions exist, such as enterprise VoIP gateway routers, IP PBXs, service-provider VoIP gateways, VoIP access concentrators, and SS7 gateways. The overlapping functionality of the different types of gateways will progress further as mergers and acquisitions continue to occur. When the user dials the number from a VoIP telephone, the VoIP gateway communicates the number to the server; the call-agent software (softswitch) decides what the IP address is for the destination call number and presents back the IP address to the VoIP gateway. The gateway converts the voice signal to IP format, adds the address of the destination node, and sends the signal. The softswitch could be utilized again if enhanced services are required for additional functions.

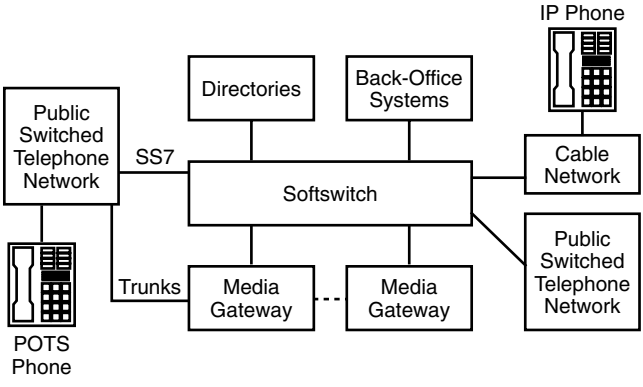


EXHIBIT 52.3 VoIP network architecture.

Media gateways interconnect with the SS7 network, enabling interoperability between the PSTN and packet-switched domains. They handle IP services and support various telephony-signaling protocols and Class 4 and Class 5 services. Media servers include categories of VoIP trunking gateways, VoIP access gateways, and network access service devices.

Vocoders compress and transmit audio over the network; they are another evolving area of standards for Voice-over-the-Internet (VOI). Vocoders used for VoI such as G.711 (48, 56, and 64 kbps high-bit rate) and G.723 (5.3 and 6.3 kbps high-bit rate) are based on existing standards created for digital telephony applications, limiting the telephony signal band of 200–3400 Hz with 8 kHz sampling. This toll-level audio quality is geared for the minimum a human ear needs to recognize speech and is not nearly that of face-to-face communications. With VoIP in a wideband IP end-to-end environment, better vocoders are possible that can achieve more transparent communication and better speaker recognition. New ITU vocoders — G.722.1 operating at 24 kbps and 32 kbps rates and 16 kHz sampling rate — are now used in some IP phone applications. The third-generation partnership project (3GPP)/ETSI (for GSM and WCDMA) merged on the adaptive multi-rate wideband (AMR-WB) at the 50–7000 Hz bandwidth to form the newly approved ITU G722.2 standard, which provides better voice quality at reduced bit rates and allows seamless interface between VoIP systems and wireless base stations. This eliminates the normal degradation of voice quality between vocoders of different systems.

Numbering

The Internet telephony directory, an IETF RFC known as ENUM services, is an important piece in the evolving VoI solution. ENUM is a standard for mapping telephone numbers to an IP address, a scheme wherein DNS maps PSTN phone numbers to appropriate URLs based on the E.164 standard.

To enable a faster time to market, VoIP continues as new features and service models supporting the PSTN and associated legacy standards are introduced. For example, in response to DTMF tone issues, the IETF RFC *RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals* evolved, which specifies how to carry and format tones and events using RTP. In addition to the incorporation of traditional telephone features and new integrated media features, VoIP networks need to provide emergency services and comply with law enforcement surveillance requirements. The requirements as well as various aspects of the technical standards and solutions are evolving.

The move toward IP PBXs is evolving. Companies that cost-effectively integrate voice and data between locations can utilize IP PBXs on their IP networks, gaining additional advantages from simple moves and changes. Challenges exist regarding the nonproprietary telephony-grade server reliability (built for 99.99 percent reliability) and power distribution compared to traditional PBXs. Complete solutions related to voice quality, QoS, lack of features, and cabling distance limitations are yet evolving. A cost-effective, phased approach to an IP converged system (for example, an IP card in a PBX) enables the enterprise to make IP migration choices, support new applications such as messaging, and maintain the traditional PBX investment where appropriate. The move toward computer telephony greatly increases similar types of PBX security threats discussed previously and is explored further in the "VoI Security" section of this chapter.

Quality-of-Service (QoS)

Network performance requirements are dictated by both the ITU SS7/C7 standards and user expectations. The standard requires that the end-to-end call-setup delay cannot exceed 20 to 30 seconds after the ISDN User Part (ISUP) initial address message (IAM) is sent; users expect much faster response times. Human beings do not like delays when they communicate; acceptable end-to-end delays usually need to meet the recommended 150 milliseconds.

QoS guarantees, at very granulated levels of service, are a requirement of next-generation voice networks. QoS is the ability to deliver various levels of service to different kinds of traffic or traffic flows, providing the foundation for tiered pricing based on class-of-service (CoS) and QoS. QoS methods fall into three major categories: first is an architected approach such as ATM; second is a per-flow or session method such as with the reservation protocol of IETF IntServ definitions and MPLS specifications; and third is a packet labeling approach utilizing a QoS priority mark as specified in 802.1p and IETF DiffServ.

ATM is a cell-based (small cell), wide area network (WAN) transport that came from the carrier environment for streaming applications. It is connection oriented, providing a way to set up a predetermined path between source and destination, and it allows for control of network resources in real-time. ATM network resource allocation of CoS and QoS provisioning is well defined; there are four service classes based on traffic characteristics. Further options include the definition of QoS and traffic parameters at the cell level that establish service classes and levels. ATM transmission-path virtual circuits include virtual paths and their virtual channels. The ATM virtual path groups the virtual channels that share the same QoS definitions, easing network management and administration functions.

IP is a flexible, efficient, connectionless, packet-based network transport that extends all the way to the desktop. Packet-switching methods have certain insufficiencies, including delays due to store-and-forward packet-switching mechanisms, jitter, and packet loss. Jitter is the delay in sending bits between two switches. Jitter results in both an end-to-end delay and delay differences between switches that adversely affect certain applications. As congestion occurs at packet switches or routers, packets are lost, hampering real-time applications. Losses of 30 or 40 percent in the voice stream could result in speech with missing syllables that sounds like gibberish.

IntServ and DiffServ are two IP schemes for QoS. IntServ broadens a best-efforts service model, enabling the management of end-to-end packet delays. IntServ reserves resources on a per-flow basis and requires Resource Reservation Protocol (RSVP) as a setup protocol that guarantees bandwidth and a limit to packet delay using router-to-router signaling schemes. Participating protocols include the Real-time Transport Protocol (RTP), which is the transport protocol in which receivers sequence information through packet headers. Real-Time Control Protocol (RTCP) gives feedback of status from senders to receivers. RTP and RTCP are ITU standards under H.225. Real-Time Streaming Protocol (RTSP) runs on top of IP Multicast, UDP, RTP, and RTCP. RSVP supports both IPv4 and IPv6, and is important to scalability and security; it provides a way to ensure that policy-based decisions are followed.

DiffServ is a follow-on QoS approach to IntServ. DiffServ is based on a CoS model; it uses a specified set of building blocks from which many services can be built. DiffServ implements a prioritization scheme that differentiates traffic using certain bits in each packet (IPv4 type-of-service [ToS] byte or IPv6 traffic class byte) that designate how a packet is to be forwarded at each network node. The move to IPv6 is advantageous because the ToS field has limited functionality and there are various interpretations. DiffServ uses traffic classification to prioritize the allocation of resources. The IETF DiffServ draft specifies a management information base, which would allow for DiffServ products to be managed by Simple Network Management Protocol (SNMP).

Multi-Protocol Label Switching (MPLS) is an evolving protocol with standards originally out of the IETF that designates static IP paths. It provides for the traffic engineering capability essential to QoS control and network optimization, and it forms a basis for VPNs. Unlike IP, MPLS can direct traffic through different paths to overcome IP congested route conditions that adversely affect network availability. To steer IPv4 or IPv6 packets over a particular route through the Internet, MPLS adds a label to the packet. To enable routers to direct classes of traffic, MPLS also labels the type of traffic, path, and destination information. A packet on an MPLS network is transmitted through a web of MPLS-enabled routers or ATM switches called label-switching routers (LSRs). At each hop in the MPLS network, the LSR uses the local label to index a forwarding table, which designates a new label to each packet, and sends the packet to an output port. Routes can be defined manually or via RSVP-TE (RSVP with traffic engineering extensions) or MPLS Label Distribution Protocol (LDP). MPLS supports the desired qualities of circuit-switching technology such as bandwidth reservation and delay variation as well as a best-efforts hop-by-hop routing. Using MPLS, service providers can build

VPNs with the benefits of both ATM-like QoS and the flexibility of IP. The potential capabilities of the encapsulating label-based protocol continues to grow; however, there are a number of issues between the IETF and MPLS Forum that need full resolution, such as the transfer of ToS markings from IP headers to MPLS labels and standard LSR interpretation when using MPLS with DiffServ.

The management of voice availability and quality issues is performed through policy-based networking. Information about individual users and groups is associated with network services or classes of service. Network protocols, methods, and directories used to enable the granular time-sensitive requirements of policy-based QoS are Common Open Policy Services (COPS), Directory Enabled Networking (DEN), and Lightweight Directory Access Protocol (LDAP).

VOI Security

Threats to voice communication systems increase given the move to the inherently open Internet. Voice security policies, procedures, and methods discussed previously reflect the legacy closed voice network architecture; they are not adequate for IP telephony networks, which are essentially wide open and require little or no authentication to gain access. New-generation networks require protection from attacks across the legacy voice network, wireless network, WAN, and LAN. Should invalid signaling occur on the legacy network, trunk groups could be taken out of service, calls placed to invalid destinations, resources locked up without proper release, and switches directed to incorrectly reduce the flow of calls. As new IP telephony security standards and vendor functions continue to evolve, service providers and enterprises can make use of voice-oriented firewalls as well as many of the same data security techniques to increase voice security.

Inherent characteristics of Voice-over-IP protocols and multimedia security schemes are in conflict with many current methods used by firewalls or network address translation (NAT). Although no official standards exist, multiple security techniques are available to operate within firewall and NAT constraints. These methods typically use some form of dynamic mediation of ports and addresses whereby each scheme has certain advantages given the configuration and overall requirements of the network. Security standards, issues, and solutions continue to evolve as security extensions to signaling protocols, related standards, and products likewise evolve and proliferate.

SIP, H.323, MGCP, and Megaco/H.248 signaling protocols use TCP as well as UDP for call setup and transport. Transport addresses are embedded in the protocol messages, resulting in a conflict of interest. Secure firewall rules specify static ports for desirable data block H.323 because the signaling protocol uses dynamically allocated port numbers. Related issues trouble NAT devices. An SIP user on an internal network behind a NAT sends an INVITE message to another user outside the network. The outside user extracts the FROM address from the INVITE message and sends a 200(Ok) response back. Because the INVITE message comes from behind the NAT, the FROM address is not correct. The call never connects because the 200 response message does not succeed.

H.323 and SIP security solution examples available today are described. H.323, an established ITU standard designed to handle real-time voice and videoconferencing, has been used successfully for VoIP. The standard is based on the IETF Real-Time Protocol (RTP) and Real-Time Control Protocol (RTCP) in addition to other protocols for call signaling and data and audiovisual communications. This standard is applied to peer-to-peer applications where the intelligence is distributed throughout the network. The network can be partitioned into zones, and each zone is under the control of an intelligent gatekeeper. One voice firewall solution in an H.323 environment makes use of the mediating element that intervenes in the logical process of call setup and tear-down, handles billing capabilities, and provides high-level policy control. In this solution, the mediating element is the H323 gatekeeper; it is call-state aware and trusted to make network-wide policy decisions. The data ports of the voice firewall device connect to the output of the H.323 gateway device. The gatekeeper incorporates firewall management capabilities via API calls; it controls connections to the voice firewall device that opens dynamic "pinholes," which permit the relevant traffic through the voice firewall. Voice firewalls are configured with required pinholes and policy for the domain, and no other traffic can flow through the firewall. For each call setup, additional pinholes are configured dynamically to permit the precise traffic required to carry that call; and no other traffic is allowed. The voice firewall simplicity using stateless packet filtering can perform faster at lower costs compared to a traditional application firewall, with claims of 100 calls per second to drill and seal pinholes and a chassis that supports hundreds of simultaneous calls with less than one millisecond of latency

SIP, an increasingly popular approach, operates at the application layer of the OSI model and is based on IETF RFC 2543. SIP is a peer-to-peer signaling protocol controlling the creation, modification, and termination of sessions with one or more participants. SIP establishes a temporary call to the server, which performs required, enhanced service logic. The SIP stack consists of SIP using Session Description Protocol (SDP), RTCP, and RTP. Recent announcements — a Windows XP® SIP telephony client and designation of SIP as the signaling and call control standard for IP 3G mobile networks — have accelerated service providers' deployments of SIP infrastructures.

Comprehensive firewall and NAT security solutions for SIP service providers include a combination of technologies, including an edge proxy, a firewall control proxy, and a media-enabled firewall. An edge proxy acts as a guard, serving the incoming and outgoing SIP signaling traffic. It performs authentication and authorization of services through transport layer security (TLS) and hides the downstream proxies from the outside network. The edge proxy forwards calls from trusted peers to the next internal hop. The firewall control proxy works in conjunction with the edge proxy and firewall. For each authorized media stream, it dynamically opens and closes pinhole pairs in the firewall. The firewall control proxy also operates closely with the firewall to perform NAT and remotely manages firewall policy and message routing. Dynamic control and failover functions of these firewall control proxies provide the additional required reliability in the service provider network. The media-enabled firewall is a transparent, non-addressable VoIP firewall that does not allow access to the internal network except from the edge proxy. Carrier-class high-performance firewalls can limit entering traffic to the edge proxy and require a secure TLS connection for only media traffic for authorized calls.

Enterprise IP Telephony Security

Threats associated with conversation eavesdropping, call recording and modification, and voicemail forwarding or broadcasting are greater in a VoIP network, where voice files are stored on servers and control and media flows reside on the open network. Threats related to fraud increase given the availability of control information on the network such as billing and call routing. Given the minimal authentication functionality of voice systems, threats related to rogue devices or users increase and can also make it more difficult to track the hacker of a compromised system if an attack is initiated in a phone system.

Protection needs to be provided against denial-of-service (DoS) conditions, malicious software to perform a remote boot, TCP SYN flooding, ping of death, UDP fragment flooding, and ICMP flooding attacks. Control and data flows are prone to eavesdropping and interception given the use of packet sniffers and tools to capture and reassemble generally unencrypted voice streams. Viruses and Trojan horse attacks are possible against PC-based phones that connect to the voice network. Other attacks include a caller identity attack on the IP phone system to gain access as a legitimate user or administrator. Attacks to user registration on the gatekeeper could result in redirected calls. IP spoofing attacks using trusted IP addresses could fool the network that a hacker conversation is that of a trusted computer such as the IP-PBX, resulting in a UDP flood of the voice network.

Although attack mitigation is a primary consideration in VoIP designs, issues of QoS, reliability, performance, scalability, authentication of users and devices, availability, and management are crucial to security. VoIP security requirements are different from data security requirements for several reasons. VoIP applications are under no-downtime, high-availability requirements; operate in a badly behaved manner using dynamically negotiated ports; and are subject to extremely sensitive performance needs. VoIP security solutions are comprehensive; they include signaling protocols, operating systems, administration interface; and they need to fit into existing security environments consisting of firewalls, VPNs, and access servers. Security policies must be in place because they form a basis for an organization's acceptance of benefits and risks associated with VoIP. Certain signaling protocol security recommendations exist and are evolving. For example, the ITU-T H.235 Recommendation under the umbrella of H.323 provides for authentication, privacy, and integrity within the current H-Series protocol framework. Vendor products, however, do not necessarily fully implement such protection. In the absence of widely adopted standards, today's efforts rely on securing the surrounding network and its components.

Enterprise VoIP security design makes use of segmentation and the switched infrastructure for QoS, scalability, manageability, and security. Today, layer 3 segmentation of IP voice from the traditional IP data network aids in the mitigation of attacks. A combination of virtual LANs (VLANs), access control, and stateful firewall provides for voice and data segmentation at the network access layer. Data devices on a separate segment from the voice segment cannot instigate call monitoring, and the use of a switched infrastructure baffles devices on the same segment sufficiently to prevent call monitoring and maintain confidentiality. Not all IP phones with

data ports, however, support other than basic layer 2 connectivity that acts as a hub, combining the data and voice segments. Enhanced layer 2 support is required in the IP phone for VLAN technology (like 802.1q), which is one aspect needed to perform network segmentation today. The use of PC-based IP phones provides an avenue for attacks such as a UDP flood DoS attack on the voice segment making a stateful firewall that brokers the data–voice interaction required. PC-based IP phones are more susceptible to attacks than closed custom operating system IP phones because they are open and sit within the data network that is prone to network attacks such as worms or viruses. Controlling access between the data and voice segments uses a strategically located stateful firewall. The voice firewall provides host-based DoS protection against connection starvation and fragmentation attacks, dynamic per-port granular access through the firewall, spoof mitigation, and general filtering. Typical authorized connections such as voicemail connections in the data segment, call establishment, voice browsing via the voice segment proxy server, IP phone configuration setting, and voice proxy server data resource access generally use well-known TCP ports or a combination of well-known TCP ports and UDP. The VoIP firewall handles known TCP traditionally and opens port-level granular access for UDP between segments. If higher-risk PC-based IP phones are utilized, it is possible to implement a private address space for IP telephony devices as provided by RFC 1918. Separate address spaces reduce potential traffic communication outside the network and keep hackers from being able to scan a properly configured voice segment for vulnerabilities.

The main mechanism for device authentication of IP phones is via the MAC address. Assuming automatic configuration has been disabled, an IP phone that tries to download a network configuration from an IP-PBX needs to exhibit a MAC address known to the IP-PBX to proceed with the configuration process. This precludes the insertion of a rogue phone into the network and subsequent call placement unless a MAC address is spoofed. User log-on is supported on some IP phones for device setup as well as identification of the user to the IP-PBX, although this could be inconvenient in certain environments. To prevent rogue device attacks, employ traditional best practice regarding locking down switched ports, segments, and services holds. In an IP telephony environment, several additional methods could be deployed to further guard against such attacks. Assignment of static IP addresses to known MAC addresses versus Dynamic Host Configuration Protocol (DHCP) could be used so that, if an unknown device is plugged into the network, it does not receive an address. Also, assuming segmentation, separate voice and data DHCP servers means that a DoS attack on the DHCP data segment server has little chance of affecting the voice segment. The *temporary use only when needed* guideline should be implemented for the commonly available automatic phone registration feature that boot-straps an unknown phone with a temporary configuration. A MAC address monitoring tool on the voice network that tracks changes in MAC to IP address pairings could be helpful, given that voice MAC addresses are fairly static. Assuming network segmentation, filtering could be used to limit devices from unknown segments as well as keeping unknown devices within the segment from connecting to the IP-PBX.

Voice servers are prone to similar attacks as data servers and therefore could require tools such as an intrusion detection system (IDS) to alarm, log, and perhaps react to attack signatures found in the voice network. There are no voice control protocol attack signatures today, but an IDS could be used for UDP DoS attack and HTTP exploits that apply to a voice network. Protection of servers also includes best practices, such as disabling unnecessary services, applying OS patches, turning off unused voice features, and limiting the number of applications running on the server. Traditional best practices should be followed for the variety of voice server management techniques, such as HTTP, SSL, and SNMP.

Wireless Convergence

Wireless carriers look to next-generation networks to cost-effectively accommodate increased traffic loads and to form a basis for a pure packet network as they gradually move toward 3G networks. The MSCs in a circuit-switched wireless network as described earlier in this chapter interconnect in a meshed architecture that lacks easy scaling or cost-effective expansion; a common packet infrastructure to interconnect MSCs could overcome limitations and aid in the move to 3G networks. In this architecture, the common packet framework uses packet tandems consisting of centralized MGCs or softswitches that control distributed MGs deployed and located with MSCs. TDM trunks from each MSC are terminated on an MG that performs IP or ATM conversion under the management of the softswitch. Because point-to-point connections no longer exist between MSCs, a less complicated network emerges that requires less bandwidth. Now MSCs can be added to the network with one softswitch connection instead of multiple MSC connections. Using media gateways negates the need to upgrade software at each MSC to deploy next-generation services, and it offloads precious switching center

resources. Centrally located softswitches with gateway intelligence can perform lookups and route calls directly to the serving MSC versus the extensive routing required among MSCs or gateway MSCs to perform lookups at the home location register. With the progression of this and other IP-centric models, crucial registration, authentication, and equipment network databases need to be protected.

Evolving new-generation services require real-time metering and integration of session management with the transfer data. Service providers look to support secure virtual private networks (VPNs) between subscribers and providers of content, services, and applications. While the emphasis of 2.5G and 3G mobile networks is on the delivery of data and new multimedia applications, current voice services must be sustained and new integrated voice capabilities exploited. Regardless of specific implementations, it is clear that voice networks and systems will continue to change along with new-generation networks.

References

Telecommunications Essentials, Addison-Wesley, 2002, Lillian Goleniewski.

Voice over IP Fundamentals, Cisco Press, 2002, Jonathan Davidson and James Peters.

SS7 Tutorial, Network History, 2001, SS8 Networks.

Securing future IP-based phone networks, *ISSA Password*, Sept/Oct. 2001, David K. Dumas, CISSP.

SAFE: IP Telephony Security in Depth, Cisco Press, 2002, Jason Halpern.

Security Analysis of IP-Telephony Scenarios, Darmstadt University of Technology, KOM — Industrial Process and System Communications, 2001, Utz Roedig.

Deploying a Dynamic Voice over IP Firewall with IP Telephony Applications, Aravox Technologies, 2001, Andrew Molitor.

Building a strong foundation for SIP-based networks, *Internet Telephony*, February 2002, Erik Giesa and Matt Lazaro.

Traversal of IP Voice and Video Data through Firewalls and NATS, RADVision, 2001.

PBX Vulnerability Analysis, Finding Holes in Your PBX Before Someone Else Does, U.S. Department of Commerce, National Institute of Standards and Technology, Special Publication 800-24.

The Start of the Digital Revolution: SIGSALY Secure Digital Voice Communications in World War II, The National Security Agency (NSA), J.V. Boone and R.R. Peterson.

Wireless carriers address network evolution with packet technology, *Internet Telephony,* November 2001, Ravi Ravishankar.

Glossary of Terms

AIN (Advanced Intelligent Network) — The second generation of intelligent networks, which was pioneered by Bellcore and later spun off as Telcordia. A common service-independent network architecture geared to quickly produce customizable telecommunication services.

ATM (Asynchronous Transfer Mode) — A cell-based international packet-switching standard where each packet has a uniform cell size of 53 bytes. It is a high-bandwidth, fast packet-switching and multiplexing method that enables end-to-end communication of multimedia traffic. ATM is an architected quality-of-service solution that facilitates multi-service and multi-rate connections using a high-capacity, low-latency switching method.

CCITT (Comité Consultatif International de Télephonie et de Télégraphie) — Advisory committee to the ITU, now known as the ITU-T, that influences engineers, manufacturers, and administrators.

CoS (Class-of-Service) — Categories of subscribers or traffic corresponding to priority levels that form the basis for network resource allocation.

CPE (Customer Premise Equipment) — Equipment owned and managed by the customer and located on the customer premise.

DTMF (Dual-Tone Multi-Frequency Signaling) — A signaling technique for push-button telephone sets in which a matrix combination of two frequencies, each from a set of four, is used to send numerical address information. The two sets of four frequencies are (1) 697, 770, 852, and 941 Hz; and (2) 1209, 1336, 1477, and 1633 Hz.

IP (Internet Protocol) — A protocol that specifies data format and performs routing functions and path selection through a TCP/IP network. These functions provide techniques for handling unreliable data and specifying the way network nodes process data, how to perform error processing, and when to throw out unreliable data.

IN (Intelligent Network) — An advanced services architecture for telecommunications networks.

ITU-T (International Telecommunication Union) — A telecommunications advisory committee to the ITU that influences engineers, manufacturers, and administrators.
MPLS (Multi-Protocol Label Switching) — An IETF effort designed to simplify and improve IP packet exchange and provide network operators with a flexible way to engineer traffic during link failures and congestion. MPLS integrates information about network links (layer 2) such as bandwidth, latency, and utilization with the IP (layer 3) into one system.
NIST (National Institute of Standards and Technology) — A U.S. national group that was referred to as the National Bureau of Standards prior to 1988.
PBX (Private Branch Exchange) — A telephone switch residing at the customer location that sets up and manages voice-grade circuits between telephone users and the switched telephone network. Customer premise switching is usually performed by the PBX as well as a number of additional enhanced features, such as least-cost routing and call-detail recording.
PSTN (Public Switched Telephone Network) — The entire legacy public telephone network, which includes telephones, local and interexchange trunks, communication equipment, and exchanges.
QoS (Quality-of-Service) — A network service methodology where network applications specify their requirements to the network prior to transmission, either implicitly by the application or explicitly by the network manager.
RSVP (Reservation Resource Protocol) — An Internet protocol that enables QoS; an application can reserve resources along a path from source to destination. RSVP-enabled routers then schedule and prioritize packets in support of specified levels of QoS.
RTP (Real-Time Transport Protocol) — A protocol that transmits real-time data on the Internet. Sending and receiving applications use RTP mechanisms to support streaming data such as audio and video.
RTSP (Real-Time Streaming Protocol) — A protocol that runs on top of IP multicasting, UDP, RTP, and RTCP.
SCP (Service Control Point) — A centralized node that holds service logic for call management.
SSP (Service-Switching Point) — An origination or termination call switch.
STP (Service Transfer Point) — A switch that translates SS7 messages and routes them to the appropriate network nodes and databases.
SS7 (Signaling System 7) — An ITU-defined common signaling protocol that offloads PSTN data traffic congestion onto a wireless or wireline digital broadband network. SS7 signaling can occur between any SS7 node, and not only between switches that are immediately connected to one another.

PREVENTING DNS ATTACKS

Mark Bell

INSIDE

How DNS Works, Opportunities — Abusing the DNS Trust, Poisoning the Cache, Averting DNS Attacks, Encryption

INTRODUCTION

Of all the Internet services, the Domain Name Service (DNS) is the most used, and perhaps the most vulnerable. Without DNS, users would have to know the "dotted quad" address of every resource that they use on the Internet; humans have a poor memory for numbers, but can recall the names of Web sites without much difficulty. In many cases, a company's Web address can be derived by adding "www" and "com" to each side of the company name; for example, www.microsoft.com. All of this depends on DNS. Imagine having to enter 199.29.24.3 instead of www.crcpress.com into a browser, and having to somehow remember the address of all of the other Web sites in that way. If DNS is essential now, when Internet addresses are only 32 bits long (IPv4), imagine the problem when IPv6 is widely adopted and many addresses increase to 128 bits.

DNS was designed by Paul Mockapetris of USC in 1984, and was described in RFCs 882 and 883. At that time, little thought was given to security; the service was designed to be efficient and reliable so that it could be deployed and used with the minimum of effort. It fulfilled all expectations and has been in continuous use since its inception, with remarkably few problems. If the lack of concern for security seems surprising, it should be remembered that at that time, the Internet community was much smaller, and use of the Internet was restricted to universities, government departments, the military, etc. — and widespread use of the Internet was not envisaged. TCP/IP was supposed to be the "temporary" network, a stopgap suite of protocols that would be replaced in a few years by the OSI suite — what was the point of spending much time on something that would be obsolete in a few years? The fiction that OSI

PAYOFF IDEA

Although still a rare occurrence, DNS attacks are increasing. Data encryption makes DNS attacks pointless, and also protects many other services on the Internet.

Auerbach Publications

would replace TCP/IP continued until the late 1980s, and was responsible for many decisions that seem poor in retrospect.

How DNS Works

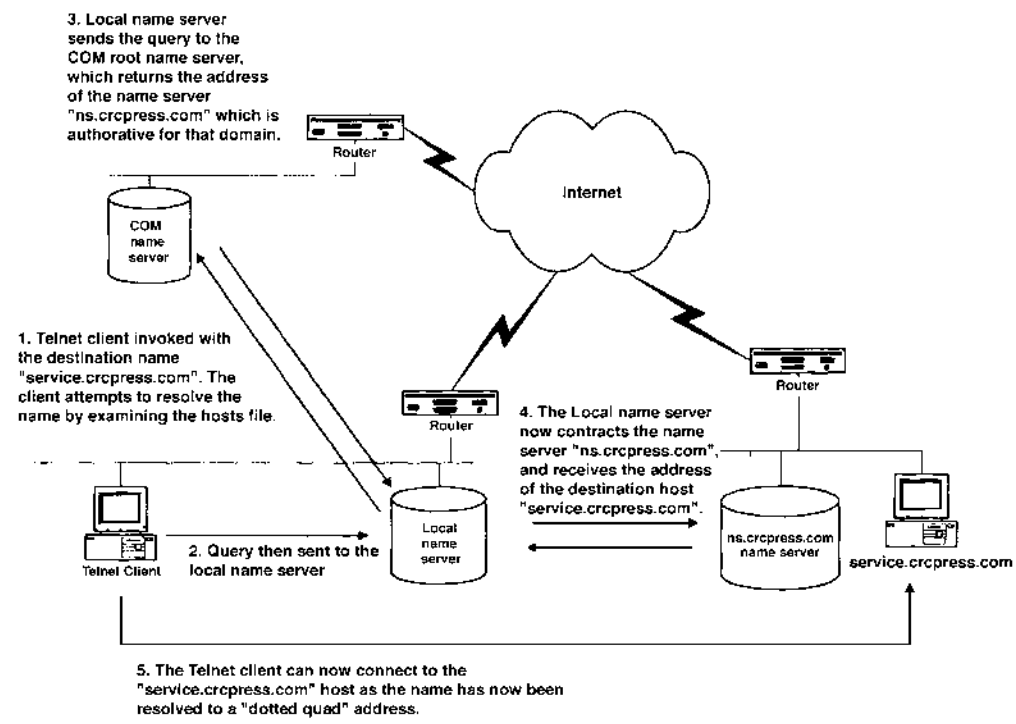
In order to understand the vulnerability of DNS, one needs to know how it functions — at least in enough detail to follow the flow of information. Nearly all TCP/IP applications and services are based on the client/server model; in Unix, telnet is the client and telnetd is the server. With telnet, a user enters "telnet service.crcpress.com," invoking the telnet client and passing the name of the host as a parameter. The client passes the parameter — in this case, service.crcpress.com — to a resolver routine compiled into the client code, and the resolver then attempts to resolve the name into a dotted quad address. The resolver first looks in a file on the host machine — "/etc/hosts" on Unix — to see if there is an entry matching the destination host name. If there is no entry, the resolver then sends a query to the local name server, whose address must be known to the client host. (See Exhibit 1.)

The name server will look in his cache to see if there is an entry for the host. There are several ways that the name may have been placed in the cache:

1. If the destination host is on the local network, the name and address will have been entered into a table by the DNS administrator. This table is kept on the hard disk, and is reloaded every time the name server boots up. These are the addresses for which this server is considered to be "authoritative." The name server is the "primary" server for these names.
2. The name server also has a list of servers entered into this same table, which are authoritative for other sites. When the name server boots, it will download the name/address table from each of these servers and add the contents to its own table. The name server is said to be "secondary" to these other servers.
3. The name server has resolved the same name for another client recently, and the entry is still valid.

If there is no entry in the cache, the name server will pass the query to one of the root servers or a parent server to see if the name exists in the cache on one of these machines. How does the name server know which root server to query? The last part of the name will be the domain in which the name resides, so the address "service.crcpress.com" is in the .com domain. The domains are organized by function — a commercial, for-profit company is in the .com domain, government departments are in the .gov domain, and military installations are in the .mil domain, etc.

EXHIBIT 1 — Resolving a Domain Name



The com server has a table containing the address of all the name servers (at least "top level" servers) in the .com domain; every time a new company joins the .com domain, the address of its main name server is added to the table. The com server will return the address of a name server that can be queried for the address of the destination host, in this case ns.crcpress.com. The client's name server will then query ns.crcpress.com for the address associated with service.crcpress.com, and will be sent the dotted quad address, together with a time for which the name/address pair is guaranteed to be valid (TTL — Time to Live). This address will be added to the cache in the local name server, and kept there until the TTL has expired. In this way, the name server that resolves a query can control how long the query is to be considered valid by other name servers.

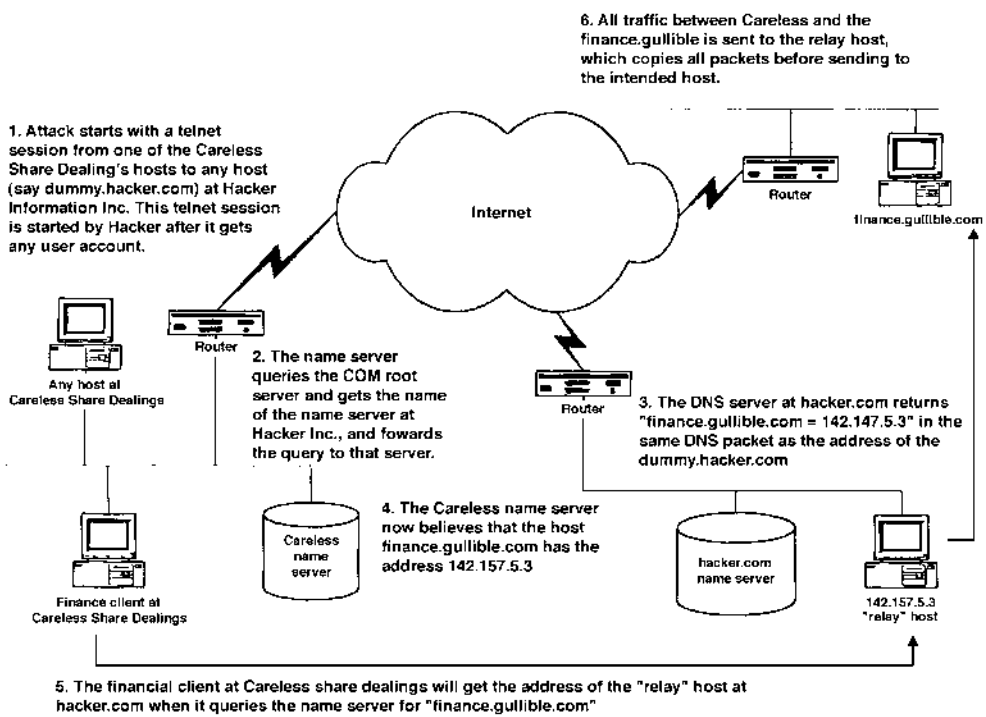
The client host can now open the telnet session with the destination host, but has had to trust the Domain Name Service completely — there is no real way of checking that the resolution is correct, and that the telnet session is being opened with the correct host. This is the problem with the DNS service — it relies on trust, which means that it is open to abuse.

Opportunities — Abusing the DNS Trust

The most obvious damage that can be inflicted on an Internet site is to corrupt the name server's table, or enter the names of invalid addresses with hosts, so that the users would not be able to initiate Internet services without knowing their dotted quad address; this would be relatively harmless, because the users would realize a problem existed, and would be able to flush the name server's cache. This is just mindless vandalism, quickly discovered and speedily resolved. There are other possibilities, however, one of which will be explored in this article.

Imagine a company (Careless Share Dealings, Inc.) carries out financial transactions with another company (Gullible Stocks PLC) via the Internet on a regular basis, and that another person or company (Hacker Information) could benefit if it could read these transactions in a timely manner — perhaps these transactions could be shared dealings between two brokers, with the transactions being automatically recorded every 30 minutes (see Exhibit 2). If the company's name server could be persuaded that the address of the financial server at Gullible had changed, and the new address was relay.hacker.com, then all of the information would be sent to the new address. Now, all that the host at Hacker has to do is to copy the details of every packet to a file on the hard disk, and relay the packet on to its proper destination, Gullible Stocks. Assuming that the share dealing software used TCP, two sessions would be set up: one between Careless and Hacker and another between Hacker and Gullible. Note that the traffic from Gullible back to Careless would follow the same path; DNS is used by the original client (Careless) because Careless does not know the

EXHIBIT 2 — Poisoning the DNS Cache



address of the server (Gullible). When the server receives a request to start a session, he gets the address of the client in the TCP/IP packet, so he has no need to use DNS. He is trusting that the client is who he says he is — in this case, a bad assumption. Even if the service between Careless and Gullible had passwords that changed every 30 seconds, this would not prevent the attack from taking place because Hacker does not need to know the password — he is passing on valid information from Careless.

Poisoning the Cache

The most obvious way to fool the Careless name server into believing that the address of the Gullible Finance server had changed would be to break into the name server at Gullible and place the new address in the table. This would risk discovery, however, because all of the other hosts at Gullible would also get the incorrect address; if Gullible had a firewall, there would be logs showing that traffic between two hosts on the Gullible site is being diverted to another host on the Internet and relayed back, and the game would be over.

Another way that the attack can succeed for the longer term would be to penetrate the name server at Careless. Because the Careless name server does not have the name/address pair of the Gullible finance server in a permanent table, the Careless name server would have to serve as the primary server for the Gullible domain; this could be done without affecting the hosts at Gullible — they would still be sending their queries to the Gullible name server. However, the changes to the Careless name server would soon be noticed by the DNS administrator, and again the game would be over.

The best strategy is to use a name server somewhere else on the Internet and exploit the biggest weakness of name servers — the complete trust they must have in any other name servers in order to be efficient. When a name server sends a query, it not only accepts the answer to the original query, but will also accept answers to queries it has not made, and will cache those answers without attempting authentication. This allows a name server to send a list of recent updates any time that it answers a query, and helps to reduce the name resolution traffic on the Internet. Now, all that the attacker needs to do is to make a modification to the server at Hacker Information and then trigger a query to Hacker from Careless. This can be done in several ways; some examples follow:

1. Forge mail on the Careless mail server and address the mail to a user at Hacker Information. This would cause the mail server to query the name server at Careless for the Hacker mail server and, of course, the name server would eventually query the name server at Hacker after contacting the root name server.

2. Break into any machine on the Careless site, and telnet to any host on the Hacker site. This would trigger a DNS query, with the same results.
3. Alter a URL on a Web server at the Careless site or any Web server the users at Careless will visit. This will also trigger a name query.

This is almost a perfect "man-in-the-middle" attack, with very little chance of tracking down the attacker. Even if one could trace the machine running the relay software, it is probable that it belongs to an innocent third party who is unaware that his machine has been compromised. The hacker will be sending the captured information to an unused account on another innocent's machine, and will log in at leisure to collect it. The attacker could also improve the strategy by using someone else's DNS server to launch the initial poisoning attack

Averting DNS Attacks

One of the popular misconceptions about DNS is that a "double reverse lookup" can be used to authenticate name resolution and prevent this attack. This works as follows:

1. The name is resolved in the usual manner; i.e., DNS.
2. When the client receives the answer, an inverse query is made, where the address is sent to a DNS server and a name is returned.
3. The client then compares the name returned with the name used in the original query, and aborts the transaction if the names do not coincide.

This sounds good, but in practice, this is unworkable for several reasons. What happens if the attacker has not only poisoned the cache with the name/address pair, but has also poisoned the inverse cache? The names would then coincide. If two servers were used — one to resolve the original query and one for the inverse check — there would be no guarantee that both servers had not been poisoned. The biggest problem with the double reverse lookup is that it can only be performed on the primary and secondary servers — the client would have to send the inverse query directly to the name server at Gullible. Name servers do not refer inverse queries they cannot resolve to other name servers; so, if the inverse query is sent to the Careless name server, it would be returned as unresolved.

The best way to prevent DNS attacks is to put the names and addresses of critical hosts in the host's file. The client resolver will look at this file before sending a query to DNS, and this will avert all DNS attacks using any of the host names in the host's file. The problem here is that this

is labor intensive; the whole purpose of the DNS system is to prevent this kind of maintenance burden. In any case, this is only possible for relatively few hosts, although one could cut the amount of duplication by maintaining a central copy of the host's file, and distributing it to other hosts as needed.

Encryption

Many of the security problems on the Internet have a common cause — data is being transported in clear text or in other forms that can easily be read. The real answer to most of these problems is to encrypt all data in transit. What would be the point of the DNS attack, and copying the data from the resulting relay software on the Hacker host, if the data could not be read? The technology to encrypt data has been available for years. SNA has always been able to encrypt data so that it cannot be read in transit, and it is obvious (with hindsight) that this should have been part of the original IP specification. A secure channel can be imposed between Internet sites by the use of encryption routers that will scramble all of the data transmitted between specified sites. Custom applications should be written to encrypt all data in transit; the availability of encryption libraries from RSA and other vendors has simplified development of secure applications.

The real point here is that the choice has to be made between replacing DNS with a more secure service, or rendering DNS attacks pointless by data encryption. The first option only cures the problems with DNS — assuming that a truly secure version of DNS is possible. The second option will render DNS attacks pointless and also protect many other services on the Internet at the same time.

DNS attacks are still a rare occurrence on the Internet. Other attacks, such as the sniffer attack, can be launched more easily and require less knowledge. There are simpler and more direct ways of achieving the same ends — intercepting and copying data in transit — but as precautions are taken against these, simpler methods become more common, and life becomes more difficult for the hacker, one can expect to see an increase in the incidents of DNS attacks.

Mark Bell is an independent consultant with 20 years experience in the computer industry. His work has focused on enterprise networking since 1993. In addition to consulting, he has been teaching courses on TCP/IP and networking for the last 5 years and holds MCSE, MCT, and CNE certifications.

PROTECTING A NETWORK FROM SPOOFING AND DENIAL OF SERVICE ATTACKS

Gilbert Held

INSIDE

Spoofing; Spoofing Methods; Blocking Spoofed Addresses; Anti-spoofing Statements; Ping Attacks; Directed Broadcasts

INTRODUCTION

Along with the evolution of technology, we have witnessed an unfortunate increase in random violence in society. While it is doubtful if the two are related, it is a matter of fact that some violence is directed at computers operated by federal, state, and local governments, universities, and commercial organizations. That violence typically occurs in the form of attempts to break into computers via a remote communications link or to deny other persons the use of computational facilities by transmitting a sequence of bogus requests to the network to which a computer is connected. Because either situation can adversely affect the operational capability of an organization's computational facilities, any steps one can initiate to enhance the security of a network and networked computers may alleviate such attacks.

This article examines several common types of hacker attacks against networks and networked computers. In doing so, it first examines how the attack occurs. Once an appreciation for the method associated with an attack is obtained, attention can focus on techniques that can be used to prevent such attacks. Because the vast majority of routers used for Internet and intranet communications are manufactured by Cisco Systems, examples illustrating the use of the Cisco Systems' Internetwork Opera-

PAYOFF IDEA

Protecting one's network from outside attack has become more critical than ever. This article examines several common types of hacker attacks against networks and illustrates methods to prevent those attacks.

tion System (IOS) will be used when applicable to denote different methods to enhance network security. By examining the information presented in this article, one will note practical methods that can be implemented to add additional protection to an organization's network. Thus, this article serves both as a tutorial concerning spoofing and denial of service attacks, as well as a practical guide to prevent such activities.

SPOOFING

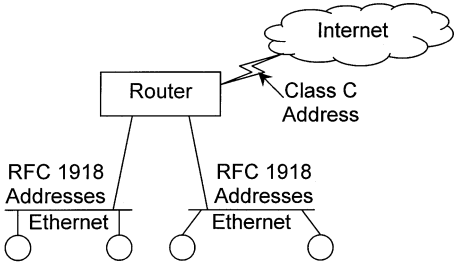
According to Mr. Webster, the term "spoof" means to "deceive or hide." In communications, the term "spoofing" is typically associated with a person attempting to perform an illegal operation. That person, commonly referred to as a hacker, spoofs or hides the source address contained in the packets he or she transmits. The rationale for hiding the hacker's source address is to make it difficult, if not impossible, for the true source of the attack to be identified. Because spoofing is employed by most hackers that spend the time to develop different types of network attacks, one should first examine how spoofing occurs. This is followed by a discussion of methods one can employ to prevent certain types of spoofed packets from flowing into a network.

SPOOFING METHODS

There are several methods hackers can use to spoof their source addresses. The easiest method is to configure their protocol stack with a bogus address. In a TCP/IP environment, this can be easily accomplished by a person coding a bogus IP address in the network address configuration screen displayed by the operating system supported by their computer. Because only the destination address is normally checked by networking devices (such as routers and gateways), it is relatively easy to hide one's identity by configuring a bogus source IP address in one's protocol stack.

When configuring a bogus IP address, hackers, for some unknown reason, commonly use either an address associated with the attacked network or with an RFC 1918 address. Concerning the latter, RFC 1918 defines three blocks of IP addresses for use on private IP networks. Because the use of RFC 1918 addresses on networks directly connected to the Internet would result in duplicated IP addresses, they are barred from direct use on the Internet. Instead, they are commonly used by organizations that have more computers than assigned IP addresses. For example, assume an organization originally requested one Class C IP address from their Internet Service Provider (ISP). A Class C IP address is capable of supporting up to 254 hosts, because host addresses 0 and 255 cannot be used. Now suppose the organization grew and required more than 254 workstations to be connected to the Internet. While the organization could request another Class C network address from its ISP, such addresses are becoming difficult to obtain and the organization might have

EXHIBIT 1 — Using RFC 1918 Addresses and Network Address Translation to Support Internet Connectivity for Many Workstations



to wait weeks or months to obtain the requested address. As an alternative, the organization could use RFC 1918 addresses and use its router to perform network address translation as illustrated in Exhibit 1.

In examining Exhibit 1, note that two Ethernet segments are shown behind the router. Each segment could represent an individual Class C network using RFC 1918 addresses. The router would translate those RFC 1918 addresses to either a group of pooled Class C addresses or one Class C address, with the method of translation based on the manner in which the router's translation facility was configured.

If a pooled Class C address is used, the number of simultaneous sessions is limited to 254. If one Class C address is used, the router uses TCP and UDP port numbers to translate from RFC 1918 addresses to a common Class C address, with port numbers used to keep track of each address translation. Because there are thousands of unused port numbers, this method provides a greater translation capability as it limits or avoids potential contention between users behind the router requesting access to the Internet and available IP addresses.

Perhaps because RFC 1918 addresses are popularly used by many organizations, yet hidden by network address translation, they are commonly used as a source address when a hacker configures his or her protocol stack. Exhibit 2 lists the three address blocks reserved for private IP networks under RFC 1918.

EXHIBIT 2 — RFC 1918 Address Blocks

10.0.0.0	10.255.255.255
172.16.0.0	172.31.255.255
192.168.0.0	192.168.255.255

The use of an RFC 1918 address or the selection of an address from the target network results in a static source address. While this is by far the most common method of IP address spoofing, on occasion a sophisticated hacker will write a program that randomly generates source addresses. As will be noted shortly, only when those randomly generated source addresses represent an address on the target network or an RFC 1918 address are they relatively easy to block.

BLOCKING SPOOFED ADDRESSES

Because a router represents the point of entry into a network, it also represents one's first line of defense. Most routers support packet filtering, allowing the network administrator to configure the router to either permit or deny the flow of packets, based on the contents of one or more fields in a packet.

Cisco routers use access lists as a mechanism to perform packet filtering. A Cisco router supports two basic types of access lists: standard and extended. A Cisco standard IP access list performs filtering based on the source address in each packet. The format of a standard IP access list statement is shown below:

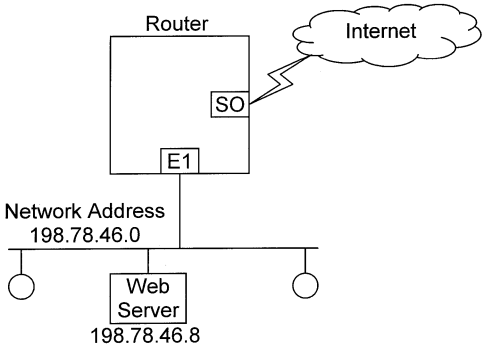
access-list list# [permit/deny][ip address][mask][log]

The list# is a number between 1 and 99 and identifies the access list as a standard access list. Each access list statement contains either the keyword "permit" or "deny," which results in the packet with the indicated IP address either being permitted to flow through a router or sent to the great bit bucket in the sky. The mask represents a wildcard mask that functions in a reverse manner to a subnet mask. That is, a binary 0 is used to represent a "don't-care" condition. Note this is the opposite of the use of binary 0s and 1s in a subnet mask. In fact, the wildcard mask used by a Cisco router is the inverse of a subnet mask, and each position in the wildcard mask can be obtained by subtracting the value of the subnet mask for that position from 255.

The keyword "log" is optional and when included results in each match against a packet being displayed on the router's console. Logging can facilitate the development of access lists as well as serve as a mechanism to display activity that the access list was constructed to permit or deny. Thus, on occasion, it can be used to see if one's router is under attack or if suspicious activity is occurring.

In a Cisco router environment, access lists are applied to an interface in the inbound or outbound direction. To do so, one would use an interface command and an ip access-group command. Because spoofed IP addresses represent packets with bogus source addresses, one can use either standard or extended access lists to block such packets from enter-

EXHIBIT 3 — Connecting an Ethernet Segment to the Ethernet



ing a network. Since extended access lists will be discussed and described later in this article, we first illustrate the use of a standard access list to block packets with spoofed IP addresses. In doing so, assume an organization uses a Cisco router as illustrated in Exhibit 3 to connect a single Ethernet segment with a Web server and conventional workstations to the Internet. In examining Exhibit 3, note that it is assumed that the network address is 198.78.46.0 and the server has the IP address of 198.78.46.8.

ANTI-SPOOFING STATEMENTS

Because statements in a Cisco access list are operated upon in their sequence, top down, one should place anti-spoofing statements at the beginning of the access list. Since one wants to protect the network from persons attempting to remotely access the network via the Internet, one would apply the anti-spoofing statements in the access list to be created to the serial interface of the router. The access list will be applied in the inbound direction since one wants to examine packets flowing from the Internet toward the organization's Ethernet segment for bogus IP addresses.

The example shown in Exhibit 4 illustrates the configuration and application of a Cisco standard IP access list to effect anti-spoofing operations. In this example, four deny statements at the beginning of the access list preclude packets with a source address of any possible host on the organization's network, as well as any RFC 1918 address from flowing through the router.

The first deny statement checks each packet for a source address associated with the 198.78.46.0 network. Note that the wildcard mask of 0.0.0.255 results in the router matching the first three positions of each dotted decimal address but not caring about the fourth position. Thus, any

EXHIBIT 4 — An Access List that Performs Anti-Spoofing Operations

```
interface serial 0
ip access-group1 in
!
ip access-list1 deny 198.78.46.0 0.0.0.255
ip access-list1 deny 10.0.0.0 0.255.255.255
ip access-list1 deny 172.16.0.0.0 0.31.255.255
ip access-list1 deny 192.168.0.0. 0.0.255.255   ip access-list1 permit 0.0.0.0 255.255.255.255
```

packet with a source address associated with the internal network will be tossed into the great bit bucket in the sky. The next three deny statements in effect bar packets that use any RFC 1918 address as their source address. Because an access list denies all packets unless explicitly permitted, the access list just created would support anti-spoofing but disallow all other packets. Thus, a permit statement was added at the end of the access list. That statement uses a wildcard mask of 255.255.255.255, which in effect is a complete don't-care and represents the keyword "any" that one can use synonymously in a Cisco access list to represent an address and mask value of 0.0.0.0 255.255.255.255. Since statements are evaluated in their order in the list, if a packet does not have a source address on the 198.78.46.0 network or an RFC 1918 address, it is permitted to flow through the router. Also note that the command "interface serial 0" defines serial port 0 as the interface the access list will be applied to, while the command "ip access-group 1 in" defines that access-list1 will be applied to the serial 0 port in the inbound direction.

Now that there is an appreciation for how one can prevent packets with spoofed IP addresses from flowing into a network, attention can be turned to the manner by which one can prevent several types of denial of service attacks.

PING ATTACKS

One of the more common methods of creating a denial of service attack occurs when a person in a computer laboratory goes from workstation to workstation and configures each computer to ping a target using the -t option supported by most versions of Windows. The -t option results in the computer continuously pinging the target IP address. While one or a few workstations continuously pinging a Web server will only slightly impact the performance of the server, setting 50 or 100 or more workstations to continuously ping a server can result in the server spending most of its time responding to pings instead of user queries.

One method that can be used to prevent a ping attack is to block pings from entering the network. If the organization uses a Cisco router, one can block pings through the use of an extended IP access list. The format of a Cisco extended IP access list is shown below.

access-list list# [permit/deny] protocol [source address]
[source-wildcard][source port][destination address]
[destination-wildcard][destination port][options]

Unlike a standard IP access list that is limited to filtering based on the source address in a packet, an extended access list permits filtering based on several fields. Those fields include the type of protocol transported in the packet, its source address and destination address, and upper layer protocol information. Concerning the latter, one can use extended IP access lists to filter packets based on the value in their source and destination port fields. In addition to the preceding, an extended access list supports a range of options (such as "log"), as well as other keywords to enable specific types of access-list functions.

Returning to the problem at hand, how can one bar pings into an organization's network? The answer to this question is to use an extended IP access list. To do so, one would configure an access list statement that uses the ICMP protocol, since pings are transported by ICMP echo-request packets. The following Cisco extended IP access list statement could be used to block pings:

access-list 101 deny icmp any any echo-request

In the above extended IP access list statement, one will block echo-requests (pings) from any source address flowing to any destination address. Because one would apply the access list to the serial interface in the inbound direction, it would block pings from any address on the Internet destined to any address on the organization's Ethernet network. Knowing how to block pings, one can focus attention on another type of hacker denial of service attack — as directed broadcasts.

DIRECTED BROADCASTS

Refocusing on Exhibit 3, one notes that the network address of 198.78.46.0 represents a Class C network. A Class C network uses 3 bytes of its 4-byte address for the network address and 1 byte for the host address. Although an 8-bit byte can support 256 distinct numbers (0 to 255), an address of 0 is used to represent "this network," while an address of 255 is used to represent a "broadcast" address. Thus, a maximum of 254 hosts can be connected to a Class C network.

A directed broadcast occurs when a user on one network addresses a packet to the broadcast address of another network. In this example, that would be accomplished by sending a packet to the destination address of 198.78.46.255. The arrival of this packet results in the router converting the layer 3 packet into a layer 2 Ethernet frame addressed to everyone on the network as a layer 2 broadcast. This means that each host on

the Ethernet network will respond to the frame and results in a heavy load of traffic flowing on the LAN.

One of the first types of directed broadcast attacks is referred to as a Smurf attack. Under this denial of service attack method, a hacker created a program that transmitted thousands of echo-request packets to the broadcast address of a target network. To provide an even more insidious attack, the hacker spoofed his or her IP address to that of a host on another network that he or she also desired to attack. The result of this directed broadcast attack was to deny service to *two* networks through a *single* attack.

Each host on the target network that is attacked with a directed broadcast responds to each echo-request with an echo-response. Thus, each ping flowing onto the target network can result in up to 254 responses. When multiplied by a continuous sequence of echo-requests flowing to the target network, this will literally flood the target network, denying bandwidth to other applications. Because the source IP address is spoofed, responses are directed to the spoofed address. If the hacker used an IP address of a host on another network that the hacker wishes to harm, the effect of the attack is a secondary attack. The secondary attack results in tens of thousands to millions of echo-responses flowing to the spoofed IP address, clogging the Internet access connection to the secondary network.

Although the original Smurf attack used ICMP echo-requests that could be blocked by an access list constructed to block inbound pings, hackers soon turned to the directed broadcast of other types of packets in an attempt to deny service by using a large amount of network bandwidth. Recognizing the problem of directed broadcasts, Cisco Systems and other router manufacturers soon added the capability to block directed broadcasts on each router interface. On a Cisco router, one would use the following IOS command to turn off the ability for packets containing a directed broadcast address to flow through the router:

```
no ip directed-broadcast
```

SUMMARY

This article focused on methods that can be used to prevent packets containing commonly used spoofed IP addresses from flowing into an organization's network. In addition, it also examined how several popular denial of service attacks operate and methods one can employ to block such attacks.

When considering measures that one can employ to secure a network, it is important to note that there is no such thing as a totally secure network. Unfortunately for society, many hackers are very smart and view the disruption of the operational status of a network as a challenge, pe-

riodically developing new methods to disrupt network activity. To keep up with the latest threats in network security, one should subscribe to security bulletins issued by the Computer Emergency Response Team (CERT) as well as periodically review release notes issued by the manufacturer of your organization's routers and firewalls. Doing so will alert one to new threats, as well as potential methods one can use to alleviate or minimize the effect of such threats.

Gilbert Held is an award-winning author and lecturer. Gil is the author of over 40 books and 400 technical articles focused on computers and data communications. Some of Gil's recent titles include ***Voice over Data Networks Covering IP and Frame Relay,*** 2nd ed., and ***Cisco Security Architecture,*** both published by McGraw-Hill. Gil can be reached via e-mail at 235-8068@mcimail.com.

53

Packet Sniffers: Use and Misuse

Steve A. Rodgers, CISSP

A packet sniffer is a tool used to monitor and capture data traveling over a network. The packet sniffer is similar to a telephone wiretap; but instead of listening to phone conversations, it listens to network packets and conversations between hosts on the network. The word *sniffer* is generically used to describe packet capture tools, similar to the way *crescent wrench* is used to describe an adjustable wrench. The original sniffer was a product created by Network General (now a division of Network Associates called Sniffer Technologies).

Packet sniffers were originally designed to assist network administrators in troubleshooting their networks. Packet sniffers have many other legitimate uses, but they also have an equal number of sinister uses. This chapter discusses some legitimate uses for sniffers, as well as several ways an unauthorized user or hacker might use a sniffer to compromise the security of a network.

How Do Packet Sniffers Work?

The idea of sniffing or packet capturing may seem very high-tech. In reality it is a very simple technology. First, a quick primer on Ethernet. Ethernet operates on a principle called *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD). In essence, the network interface card (NIC) attempts to communicate on the wire (or Ethernet). Because Ethernet is a shared technology, the NIC must wait for an "opening" on the wire before communicating. If no other host is communicating, then the NIC simply sends the packet. If, however, another host is already communicating, the network card will wait for a random, short period of time and then try to retransmit.

Normally, the host is only interested in packets destined for its address; but because Ethernet is a shared technology, all the packet sniffer needs to do is turn the NIC on in promiscuous mode and "listen" to the packets on the wire. The network adapter can capture packets from the data-link layer all the way through the application layer of the OSI model. Once these packets have been captured, they can be summarized in reports or viewed individually. In addition, filters can be set up either before or after a capture session. A filter allows the capturing or displaying of only those protocols defined in the filter.

Ethereal

Several software packages exist for capturing and analyzing packets and network traffic. One of the most popular is Ethereal. This network protocol analyzer can be downloaded from http://www.ethereal.com/ and installed in a matter of minutes. Various operating systems are supported, including Sun Solaris, HP-UX, BSD (several distributions), Linux (several distributions), and Microsoft Windows (95/98/ME, NT4/2000/XP). At the time of this writing, Ethereal was open-source software licensed under the GNU General Public License.

After download and installation, the security practitioner can simply click on "Capture" and then "Start," choose the appropriate network adapter, and then click on "OK." The capture session begins, and a summary window displays statistics about the packets as they are being captured (see Exhibit 53.1).

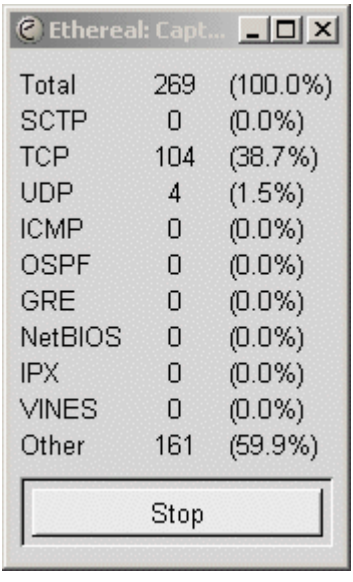


EXHIBIT 53.1 Summary window with statistics about the packets as they are being captured.

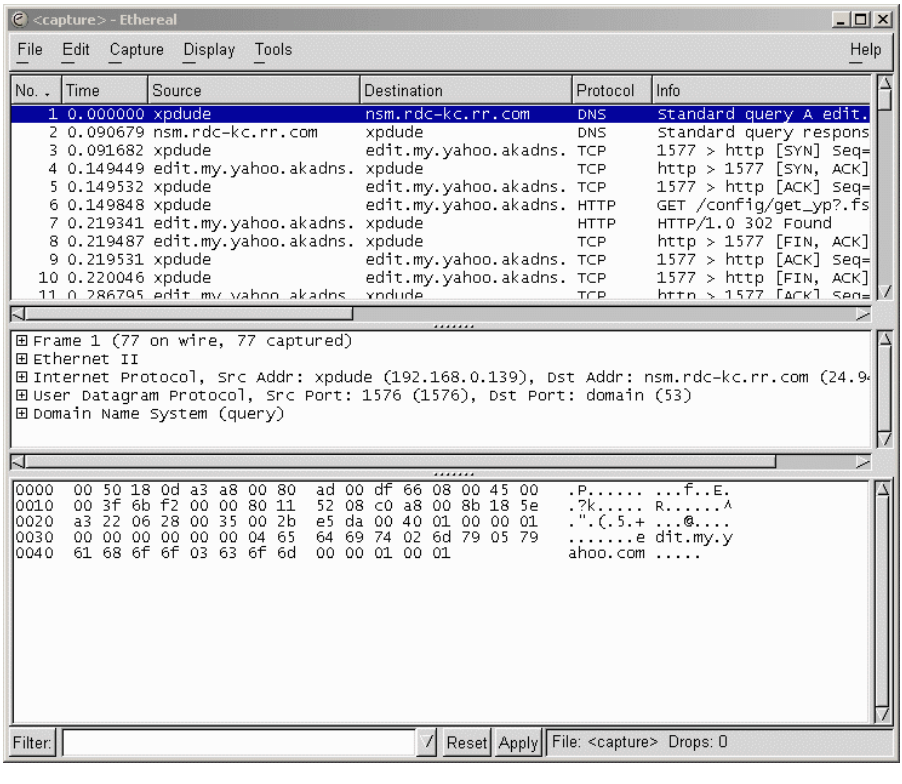
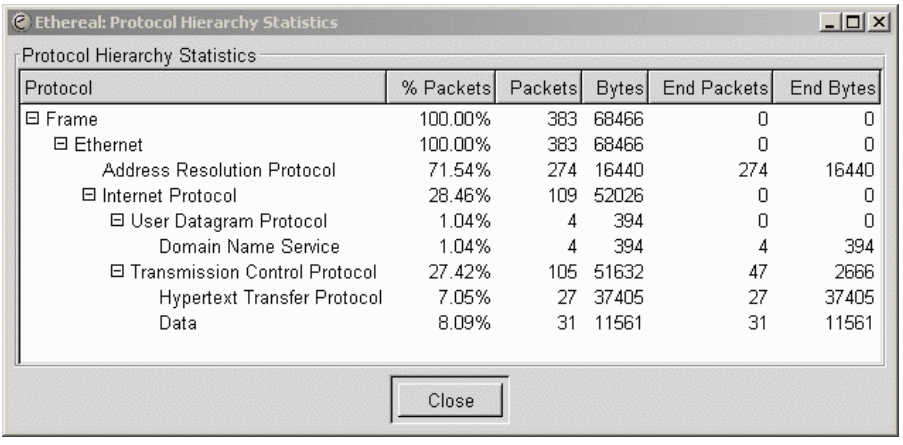


EXHIBIT 53.2 The Ethereal capture session.

Simply click on "Stop" to end the capture session. Exhibit 53.2 shows an example of what the Ethereal capture session looks like. The top window of the session displays the individual packets in the capture session. The information displayed includes the packet number, the time the packet arrived since the capture was



Protocol	% Packets	Packets	Bytes	End Packets	End Bytes
⊟ Frame	100.00%	383	68466	0	0
⊟ Ethernet	100.00%	383	68466	0	0
Address Resolution Protocol	71.54%	274	16440	274	16440
⊟ Internet Protocol	28.46%	109	52026	0	0
⊟ User Datagram Protocol	1.04%	4	394	0	0
Domain Name Service	1.04%	4	394	4	394
⊟ Transmission Control Protocol	27.42%	105	51632	47	2666
Hypertext Transfer Protocol	7.05%	27	37405	27	37405
Data	8.09%	31	11561	31	11561

EXHIBIT 53.3 The protocol hierarchy statistics.

started, the source address of the packet, the destination address of the packet, the protocol, and other information about the packet.

The second window parses and displays the individual packet in an easily readable format, in this case packet number one. Further detail regarding the protocol and the source and destination addresses is displayed in summary format.

The third window shows a data dump of the packet displaying both the hex and ASCII values of the entire packet.

Further packet analysis can be done by clicking on the "Tools" menu. Clicking on "Protocol Hierarchy Statistics" will generate a summary report of the protocols captured during the session. Exhibit 53.3 shows an example of what the protocol hierarchy statistics would look like.

The security practitioner can also get overall statistics on the session, including total packets captured, elapsed time, average packets per second, and the number of dropped packets.

Ethereal is a very powerful tool that is freely available over the Internet. While it may take an expert to fully understand the capture sessions, it does not take an expert to download and install the tool. Certainly the aspiring hacker would have no trouble with the installation and configuration. The security practitioner should understand the availability, features, and ease of use of packet sniffers like Ethereal. Having an awareness of these tools will allow the security practitioner to better understand how the packet sniffer could be used to exploit weaknesses and how to mitigate risk associated with them.

Legitimate Uses

Because the sniffer was invented to help network administrators, many legitimate uses exist for it. Troubleshooting was the first use for the sniffer, but performance analysis quickly followed. Now, many uses for sniffers exist, including those for intrusion detection.

Troubleshooting

The most obvious use for a sniffer is to troubleshoot a network or application problem. From a network troubleshooting perspective, capture tools can tell the network administrator how many computers are communicating on a network segment, what protocols are used, who is sending or receiving the most traffic, and many other details about the network and its hosts. For example, some network-centric applications are very complex and have many components. Here is a list of some of some components that play a role in a typical client/server application:

- Client hardware
- Client software (OS and application)
- Server hardware

- Server software (OS and application)
- Routers
- Switches
- Hubs
- Ethernet network, T1s, T3s, etc.

This complexity often makes the application extremely difficult to troubleshoot from a network perspective. A packet sniffer can be placed anywhere along the path of the client/server application and can unravel the mystery of why an application is not functioning correctly. Is it the network? Is it the application? Perhaps it has to do with lookup issues in a database. The sniffer, in the hands of a skilled network analyst, can help determine the answers to these questions.

A packet sniffer is a powerful troubleshooting tool for several reasons. It can filter traffic based on many variables. For example, let us say the network administrator is trying to troubleshoot a slow client/server application. He knows the server name is *slopoke.xyzcompany.com* and the host's name is *impatient.xyzcompany.com*. The administrator can set up a filter to only watch traffic between the server and client.

The placement of the packet sniffer is critical to the success of the troubleshooting. Because the sniffer only sees packets on the *local* network segment, the sniffer must be placed in the correct location. In addition, when analyzing the capture, the analyst must keep the location of the packet sniffer in mind in order to interpret the capture correctly.

If the analyst suspects the server is responding slowly, the sniffer could be placed on the same network segment as the server to gather as much information about the server traffic as possible. Conversely, if the client is suspected of being the cause, the sniffer should be placed on the same network segment as the client. It may be necessary to place the tool somewhere between the two endpoints.

In addition to placement, the network administrator may need to set up a filter to only watch certain protocols. For instance, if a Web application using HTTP on port 80 is having problems, it may be beneficial to create a filter to only capture HTTP packets on port 80. This filter will significantly reduce the amount of data the troubleshooting will need to sift through to find the problem. Keep in mind, however, that setting this filter can configure the sniffer to miss important packets that could be the root cause of the problem.

Performance and Network Analysis

Another legitimate use of a packet sniffer is for network performance analysis. Many packet sniffer tools can also provide a basic level of network performance and analysis. They can display the general health of the network, network utilization, error rates, summary of protocols, etc. Specialized performance management tools use specialized packet sniffers called RMON probes to capture and forward information to a reporting console. These systems collect and store network performance and analysis information in a database so the information can be displayed on an operator console, or displayed in graphs or summary reports.

Network-Based Intrusion Detection

Network-based intrusion detection systems (IDSs) use a sniffer-like packet capture tool as the primary means of capturing data for analysis. A network IDS captures packets and compares the packet signatures to its database of attacks for known attack signatures. If it sees a match, it logs the appropriate information to the IDS logs. The security practitioner can then go back and review these logs to determine what happened. If in fact the attack was successful, this information can later be used to determine how to mitigate the attack or vulnerability to prevent it from happening in the future.

Verifying Security Configurations

Just as the network administrator can use the sniffer to troubleshoot a network problem, so too can the security practitioner use the sniffer to verify security configurations. A security practitioner can use a packet sniffer to review a VPN application to see if data is being transferred between gateways or hosts in encrypted format.

The packet sniffer can also be used to verify a firewall configuration. For example, if a security practitioner has recently installed a new firewall, it would be prudent to test the firewall to make sure its configuration is stopping the protocols it has been configured to stop. The security practitioner can place a packet sniffer on

the network behind the firewall and then use a separate host to scan ports of the firewall, or open up connections to hosts that sit behind the firewall. If the firewall is configured correctly, it will only allow ports and connections to be established based on its rule set. Any discrepancies could be reviewed to determine if the firewall is misconfigured or if there is simply an underlying problem with the firewall architecture.

Misuse

Sniffing has long been one of the most popular forms of passive attacks by hackers. The ability to "listen" to network conversations is very powerful and intriguing. A hacker can use the packet sniffer for a variety of attacks and information-gathering activities. They can be installed to capture usernames and passwords, gather information on other hosts attached to the same network, read e-mail, or capture other proprietary information or data.

Hackers are notorious for installing *root kits* on their victim hosts. These root kits contain various programs designed to circumvent security on a host and allow a hacker to access a host without the administrator's knowledge. Most modern root kits, or backdoor programs, include tools such as stealth backdoors, keystroke loggers, and often specialized packet sniffers that can capture sensitive information. The SubSeven backdoor for Windows even includes a remotely accessible GUI (graphical user interface) packet sniffer. The GUI makes the packet sniffer easily accessible and simple to use. The packet sniffer can be configured to collect network traffic, save this information into a log, and relay these logs.

Network Discovery

Information gathering is one of the first steps hackers must take when attacking a host. In this phase of the attack, they are trying to learn as much about a host or network as they can. If the attackers have already compromised a host and installed a packet sniffer, they can quickly learn more about the compromised host as well as other hosts with whom that host communicates. Hosts are often configured to trust one another. This trust can quickly be discovered using a packet sniffer. In addition, the attacker can quickly learn about other hosts on the same network by monitoring the network traffic and activity.

Network topology information can also be gathered. By reviewing the IP addresses and subnets in the captures, the attacker can quickly get a feel for the layout of the network. What hosts exist on the network and are critical? What other subnets exist on the network? Are there extranet connections to other companies or vendors? All of these questions can be answered by analyzing the network traffic captured by the packet sniffer.

Credential Sniffing

Credential sniffing is the act of using a packet capture tool to specifically look for usernames and passwords. Several programs exist only for this specific purpose. One such UNIX program called *Esniff.c* only captures the first 300 bytes of all Telnet, FTP, and rlogin sessions. This particular program can capture username and password information very quickly and efficiently.

In the Windows environment, L0phtcrack is a program that contains a sniffer that can capture hashed passwords used by Windows systems using LAN manager authentication. Once the hash has been captured, the L0phtcrack program runs a dictionary attack against the password. Depending on the length and complexity of the password, it can be cracked in a matter of minutes, hours, or days.

Another popular and powerful password sniffing program is *dsniff*. This tool's primary purpose is credential sniffing and can be used on a wide range of protocols including, but not limited to, HTTP, HTTPS, POP3, and SSH.

Use of a specific program like Esniff.c, L0phtcrack, or dsniff is not even necessary, depending on the application or protocol. A simple packet sniffer tool in the hands of a skilled hacker can be very effective. This is due to the very insecure nature of the various protocols. Exhibit 53.4lists some of the protocols that are susceptible to packet sniffing.

E-Mail Sniffing

How many network administrators or security practitioners have sent or received a password via e-mail? Most, if not all, have at some point in time. Very few e-mail systems are configured to use encryption and are therefore

EXHIBIT 53.4 Protocols Vulnerable to Packet Sniffing

Protocol	Vulnerability
Telnet and rlogin	Credentials and data are sent in cleartext
HTTP	Basic authentication sends credentials in a simple encoded form, not encrypted; easily readable if SSL or other encryption is not used
FTP	Credentials and data are sent in cleartext
POP3 and IMAP	Credentials and data are sent in cleartext
SNMP	Community strings for SNMPv1 (the most widely used) are sent in cleartext, including both *public* and *private* community strings

vulnerable to packet sniffers. Not only is the content of the e-mail vulnerable but the usernames and passwords are often vulnerable as well. POP3 (Post Office Protocol version 3) is a very popular way to access Internet e-mail. POP3 in its basic form uses usernames and passwords that are not encrypted. In addition, the data can be easily read.

Security is always a balance of what is secure and what is convenient. Accessing e-mail via a POP3 client is very convenient. It is also very insecure. One of the risks security practitioners must be aware of is that, by allowing POP3 e-mail into their enterprise network, they may also be giving hackers both a username and password to access their internal network. Many systems within an enterprise are configured with the same usernames; and from the user's standpoint, they often synchronize their passwords across multiple systems for simplicity's sake or possibly use a single sign-on system. For example, say John Smith has a username of "JSMITH" and has a password of "FvYQ-6d3." His username would not be difficult to guess, but his password is fairly complex and contains a random string of characters and numbers. The enterprise network that John is accessing has decided to configure its e-mail server to accept POP3 connections because several users, including John, wanted to use a POP3 client to remotely access their e-mail. The enterprise also has a VPN device configured with the same username and password as the e-mail system. If attackers compromise John's password via a packet sniffer watching the POP3 authentication sequence, they may quickly learn they now have access directly into the enterprise network using the same username and password on the Internet-accessible host called "VPN."

This example demonstrates the vulnerability associated with allowing certain insecure protocols and system configurations. Although the password may not have been accessible through brute force, the attackers were able to capture the password in the clear along with its associated username. In addition, they were able to capitalize on the vulnerability by applying the same username and password to a completely separate system.

Advanced Sniffing Tools

Switched Ethernet Networks

"No need to worry. I have a switched Ethernet network." Wrong! It used to be common for network administrators to refer to a switched network as secure. While it is true they are more secure, several vulnerabilities and techniques have surfaced over the past several years that make them less secure.

Reconfigure SPAN/Mirror Port

The most obvious way to capture packets in a switched network is to reconfigure the switch to send all packets to the port into which the packet sniffer is plugged. This can be done with one simple command line in a Cisco router. Once configured, the switch will send all packets for a port, group of ports, or even an entire VLAN directly to the specified port.

This emphasizes the need for increased switch security in today's environments. A single switch without a password, or with a simple password, can allow an intruder access to a plethora of data and information. Incidentally, this is an excellent reason why a single Ethernet switch should not be used inside and outside a firewall. Ideally, the outside, inside, and DMZ should have their own separate physical switches. Also, use a

stronger form of authentication on the network devices other than passwords only. If passwords must be used, make sure they are very complex; and do not use the same password for the outside, DMZ, and inside switches.

Switch Jamming

Switch jamming involves overflowing the address table of a switch with a flood of false MAC addresses. For some switches this will cause the switch to change from "bridging" mode into "repeating" mode, where all frames are broadcast to all ports. When the switch is in repeating mode, it acts like a hub and allows an attacker to capture packets as if they were on the same local area network.

ARP Redirect

An ARP redirect is where a host is configured to send a false ARP request to another host or router. This false request essentially tricks the target host or router into sending traffic destined for the victim host to the attack host. Packets are then forwarded from the attacker's computer back to the victim host, so the victim cannot tell the communication is being intercepted. Several programs exist that allow this to occur, such as *ettercap*, *angst*, and *dsniff*.

ICMP Redirect

An ICMP redirect is similar to the ARP redirect, but in this case the victim's host is told to send packets directly to an attacker's host, regardless of how the switch thinks the information should be sent. This too would allow an attacker to capture packets to and from a remote host.

Fake MAC Address

Switches forward information based on the MAC (Media Access Control) address of the various hosts to which it is connected. The MAC address is a hardware address that is supposed to uniquely identify each node of a network. This MAC address can be faked or forged, which can result in the switch forwarding packets (originally destined for the victim's host) to the attacker's host. It is possible to intercept this traffic and then forward the traffic back to the victim computer, so the victim host does not know the traffic is being intercepted.

Other Switch Vulnerabilities

Several other vulnerabilities related to switched networks exist; but the important thing to remember is that, just because a network is built entirely of switches, it does not mean that the network is not vulnerable to packet sniffing. Even without exploiting a switch network vulnerability, an attacker could install a packet sniffer on a compromised host.

Wireless Networks

Wireless networks add a new dimension to packet sniffing. In the wired world, an attacker must either remotely compromise a system or gain physical access to the network in order to capture packets. The advent of the wireless network has allowed attackers to gain access to an enterprise without ever setting foot inside the premises. For example, with a simple setup including a laptop, a wireless network card, and software packages downloaded over the Internet, an attacker has the ability to detect, connect to, and monitor traffic on a victim's network.

The increase in the popularity of wireless networks has also been followed by an increase in *war-driving*. War-driving is the act of driving around in a car searching for wireless access points and networks with wireless sniffer-like tools. The hacker can even configure a GPS device to log the exact location of the wireless network. Information on these wireless networks and their locations can be added to a database for future reference. Several sites on the Internet even compile information that people have gathered from around the world on wireless networks and their locations.

Reducing the Risk

There are many ways to reduce the risk associated with packet sniffers. Some of them are easy to implement, while others take complete reengineering of systems and processes.

Use Encryption

The best way to mitigate risk associated with packet sniffers is to use encryption. Encryption can be deployed at the network level, in the applications, and even at the host level. Exhibit 53.5 lists the "insecure" protocols discussed in the previous section, and suggests a "secure" solution that can be deployed.

Security practitioners should be aware of the protocols in use on their networks. They should also be aware of the protocols used to connect to and transfer information outside their network (either over the Internet or via extranet connections). A quick way to determine if protocols vulnerable to sniffing are being used is to check the rule set on the Internet or extranet firewalls. If insecure protocols are found, the security practitioner should investigate each instance and determine exactly what information is being transferred and how sensitive the information is. If the information is sensitive and a more secure alternative exists, the practitioner should recommend and implement a secure alternative. Often, this requires the security practitioner to educate the users on the issues associated with using insecure means to connect to and send information to external parties.

IPSec VPNs

A properly configured IPSec VPN can significantly reduce the risk associated with insecure protocols as well. The VPN can be configured from host to host, host to gateway, or gateway to gateway, depending on the environment and its requirements. The VPN "tunnels" the traffic in a secure fashion that prevents an attacker from sniffing the traffic as it traverses the network. Keep in mind, however, that even if a VPN is installed, an attack could still compromise the endpoints of the VPN and have access to the sensitive information directly on the host. This highlights the increased need for strong host security on the VPN endpoint, whether it is a Windows client connecting from a home network or a VPN router terminating multiple VPN connections.

Use Strong Authentication

Because passwords are vulnerable to brute-force attack or outright sniffing over the network, an obvious risk mitigation would be to stop using passwords and use a stronger authentication mechanism. This could involve using Kerberos, token cards, smart cards, or even biometrics. The security practitioner must take into consideration the business requirements and the costs associated with each solution before determining which authentication method suits a particular system, application, or enterprise as a whole.

By configuring a system to use a strong authentication method, the vulnerability of discovered passwords is no longer an issue.

Patches and Updates

To capture packets on the network, a hacker must first compromise a host (assuming the hacker does not have physical access). If all the latest patches have been applied to the hosts, the risk of someone compromising a host and installing a capture tool will be significantly reduced.

EXHIBIT 53.5 Suggestions for Mitigating Risk Associated with Insecure Protocols

Insecure Protocol	Secure Solution
Telnet and rlogin	Replace Telnet or rlogin with Secure Shell (SSH)
HTTP	Run the HTTP or HTTPS session over a Secure Socket Layer (SSL) or Transport Layer Security (TLS) connection
FTP	Replace with secure copy (SCP) or create an IPSec VPN between the hosts
POP3 and IMAP	Replace with SMIME or use PGP encryption
SNMP	Increase the security by using SNMPv2 or SNMPv3, or create a management IPSec VPN between the host and the network management server

Secure the Wiring Closets

Because physical access is one way to access a network, make sure your wiring closets are locked. It is a very simple process to ensure the doors are secured to the wiring closets. A good attack and penetration test will often begin with a check of the physical security and of the security of the wiring closets. If access to a closet is gained and a packet sniffer is set up, a great deal of information can be obtained in short order.

There is an obvious reason why an attack and penetration might begin this way. If the perimeter network and the remote access into a company are strong, the physical security may likely be the weak link in the chain. A hacker who is intent on gaining access to the network goes through the same thought process. Also, keep in mind that with the majority of attacks originating from inside the network, you can mitigate the risk of an internal employee using a packet sniffer in a wiring closet by simply locking the doors.

Detecting Packet Sniffers

Another way to reduce the risk associated with packet sniffers is to monitor the monitors, so to speak. This involves running a tool that can detect a host's network interface cards running in promiscuous mode. Several tools exist, from simple command-line utilities — which tell whether or not a NIC on the local host is running in promiscuous mode — to more elaborate programs such as Antisniff, which actively scans the network segment looking for other hosts with NICs running in promiscuous mode.

Summary

The sniffer can be a powerful tool in the hands of the network administrator or security practitioner. Unfortunately, it can be equally powerful in the hands of the hacker. Not only are these tools powerful, but they are also relatively easy to download off the Internet, install, and use. Security practitioners must be aware of the dangers of packet sniffers and must design and deploy security solutions that mitigate the risks associated with them. Keep in mind that using a packet sniffer to gather credential information on one system can often be used to access other unrelated systems with the same username and password.

54

ISPs and Denial-of-Service Attacks

K. Narayanaswamy, Ph.D.

A denial-of-service (DoS) attack is any malicious attempt to deprive legitimate customers of their ability to access services, such as a Web server. DoS attacks fall into two broad categories:

1. *Server vulnerability DoS attacks:* attacks that exploit known bugs in operating systems and servers. These attacks typically will use the bugs to crash programs that users routinely rely upon, thereby depriving those users of their normal access to the services provided by those programs. Examples of vulnerable systems include all operating systems, such as Windows NT or Linux, and various Internet-based services, such as DNS, Microsoft's IIS Servers, Web servers, etc. All of these programs, which have important and useful purposes, also have bugs that hackers exploit to bring them down or hack into them. This kind of DoS attack usually comes from a single location and searches for a known vulnerability in one of the programs it is targeting. Once it finds such a program, the DoS attack will attempt to crash the program to deny service to other users. Such an attack does not require high bandwidth.
2. *Packet flooding DoS attacks:* attacks that exploit weaknesses in the Internet infrastructure and its protocols. Floods of seemingly normal packets are used to overwhelm the processing resources of programs, thereby denying users the ability to use those services. Unlike the previous category of DoS attacks, which exploit bugs, flood attacks require high bandwidth in order to succeed. Rather than use the attacker's own infrastructure to mount the attack (which might be easier to detect), the attacker is increasingly likely to carry out attacks through intermediary computers (called *zombies*) that the attacker has earlier broken into. Zombies are coordinated by the hacker at a later time to launch a *distributed* DoS (DDoS) attack on a victim. Such attacks are extremely difficult to trace and defend with the present-day Internet. Most zombies come from home computers, universities, and other vulnerable infrastructures. Often, the owners of the computers are not even aware that their machines are being co-opted in such attacks. The hacker community has invented numerous scripts to make it convenient for those interested in mounting such attacks to set up and orchestrate the zombies. Many references are available on this topic.[1–4]

We will invariably use the term "DoS attacks" to mean all denial-of-service attacks, and DDoS to mean flood attacks as described above.

As with most things in life, there is good news and bad news in regard to DDoS attacks. The bad news is that there is no "silver bullet" in terms of technology that will make the problem disappear. The good news, however, is that with a combination of common-sense processes and practices with, in due course, appropriate technology, the impact of DDoS attacks can be greatly reduced.

The Importance of DDoS Attacks

Many wonder why network security and DDoS problems in particular have seemingly increased suddenly in seriousness and importance. The main reason, ironically, is the unanticipated growth and success of ISPs. The rapid growth of affordable, high-bandwidth connection technologies (such as DSL, cable modem, etc.) offered by various ISPs has brought in every imaginable type of customer to the fast Internet access arena: corporations, community colleges, small businesses, and the full gamut of home users.

Unfortunately, people who upgrade their bandwidth do not necessarily upgrade their knowledge of network security at the same time; all they see is what they can accomplish with speed. Few foresee the potential security dangers until it is too late. As a result, the Internet has rapidly become a high-speed network with depressingly low per-site security expertise. Such a network is almost an ideal platform to exploit in various ways, including the mounting of DoS attacks. Architecturally, ISPs are ideally situated to play a crucial role in containing the problem, although they have traditionally not been proactive on security matters.

A recent study by the University of San Diego estimates that there are over 4000 DDoS attacks every week.[5] Financial damages from the infamous February 2000 attacks on Yahoo, CNN, and eBay were estimated to be around $1 billion.[6] Microsoft, Internet security watchdog CERT, the Department of Defense, and even the White House have been targeted by attackers. Of course, these are high-profile installations, with some options when it comes to responses. Stephen Gibson documents how helpless the average enterprise might be to ward off DDoS attacks (at www.scr.com). There is no doubt that DoS attacks are becoming more numerous and deadly.

Why Is DDoS an ISP Problem?

When major corporations suffer the kind of financial losses just described and given the fanatically deterministic American psyche that requires a scapegoat (if not a reasonable explanation) for every calamity and the litigious culture that has resulted from it, rightly or wrongly, someone is eventually going to pay dearly. The day is not far off when, in the wake of a devastating DDoS attack, an enterprise will pursue litigation against the owner of the infrastructure that could (arguably) have prevented an attack with due diligence. A recent article explores this issue further from the legal perspective of an ISP.[7]

Our position is not so much that you need to handle DDoS problems proactively today; however, we do believe you would be negligent not to examine the issue immediately from a cost/benefit perspective. Even if you have already undertaken such an assessment, you may need to revisit the topic in light of new developments and the state of the computing world after September 11, 2001.

The Internet has a much-ballyhooed, beloved, open, chaotic, *laissez faire* philosophical foundation. This principle permeates the underlying Internet architecture, which is optimized for speed and ease of growth and which, in turn, has facilitated the spectacular explosion and evolution of this infrastructure. For example, thus far, the market has prioritized issues of privacy, speed, and cost over other considerations such as security. However, changes may be afoot and ISPs should pay attention.

Most security problems at various enterprise networks are beyond the reasonable scope of ISPs to fix. However, the DDoS problem is indeed technically different. Individual sites *cannot* effectively defend themselves against DDoS attacks without some help from their infrastructure providers. When under DDoS attack, the enterprise cannot block out the attack traffic or attempt to clear upstream congestion to allow some of its desirable traffic to get through. Thus, the very nature of the DDoS problem virtually compels the involvement of ISPs. The best possible outcome for ISPs is to jump in and shape the emerging DDoS solutions voluntarily with dignity and concern, rather than being perceived as having been dragged, kicking and screaming, into a dialogue they do not want.

Uncle Sam is weighing in heavily on DDoS as well. In December 2001, the U.S. Government held a DDoS technology conference in Arlington, Virginia, sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Joint Task Force–Central Network Operations. Fourteen carefully screened companies were selected to present their specific DDoS solutions to the government. Newly designated cyber-security czar Richard Clarke, who keynoted the conference, stressed the critical importance of DDoS and how the administration views this problem as a threat to the nation's infrastructure, and that protecting the Internet infrastructure is indeed part of the larger problem of homeland security. The current Republican administration, one might safely assume, is disposed toward deregulation and letting the market sort out the DDoS problem. In the reality of post-September 11 thinking, however, it is entirely conceivable that ISPs will eventually be forced to contend with government regulations mandating what they should provide by way of DDoS protection.

What Can ISPs Do About DDoS Attacks?

When it comes to DDoS attacks, security becomes a two-way street. Not only must the ISP focus on providing as much protection as possible against incoming DDoS attacks against its customers, but it must also do as much as possible to prevent outgoing DDoS attacks from being launched from its own infrastructure against others. All these measures are feasible and cost very little in today's ISP environment. Minimal measures such as these can significantly reduce the impact of DDoS attacks on the infrastructure, perhaps staving off more draconian measures mandated by the government.

An ISP today must have the ability to contend with the DDoS problem at different levels:

- Understand and implement best practices to defend against DDoS attacks.
- Understand and implement necessary procedures to help customers during DDoS attacks.
- Assess DDoS technologies to see if they can help.

We address each of these major areas below.

Defending against DDoS Attacks

In discussing what an ISP can do, it is important to distinguish the ISP's own infrastructure (its routers, hosts, servers, etc.), which it fully controls, from the infrastructure of the customers who lease its Internet connectivity, which the ISP cannot, and should not, control. Most of the measures we recommend for ISPs are also appropriate for their customers to carry out. The extent to which ISPs can encourage or enable their customers to follow these practices will be directly correlated to the number of DDoS attacks.

Step 1: Ensure the Integrity of the Infrastructure

An ISP plays a critical role in the Internet infrastructure. It is, therefore, very important for ISPs to ensure that their own routers and hosts are resistant to hacker compromise. This means following all the necessary best practices to protect these machines from break-ins and intrusions of any kind. Passwords for user and root accounts must be protected with extra care, and old accounts must be rendered null and void as soon as possible.

In addition, ISPs should ensure that their critical servers (DNS, Web, etc.) are always current on software patches, particularly if they are security related. These programs will typically have bugs that the vendor eliminates through new patches.

When providing services such as Telnet, FTP, etc., ISPs should consider the secure versions of these protocols such as SSH, SCP, etc. The latter versions use encryption to set up secure connections, making it more difficult for hackers using packet sniffing tools to acquire usernames and passwords, for example.

ISPs can do little to ensure that their users are as conscientious about these matters as they ought to be. However, providing users with the knowledge and tools necessary to follow good security practices themselves will be very helpful.

Step 2: Resist Zombies in the Infrastructure

Zombies are created by hackers who break into computers. Although by no means a panacea, tools such as intrusion detection systems (IDSs) provide some amount of help in detecting when parts of an infrastructure have become compromised. These tools vary widely in functionality, capability, and cost. They have a lot of utility in securing computing assets beyond DDoS protection. (A good source on this topic is Reference 8.) Certainly, larger customers of the ISP with significant computing assets should also consider such tools.

Where possible, the ISP should provide users (e.g., home users or small businesses) with the necessary software (e.g., downloadable firewalls) to help them. Many ISPs are already providing free firewalls, such as ZoneAlarm, with their access software. Such firewalls can be set up to maximize restrictions on the customers' computers (e.g., blocking services that typical home computers are never likely to provide). Simple measures like these can greatly improve the ability of these computers to resist hackers.

Most zombies can be now be discovered and removed from a computer by the traditional virus scanning software from McAffee, Symantec, and other vendors. It is important to scan not just programs but also any documents with executable content (such as macros). In other words, everything on a disk requires scanning. The only major problem with all virus scanning regimes is that they currently use databases that have signatures of known viruses, and these databases require frequent updates as new viruses are created.

As with firewalls, at least in cases where users clearly can use the help, the ISP could try bundling its access software, if any, with appropriate virus scanning software and make it something the user has to contend with before getting on the Internet.

Step 3: Implement Appropriate Router Filters

Many DDoS attacks (e.g., Trinoo, Tribal Flood, etc.) rely on source address spoofing, an underlying vulnerability of the Internet protocols whereby the sender of a packet can conjure up a source address other than his actual address. In fact, the protocols allow packets to have completely fabricated, nonexistent source addresses. Several attacks actually rely on this weakness in the Internet. This makes attacks much more difficult to trace because one cannot figure out the source just by examining the packet contents because the attacker controls that.

There is no legitimate reason why an ISP should forward outgoing packets that do not have source addresses from its known legitimate range of addresses. It is relatively easy, given present-day routers, to filter outgoing packets at the border of an ISP that do not have valid source addresses. This is called ingress filtering, described in more detail in RFC 2267.

Routers can also implement egress filtering at the point where traffic enters the ISP to ensure that source addresses are valid to the extent possible (e.g., source addresses cannot be from the ISP, packets from specific interfaces must match expected IP addresses, etc.). Note that such filters do not eliminate all DDoS attacks; however, they do force attackers to use methods that are more sophisticated and do not rely on ISPs forwarding packets with obviously forged source addresses.

Many ISPs also have blocks of IP addresses set aside that will never be the source or destination of Internet traffic (see RFC 1918). These are addresses for traffic that will never reach the Internet. The ISP should neither accept traffic with this destination, nor should it allow outbound traffic from those IP addresses set aside in this manner.

Step 4: Disable Facilities You May Not Need

Every port that you open (albeit to provide a legitimate service) is a potential gate for hackers to exploit. Therefore, ISPs, like all enterprises, should ensure they block any and all services for which there is no need. Customer sites should certainly be provided with the same recommendations.

You should evaluate the following features to see if they are enabled and what positive value you get from their being enabled in your network:

- *Directed broadcast.* Some DDoS attacks rely on the ability to broadcast packets to many different addresses to amplify the impact of their handiwork. Directed broadcast is a feature that should not be needed for inbound traffic on border routers at the ISP.
- *Source routing.* This is a feature that enables the sender of a packet to specify an ISP address through which the packet must be routed. Unless there is a compelling reason not to, this feature should be disabled because compromised computers within the ISP infrastructure can exploit this feature to become more difficult to locate during attacks.

Step 5: Impose Rate Limits on ICMP and UDP Traffic

Many DDoS attacks exploit the vulnerability of the Internet where the entire bandwidth can be filled with undesirable packets of different descriptions. ICMP (Internet Control Message Protocol, or ping) packets and User Datagram Protocol (UDP) are examples of this class of packets. You cannot completely eliminate these kinds of packets, but neither should you allow the entire bandwidth to be filled with such packets.

The solution is to use your routers to specify rate limits for such packets. Most routers come with simple mechanisms called class-based queuing (CBQ), which you can use to specify the bandwidth allocation for different classes of packets. You can use these facilities to limit the rates allocated for ICMP, UDP, and other kinds of packets that do not have legitimate reasons to hog all available bandwidth.

Assisting Customers during a DDoS Attack

It is never wise to test a fire hydrant during a deadly blaze. In a similar manner, every ISP will do well to think through its plans should one of its customers become the target of DDoS attacks. In particular, this will entail full understanding and training of the ISP's support personnel in as many (preferably all) of the following areas as possible:

- *Know which upstream providers forward traffic to the ISP.* ISP personnel need to be familiar with the various providers with whom the ISP has Internet connections and the specific service level agreements (SLAs) with each, if any. During a DDoS attack, bad traffic will typically flow from one or more of these upstream providers, and the options of an ISP to help its customers will depend on the specifics of its agreements with its upstream providers.
- *Be able to identify and isolate traffic to a specific provider.* Once the customer calls during a DDoS directed at his infrastructure, the ISP should be able to determine the source of the bad traffic. All personnel should be trained in the necessary diagnostics to do so. Customers will typically call with the ISP addresses they see on the attack traffic. While this might not be the actual source of the attack, because of source spoofing, it should help the ISP in locating which provider is forwarding the bad traffic.
- *Be able to filter or limit the rate of traffic from a given provider.* Often, the ISP will be able to contact the upstream provider to either filter or limit the rate of attack traffic. If the SLA does not allow for this, the ISP can consider applying such a filter at its own router to block the attack traffic.
- *Have reliable points of contact with each provider.* The DDoS response by an ISP is only as good as its personnel and their knowledge of what to do and whom to contact from their upstream providers. Once again, such contacts cannot be cultivated after an attack has occurred. It is better to have these pieces of information in advance. Holding DDoS attack exercises to ensure that people can carry out their duties during such attacks is the best way to make sure that everyone knows what to do to help the customer.

Assessing DDoS Technologies

Technological solutions to the DDoS problem are intrinsically complex. DDoS attacks are a symptom of the vulnerabilities of the Internet, and a single site is impossible to protect without cooperation from upstream infrastructure. New products are indeed emerging in this field; however, if you are looking to eliminate the problem by buying an affordable rack-mountable panacea that keeps you in a safe cocoon, you are fresh out of luck.

Rather than give you a laundry list of all the vendors, I am going to categorize these products somewhat by the problems they solve, their features, and their functionality so that you can compare apples to apples. Still, the comparison can be a difficult one because various products do different things and more vendors are continually entering this emerging, niche market.

Protection against Outgoing DDoS Attacks

Unlike virus protection tools, which are very general in focus, these tools are geared just to find DoS worms and scripts. There are basically two kinds of products that you can find here.

Host-Based DDoS Protection

Such protection typically prevents hosts from being taken over as zombies in a DDoS attack. These tools work in one of two major ways: (1) signature analysis, which, like traditional virus scanners, stores a database of known scripts and patterns and scans for known attack programs; and (2) behavior analysis, which monitors key system parameters for the behavior underlying the attacks (rather than the specific attack programs) and aborts the programs and processes that induce the underlying bad behavior.

Established vendors of virus scanning products, such as McAffee, Symantec, and others, have extended their purview to include DoS attacks. Other vendors provide behavior-analytic DDoS protection that essentially detects and prevents DDoS behavior emanating from a host. The major problem with host-based DDoS protection, from an ISP's perspective, is that one cannot force the customers to use such tools or to scan their disks for zombies, etc.

Damage-Control Devices

A few recent products (such as Captus' *Captio* and Cs3, Inc.'s *Reverse Firewall*[9,10]) focus on containing the harm that DDoS attacks can do in the outgoing direction. They restrict the damage from DDoS to the smallest possible network. These devices can be quite useful in conjunction with host-based scanning tools. Note that the damage-control devices do not actually prevent an infrastructure from becoming compromised; however, they do provide notification that there is bad traffic from your network and provide its precise origin. Moreover, they give you time to act by throttling the attack at the perimeter of your network and sending you a notification.

ISPs could consider using these devices as insurance to insulate themselves from the damage bad customers can do to them as infrastructure providers.

Protection against Incoming Attacks

As we have mentioned before, defending against incoming attacks at a particular site requires cooperation from the upstream infrastructure. This makes DDoS protection products quite complex. Moreover, various vendors have tended to realize the necessary cooperation in very different ways. A full treatment of all of these products is well beyond the scope of this chapter. However, here are several issues you need to consider as an ISP when evaluating these products:

- *Are the devices inline or offline?* An inline device will add, however minimally, to the latency. Some of the devices are built using hardware in an effort to reduce latency. Offline devices, while they do not have that problem, do not have the full benefit of looking at all the traffic in real-time. This could affect their ability to defend effectively.
- *Do the devices require infrastructure changes and where do they reside?* Some of the devices either replace or deploy alongside existing routers and firewalls. Other technologies require replacement of the existing infrastructure. Some of the devices need to be close to the core routers of the network, while most require placement along upstream paths from the site being protected.
- *How do the devices detect DDoS attacks and what is the likelihood of false positives?* The degree of sophistication of the mechanism of detection and its effectiveness in indicating real attacks is all-important in any security technology. After all, a dog that barks the entire day does protect you from some burglars — but you just might stop listening to its warnings! Most of the techniques use comparisons of actual traffic to stored profiles of attacks, or "normal" traffic, etc. A variety of signature-based heuristics are applied to detect attacks. The jury is still out on how effective such techniques will be in the long run.
- *How do the devices know where the attack is coming from?* A major problem in dealing effectively with DDoS attacks is to know, with any degree of certainty, the source of the attacks. Because of source address spoofing on the Internet, packets do not necessarily have to originate where they say they do. All the technologies have to figure out is from where in the upstream infrastructure the attack traffic is flowing. It is the routers along the attack path that must cooperate to defend against the attack. Some of the approaches require that their devices communicate in real-time to form an aggregate picture of where the attack is originating.
- *What is the range of responses the devices will take and are you comfortable with them?* Any DDoS defense must minimally stop the attack from reaching the intended victim, thereby preventing the victim's computing resources from deteriorating or crashing. However, the real challenge of any DDoS defense is to find ways for legitimate customers to get through while penalizing only the attackers. This turns out to be *the* major technical challenge in this area. The most common response includes trying to install appropriate filters and rate limits to push the attack traffic to the outer edge of the realm of control of these devices. At the present time, all the devices that provide DDoS defense fall into this category. How effective they will be remains to be seen.

The products mentioned here are quite pricey even though the technologies are still being tested under fire. DDoS will have to be a very important threat in order for smaller ISPs to feel justified in investing their dollars in these devices. Finally, many of the approaches are proprietary in nature, so side-by-side technical comparisons are difficult to conduct. Some industry publications do seem to have tested some of these devices in various ways. A sampling of vendors and their offerings, applying the above yardsticks, is provided here:

- *Arbor Networks* (www.arbornetworks.com): offline devices, near core routers, anomaly-based detection; source is tracked by communication between devices, and defense is typically the positioning of a filter at a router where the bad traffic enters the network
- *Asta Networks* (www.astanetworks.com): offline devices that work alongside routers within a network and upstream, signature-based detection; source is tracked by upstream devices, and defense is to use filters at upstream routers

- *Captus Networks*
  (www.captusnetworks.com): inline device used to throttle incoming or outgoing attacks; uses windowing to detect non-TCP traffic and does not provide ways for customers to get in; works as a damage-control device for outgoing attacks
- *Cs3, Inc.*
  (www.cs3-inc.com): inline devices, modified routers, and firewalls; routers mark packets with path information to provide fair service, and firewalls throttle attacks; source of the attack provided by the path information, and upstream neighbors are used to limit attack traffic when requested; *Reverse Firewall* is a damage-control device for outgoing attacks
- *Mazu Networks*
  (www.mazunetworks.com): inline devices at key points in network; deviations from stored historical traffic profile indicate attack; the source of the attack is pinpointed by communication between devices, and defense is provided by using filters to block out the bad traffic
- *Okena*
  (www.okena.com): host-based system that has extended intrusion detection facilities to provide protection against zombies; it is a way to keep one's infrastructure clean but is not intended to protect against incoming attacks

Important Resources

Finally, the world of DoS, as is indeed the world of Internet security, is dynamic. If your customers are important to you, you should have people that are on top of the latest threats and countermeasures. Excellent resources in the DoS security arena include:

- *Computer Emergency Response Team (CERT)* (www.cert.org): a vast repository of wisdom about all security-related problems with a growing section on DoS attacks; you should monitor this site regularly to find out what you need to know about this area. This site has a very independent and academic flavor. Funded by the Department of Defense, this organization is likely to play an even bigger role in putting out alerts and other information on DDoS.
- *System Administration, Networking and Security (SANS) Institute* (www.sans.org): a cooperative forum in which you can instantly access the expertise of over 90,000 professionals worldwide. It is an organization of industry professionals, unlike CERT. There is certainly a practical orientation to this organization. It offers courses, conferences, seminars, and White Papers on various topics that are well worth the investment. It also provides alerts and analyses on security incidents through incidents.org, a related facility.

References

1. Houle, K. and Weaver, G., "Trends in Denial of Service Technology," CERT Coordination Center, October 2001, http://www.cert.org/archive/pdf/DOS_trends.pdf.
2. Myers, M., "Securing against Distributed Denial of Service Attacks," Client/Server Connection, Ltd., http://www.cscl.com/techsupp/techdocs/ddossamp.html.
3. Paul, B., "DDOS: Internet Weapons of Mass Destruction," *Network Computing,* Jan. 1, 2001, http://www.networkcomputing.com/1201/1201f1c2.html.
4. Harris, S., "Denying Denial of Service," *Internet Security,* Sept. 2001, http://www.infosecuritymag.com/articles/september01/cover.shtml.
5. Lemos, R., "DoS Attacks Underscore Net's Vulnerability," CNETnews.com, June 1, 2001, http://news.cnet.com/news/0-1003-200-6158264.html?tag=mn_hd.
6. Yankee Group News Releases, Feb. 10, 2000, http://www.yankeegroup.com/webfolder/ yg21a.nsf/press/384D3C49772576EF85256881007DC0EE?OpenDocument.
7. Radin, M.J. et al., "Distributed Denial of Service Attacks: Who Pays?," Mazu Networks, http://www.mazunetworks.com/radin-es.html.
8. SANS Institute Resources, Intrusion Detection FAQ, Version 1.52, http://www.sans.org/newlook/resources/IDFAQ/ID_FAQ.htm.

9. Savage, M., "Reverse Firewall Stymies DDOS Attacks," *Computer Reseller News*, Dec. 28, 2001, http://www.crn.com/sections/BreakingNews/breakingnews.asp? ArticleID=32305.
10. Desmond, P., "Cs3 Mounts Defense against DDOS Attacks," eComSecurity.com, Oct. 30, 2001, http://www.ecomsecurity.com/News_2001-10-30_DDos.cfm.

Further Reading

Singer, A., "Eight Things that ISPs and Network Managers Can Do to Help Mitigate DDoS Attacks," San Diego Supercomputer Center, http://security.sdsc.edu/publications/ddos.shtml.

Domain 3
Security Management Practices

Security management entails the identification of an organization's information assets and the development, documentation, and implementation of policies, standards, procedures, and guidelines. It also includes management tools such as data classification and risk assessment (risk analysis) that are used to identify threats, classify assets, and to rate their vulnerabilities so that effective security controls can be implemented.

In this domain, we address the importance of establishing the foundation for the security program with policies that reflect the organization's philosophy about information asset protection. Among the practices discussed are how to deal with risk and how a practitioner manages risk to develop the trust and assurance required from information systems.

The organization's users are a critical component in achieving and maintaining information assurance. The best information security policy will sit dormant on a shelf unless the security manager has an effective, enterprisewide, ongoing security awareness campaign. Training experts agree that a well-developed communication plan can spell the difference between the success or failure of a security program.

Contents

Section 3.2 Change Control Management

Section 3.3 Data Classification

Section 3.4 Risk Management

Section 3.5 Employment Policies and Practices

Section 3.6 Risk Management

Section 3.7 Security Awareness Training

Section 3.8 Security Management Planning

57

Measuring ROI on Security

Carl F. Endorf, CISSP, SSCP, GSEC

Finding a return on investment (ROI) has never been easy; and for technology, it has been even more difficult. To make matters more complicated, the return on security investment (ROSI) has been nebulous at best. It is easy to say that a Web defacement or hack attack will cause a "loss of customer confidence," but what does that really mean? What is the financial impact on an organization if it experiences a loss of customer confidence? What needs to be determined is the causation of the financial impact and the event itself.[1] I believe that there are clear methods to do this.

The purpose of this chapter is to discuss the basic methods of finding the ROSI for an organization and the implications that this will have on the business of security. We also examine a seven-step analysis to help determine the ROI for security.

Understanding ROI

It is easy to get security money *after* you are attacked, but the problem is trying to get the money before that happens. How do you quantify what security gets you? If you spend an additional $3,000,000 this year on security, how do you justify it? What is the return on that investment? As a security professional, you see different vulnerabilities and attacks on a daily basis and it may be very clear to you that your enterprise needs to be more secure. But from a business perspective, it is not always that clear. Executives realize that threats are a reality, but they want some way to quantify these threats and know what the cost is for implementing a security measure or the financial consequences if they do not.

Many security managers rely on a soft return on investment (SROI) that is not based on actual data but on FUD (fear, uncertainty, and doubt) to sell the need for new security measures or the expansion of existing ones. The idea is that if you can scare enough people they will give you the money. The problem with this is that it can lead to implementing technology that is not always needed or that solves a problem where there is minimal risk of that threat.

Today more than ever, with a recession in the economy, it is difficult to justify with any solid evidence what security expenses are needed. For example, if you need to add three security people and update your firewalls, this will result in more uptime and less downtime on the network, which means the company will make more money; but where is the quantifiable value associated with staffing and firewalls?[2] The SROI will not help justify these costs.

This leads to the better answer of basing security expenditures on real numbers and obtaining a hard return on investment (HROI). The HROI will give a quantitative answer that will help justify the use of security and can help determine the operational cost of security.

Getting an HROI can be accomplished in much the same way a risk assessment is done. The following seven steps are involved in the process:[3]

1. Asset identification and valuation

2. Threat and vulnerability exposure factor (EF)
3. Determine the single loss expectancy (SLE)
4. Annualized rate of occurrence (ARO)
5. Compute the annual loss expectancy (ALE)
6. Survey controls
7. Calculate the ROSI

Asset Identification and Valuation

First, you need to list your organization's tangible and intangible assets. We define "tangible" as an asset that has physical form and "intangible" items as any item that does not have physical form, such as goodwill and intellectual property. Tangible items can usually be tracked easily in small organizations, but this becomes progressively more difficult as the size increases. Typically, larger organizations will have an asset management/tracking area that can provide a list. You will then need to assign a dollar value to each tangible asset, with depreciation taken into account. One way this can be done is as follows:[4]

$$\frac{\text{Cost} - \text{Salvage Value}}{\text{Useful Life}} = \text{Yearly Depreciation}$$

Next, make a list of intangible items. This can be subjective and is based on perceived value, but the following questions will help: "Knowing what you do about the asset, what would you pay to have that asset if you did not already own it?" and "What revenue will this asset bring to the organization in the future?"

Another possibility is to rank all your assets, both tangible and intangible, according to your perceived value of them. Given that you have values for the tangible assets, placement of the intangibles relative to the tangibles should help you in valuing the intangible assets.

Threat and Vulnerability Exposure Factor

Now that the assets have been identified, it is necessary to examine the possible threats to each of these assets. This is not a definite as there are many variables involved, but the subject matter experts for many of these assets can help identify exposures. This is an estimate; it cannot include everything possible because we do not know all the possible exposures.

The next step is to examine the threat and vulnerability exposure factor (EF). The EF is the percentage of loss a realized threat event would have on a specific asset, that is, the consequence. The EF can be a large number, as is the case of a major event such as a fire or a small number like the loss of a hard drive. It can be expressed from 0 to 100 percent of loss if exposed to a specific event. For example, if a virus brought down your Web farm, this may cause a 75 percent loss in the Web farm's functionality.

Determine the Single Loss Expectancy

The single loss expectancy (SLE) measures the specific impact, monetary or otherwise, of a single event. The following formula derives the SLE:[5]

$$\text{Asset value} \times \text{Exposure factor} = \text{SLE}$$

Annualized Rate of Occurrence

The annualized rate of occurrence (ARO) is the frequency with which a threat is expected to occur. The number is based on the severity of controls and the likelihood that someone will get past these controls.[6] ARO values fall within the range from 0.0 (never) to a large number.

The ARO is not a definite number and can be subjective. It is best based on probability from observed data, much like insurance. You will need to look at your organization's metrics on hardware, software, and past threats. For example, Company X looks at the past five years' incident handling data and finds that there was

an average of three attempts per external employee for the 100 external employees attempting unauthorized access. This would calculate to an ARO of 300, or 3 attempts × 100 external employees = 300.

Annual Loss Expectancy

The annual loss expectancy (ALE) can now be determined from the data collected. The following formula sets for the calculation needed:

$$\text{Single loss expectancy (SLE)} \times \text{Annual rate of occurrence (ARO)} = \text{ALE}$$

The ALE is the number you can use to justify your security expenditures. For example, you want to protect your payroll server within the company. The server itself will not cause a direct loss to the company if compromised, but will result in loss of reputation if exposed. The value of the system itself is $10,000, and the information and loss of reputation is placed at $250,000. The SLE has been placed at 75 percent and the ARO at 0.3. Using the formula above, we obtain an SLE of $58,500 ($260,000 × 0.75) × 0.3 = $58,500. Once the ALE is known, it can be used by information security management to determine a cost-effective risk mitigation strategy.[3]

Survey Controls

It is now essential to survey the controls that you have in your existing security architecture and examine the SLE of those assets. If the loss expectancy is exceptionally high, you would want to consider new controls to mitigate those threats. For example, using the situation in the previous section, we have an SLE of $58,000; but if we are spending $100,000 a year to protect it, we are spending more than we need and new controls should be selected. It is best if each exposure has a control identified for it on a per-exposure basis.

Calculate Your ROSI

Now we are at the point of being able to calculate the ROSI. The basic calculation for ROI is the Return/Assets. Therefore, we can subtract the cost of what we expect to lose in a year for a specific asset from the annual cost of the control:

$$\text{Annual loss expectancy (ALE)} - \text{Current cost of control (CCC)} = \text{ROSI}$$

For example, if in the past we had a cost of $500,000 a year due to security breaches and we add an intrusion detection system (IDS) that costs the company $250,000 a year (this includes support, maintenance, and management) and is 80 percent effective, then we have a positive ROI of $150,000.

ROSI Example

Now apply the seven steps to the following situation. You are asked to protect a small database that contains critical business data. The data has been valued at $5,000,000 and has never been compromised. Based on recent events in similar companies with this type server and data, the probability of an attack has been estimated to happen about once every 20 years. You are asked to look at the current access controls in place that are costing the company $95,000 a year to maintain and see what the ROSI is on these controls.

As you can see from Exhibit 57.1, the total ROSI for the current access control gives the organization a positive ROSI of $130,000 per year.

Arguments against ROSI

One argument is that valuating the ROSI lacks precision and is based on approximations. This is true to an extent; but as more data is collected within your organization and the industry, the picture will become clearer, much like insurance actuarial tables can predict the probabilities of certain events. Another argument is that these hard numbers can give a company a false sense of security because the company feels these numbers are

EXHIBIT 57.1 ROSI for Proprietary Confidential Data

Steps			Formula
Asset identification and valuation	**Asset**: proprietary confidential data	Valuation: $5,000,000	
Threat and vulnerability exposure factor (EF)	**Threat:** disclosure of data	**EF:** 90%	
Determine the single loss expectancy (SLE)	$5,000,000 × .90 =	**SLE:** $4,500,000	Asset Value × Exposure Factor = SLE
Annualized rate of occurrence (ARO)	Based on observed data, the probability is 1 in 20 years	ARO = 0.05	
Compute the annual loss expectancy (ALE)	$4,500,000 × .05 =	**ALE** = $225,000	Single Loss Expectancy (SLE) × Annual Rate of Occurrence (ARO) = ALE
Survey controls	Current controls are costing $95,000		
Calculate ROSI	$225,000 – $95,000	ROSI = $130,000	Annual loss expectancy (ALE) – Current cost of control (CCC) = ROSI

exact but needs to keep in mind that they need reevaluation. Another argument is that that the ROSI is immutable; but if it is made a part of the annual review process, this should not be the case.[3]

Conclusion

This chapter discussed a seven-step methodology to help determine the ROSI for an organization. The methods used were basic and could each be explained in much more depth, but they do illustrate that hard numbers can be obtained. These hard numbers help security managers to go away from using FUD and relying on better data. The data presented here is based on the principles of probability theory and statistics.

Although much of the data that the ROSI is based on is still in its infancy, it will likely take shape in the near future. The key is getting credible data to base the numbers on. We see this taking shape in the insurance industry as hacking insurance is being offered; these are steps in the right direction. It is likely that the insurance industry will be a driving force in the science of ROSI.[2]

References

1. Karofsky, Eric, (2001). Insight into Return on Security Investment, *Secure Business Quarterly,* Volume 1, Issue Two, Fourth Quarter. www.sbq.com.
2. Berinato, Scott (2002). Finally, a Real Return on Security Spending, *CIO Magazine,* February 15, pp. 43–52.
3. Pfleeger, Charles, P. (1997). *Security in Computing*. Upper Saddle River, NJ: Prentice Hall, Inc.
4. Adams, S., Pryor, L., and Keller, D. (1999). *Financial Accounting Information: A Decision Case Approach*. Cincinnati, OH: South-Western College Publishing.
5. Tipton, Harold F. and Krause, Micki. (2000). *Information Security Management Handbook, 4th edition*. Boca Raton, FL: CRC Press LLC.
6. McCammon, Keith (2002). *Calculating Loss Expectancy*. Electronic version, retrieved March 10, 2003. http://mccammon.org/articles/loss_expectancy

58

Security Patch Management

Jeffrey Davis, CISSP

Patch management is an important part of securing your computing environment. New security vulnerabilities are found in software and systems every day, and these vulnerabilities can introduce risk into an organization's information technology infrastructure. Patches and updates to systems are needed to mitigate these vulnerabilities. Gartner Group reports that 90 percent of machines are exploited using known vulnerabilities that had patches available. A good patch management process can minimize these risks by ensuring the patches are applied. It can also shorten the timeframe that an organization is exposed to newly discovered vulnerabilities by making sure patches are applied in a timely manner. These patch management processes consist of several different components and need to take into account an organization's structure, policies, risk tolerance, and available resources.

Why Patch Management?

Patch management provides a method to significantly reduce risks to your computing environment. There have been a number of large impacting worm outbreaks that used known vulnerabilities as their mechanism for spreading. Some examples of these include:

- *Sadmind* used a known UNIX vulnerability to spread from UNIX machine to UNIX machine, as well as a Microsoft IIS vulnerability to deface Web sites.
- *Code Red I and II* used known vulnerabilities in Microsoft IIS servers.
- *Nimda* used known vulnerabilities in Microsoft IIS servers.

All of these vulnerabilities had patches available from vendors. In some cases these patches had been available for more than two years before the worm outbreak. Exhibit 58.1 shows the length of time that elapsed between the discovery of the vulnerability and the release of its patch, and the outbreak of a worm that exploited it.

Each one of these worms spread quickly and had a devastating effect on Internet traffic as well as organizations' internal networks. The speed and breadth of the spread indicated that many machines were not updated and were vulnerable to the exploit used to infect those machines. The loss of service and productivity caused by these worms could have been minimized by good patch management practices.

These worms also demonstrated the risk of vulnerable machines connected to networks and the ability of those machines to deny service to other machines. In some cases, single machines that were infected were able to flood local area networks with enough packets to saturate the available bandwidth. This shows that vulnerable machines are not only vulnerable to compromise themselves, but they are also at risk to deny service to other machines that share network resources. This greatly increases the risk of having a machine on a network that is not patched against known vulnerabilities.

Another threat is hackers looking for machines to compromise. One of the methods employed is to use automated tools to scan for vulnerable machines. Once these machines are identified, they are compromised and then used to scan for more machines to exploit. Machines that are fully patched will be more difficult to

EXHIBIT 58.1 Length of Time between Discovery and Worm Outbreak

Worm	Date of Outbreak	Vector of Infection	Date Patch for Vulnerability Was Made Available	No. of Days Systems Were Vulnerable
Sadmind/IIS	5/8/2000	Sadmind daemon on Solaris, used to deface Microsoft IIS Web sites	Sadmind patch available 12/29/1999, IIS patch available 10/17/2000	Sadmind: 496 days IIS: 203 days
Code Red I	7/19/2001	Microsoft IIS	Patch available 6/18/2001	31 days
Nimda	9/18/2001	Microsoft IIS and e-mail clients	IIS patch available 5/15/2001 E-mail client patch available 4/3/2001	IIS: 126 days E-mail client: 156 days

compromise and will be passed over in favor of machines that have vulnerabilities. Keeping systems patched against known vulnerabilities will greatly reduce the risk of being compromised in this fashion.

One specific example where patch management directly mitigates a threat is in the updating of anti-virus software. Viruses are one of the biggest threats to computer systems and spread through many different vectors. Anti-virus software is used to mitigate this threat and prevent systems from being infected. When new viruses come out, the anti-virus software needs to be updated to detect and clean them. These updates can come out at various time intervals, varying from as often as hourly to as infrequent as weekly. In older versions of anti-virus software, these updates had to be retrieved and applied manually. Most new versions will retrieve and update the software via the Internet with no user interaction and at prescribed intervals. Automation of this process and its integration into the software have greatly improved the level of protection against virus threats and reduced the threat of virus infections because of outdated anti-virus software. Other software applications have also begun to automate the patching process but a large majority of software still exists that needs to be patched manually or through the use of third-party automated tools. A good patch management process can ensure that this software is kept up-to-date and leverage the automated processes, where possible.

All in all, patch management is a proactive way of reducing risk in an organization's information systems environment. The amount of time spent in applying patches to systems is time well spent and can help prevent loss due to system compromise through vulnerabilities that those patches are meant to mitigate.

Types of Patches

Patches for software come in different types. These can be point patches/hotfixes, bundled patches, and version releases. Point patches/hotfixes are released to address specific issues and most security patches are released in this category. These patches usually undergo a minimal amount of testing because the issues need to be addressed quickly. These fixes can be riskier to apply to systems because they are not tested thoroughly and may cause errors to applications. Vendors may not guarantee that these fixes will not break other functions. Point fixes are then usually rolled up into bundled patches that undergo more rigorous testing and are fully supported by the vendor. They come out at regular time intervals and are usually larger in size because they contain multiple fixes. Organizations usually prefer to apply these bundled patches because they are more fully tested than point patches. Version releases are similar to bundled patches but may also contain new features or functions. It is also significant to note that some patches may require that a system have other patches applied (prerequisites) or be at a specific version in order to function correctly. This may require a system to be updated to a more current version just to apply a point fix that addresses a new vulnerability. A patch management process needs to take all of these types of patches into consideration and be able to deal with each one appropriately.

It is also important to make sure that patches are obtained from a trusted source and verified as authentic. There have been cases where software has been modified to include backdoors that allow unauthorized access. To mitigate this risk, some vendors will digitally sign their patches. These signatures should be checked before applying the patch to ensure that the patch is authentic and has not been changed.

Software Phases

Software will progress through different phases during its use, and vendors will support software differently, depending on the phase the software is in. The first phase that is important to a patch management process is the initial *introduction phase.* During this stage, there may be many software problems found that will need to be resolved. Patches may need to be applied quickly not only to solve security issues, but also functionality issues. Vendors are usually very supportive in this phase, with timely updates. The next phase is *production use.* This is the stage in which most software is utilized. Vulnerabilities found in this stage will usually go through an assessment to determine their impact. Some vendors will be more proactive and timely in providing patches for vulnerabilities while others may want to provide patches as part of normal version release cycles. Patch management processes need to be able to track and distribute the various point patches as well as the bundled patches and version updates that need to be applied. The last phase that software goes through that is important to patch management is the *unsupported* or *legacy stage.* In this stage, the vendor may no longer support the software and patches may not be available. This software can be risky to run as part of an organization because any new vulnerability found cannot be mitigated. While a patch management process may not be helpful in this situation, because there are no patches to be applied, it is important to know of the existence of this software so that its risks to an organization can be evaluated and understood. Vendors usually make users aware, well ahead of time, of when software is no longer supported. Planning ahead for this situation is important.

Threat Awareness and Assessment

One challenge in patch management is to know when a new vulnerability has been discovered. This is important in keeping the window of opportunity to be exploited by a vulnerability to a minimum. There are a variety of methods to keep up-to-date with new vulnerabilities. One is the monitoring of alerts by the Computer Emergency Response Team (CERT) at Carnegie Mellon. This is a U.S. Government-sponsored team that monitors and coordinates computer security incidents and vulnerabilities. The CERT Web site and mailing list provide a good source of alerts for newly discovered vulnerabilities. Another good source of information is the Bugtraq mailing list currently hosted at Security Focus. This is a moderated mailing list that discusses vulnerabilities to systems. Exhibit 58.2 lists the Web sites for these sources as well as some others, and can be very useful.

Major software vendors will also have mailing lists used to alert their users to software vulnerabilities and how to obtain patches for them. Monitoring these lists is an important part of staying aware of new vulnerabilities and the actions needed to mitigate them.

Once patches are made available, a decision on the process of applying them must be made. As mentioned, some point patches may not be fully tested before being released. Most system administrators will want to try the patches on a test machine before applying them to production systems. They will also want to fully understand the process to remove a patch if it causes errors to the application. Installing patches on a test machine will enable system administrators to practice the patch application and removal process. Production systems may also have designated maintenance windows to minimize downtime. These windows can occur as often as once a day to as infrequently as once every three months or longer. They usually occur during periods of low activity so that the impact to system users is low. There may also be designated "quiet times" during which only emergency changes can be made. These quiet times can be centered on business-critical activities

EXHIBIT 58.2 List of Vulnerability Information Sources

Organization	Web Site
Computer Emergency Response Team Coordination Center	www.cert.org
SANS Incident Storm Center	isc.incidents.org
SecurityFocus Vulnerabilities Archive	www.securityfocus.com/bid
PacketStorm	packetstorm.nl
Computer Incident Advisory Capability	ciac.llnl.gov/ciac
Security Tracker	www.securitytracker.com

such as financial book close or product release cycles. These restrictions can greatly increase the time during which a system is vulnerable because a patch cannot be deployed. In some cases, the threat of a vulnerability being used to compromise a system will be so great that it will require that the patch be applied sooner than the regular maintenance window and will need to bypass these restrictions. To facilitate this, a rating system can be used to categorize vulnerability as to the probability of being exploited and the level of threat that it presents to the organization. The different ratings can be used to determine if a patch can wait for the normal maintenance process or if it should be applied in a timelier manner. In any case, patch installation should always go through the organization's formal change control management procedure.

There are a number of different factors that should be taken into account when rating an organization's vulnerabilities, including:

- *The vector or method of exploitability.* If the vulnerability is easily exploited by an available method of exploitability, then its risk will be far greater than one in which the vector is not readily available. An example of this would be a vulnerability that is exploitable over a network versus one that requires a login to the system. The vector for the network exploit is available to more potential threats than the one that requires local access and would increase the risk of the vulnerability being exploited. Network exploits can be further broken down into those that can be mitigated by firewall filtering and those that are allowed to pass through to internal networks, thereby greatly increasing the set of systems that are at risk. Examples of this pass-through would be exploits of Internet Web servers using HTTP traffic or an exploit of a DNS server using a specially crafted DNS request. These applications must be exposed to Internet traffic in order to provide their functionality. The risk for these applications to vulnerabilities exploitable via the network can be very high.
- *Length of time to apply a patch via normal change control processes.* Most production systems follow a change control process that involves testing and documenting changes before implementing them. This can take anywhere from a couple of hours to many days, depending on the complexity of the change. If a new vulnerability presents a sufficiently high threat to a system, this process may need to be shortened or expedited on an emergency basis. This may be especially true if an active exploit is moving through the network.
- *Availability of an exploit.* If an exploit for a particular vulnerability is available at a public information source, it can greatly increase the threat of a vulnerability being exploited. The public availability of an exploit brings it to the attention of more people, which increases the risk of its being used to compromise systems. These public information sources may include vulnerability development mailing lists and also Web sites that contain archives of exploits for different operating systems and platforms. It is important to note that it should not be assumed that if an exploit is not made public, it does not exist.
- *Criticality of the systems that are vulnerable.* Vulnerabilities to systems that are more important or to ones that provide infrastructure services may present more of a threat than vulnerabilities to less important systems. Critical systems can include infrastructure systems such as DNS servers and authentication servers, network devices such as routers and switches, as well as critical application servers. These systems may be important to an organization in maintaining its business and the risk to them needs to be kept to a minimum. There may also be cases where vulnerabilities are made public but the systems that are vulnerable are not critical to the business. This may mean that the vulnerability might not need to be mitigated as quickly.
- *Complexity of the patch.* Some patches are very simple to apply and others may require many changes to a system in order to be implemented correctly. The more complex the change to a system, the more likely that it will introduce unanticipated errors. This can be especially true of bundled patches or version upgrades. Trying the patches first in a test environment can help in mitigating this risk. However, there is not always time to perform this testing — especially during an outbreak of a worm or virus.

Using these factors, vulnerabilities and the subsequent patch can be rated in four different categories:

- *Normal.* This means that there is a low threat of the vulnerability being exploited and that the patch for it can be applied using the normal change control process and within the normal maintenance window. There can also be other mitigating factors that limit the attack vector, such as firewalls or other configuration changes that can be made to mitigate the vulnerability.
- *Urgent.* This means that there is a moderate threat of the vulnerability being exploited. This can include vulnerabilities that have been disclosed, but no exploit has been made available publicly and there are

EXHIBIT 58.3 Summary of Rating Categories and Actions

Rating	Risk Level	Level of Testing	Wait for Maintenance Window?
Normal	Low	Fully test	Yes
Urgent	Medium	Fully test	No
Critical	High	Minimal testing	No
Emergency	High/exploit in progress	None or very little	No

no reports of it being currently exploited. The patch for this vulnerability may go through the normal testing processes but may not wait for the normal change control maintenance window.

- *Critical.* This means that there is a high threat of the vulnerability being exploited. The attack vector of the exploit may be easily accessible and an exploit for the vulnerability is known and in the public domain. It may also be a threat to an infrastructure system. The patch for this vulnerability must be applied as soon as possible in order to mitigate this threat. It may still undergo some limited testing but will not wait for a normal change control window.
- *Emergency.* This means that the vulnerability is being actively exploited and needs to be patched immediately. Some examples of the use of this rating include cases of worms and viruses that are spreading through a network. The patch for this vulnerability is applied immediately, bypassing any testing or change management windows. This is done because the threat of the exploit is immediate and the risk of the change to the system is outweighed by the risk of the exploit.

These rating categories are summarized in Exhibit 58.3.

Some vendors have implemented their own rating systems to assist in communicating the level of threat that a vulnerability has for their product. It is important to understand the definitions of their different rating levels. Some of them adjust their ratings, depending on whether the system is connected to the Internet or not, and this may or may not apply to your organization. These definitions need to be evaluated and the rating system adjusted for your computing environment so that you can properly rate the vulnerabilities and the actions needed to mitigate them.

Process Overview

One of the first things that needs to be done to establish a patch management process is to institute a policy, as part of the organization's security policy, that software and systems need to be kept current and secure. This policy should be agreed to and communicated to all of the information technology system owners and administrators. Doing this up-front will make everyone aware that they are expected to apply patches in a timely manner. This policy will then drive the requirements for a patch management process. One other situation that needs to be addressed is the use of machines on the internal network that are not owned by the organization and not subject to its policies. This situation may introduce machines on the internal network that have not been patched and are in an unknown state. Many organizations will not allow these machines on the network because of that risk. However, outsourcing of an organization's functions and the use of contractors may make this policy difficult to implement.

The next step that needs to be taken in instituting a patch management process is to inventory the machines and software in your organization. Information that needs to be collected includes operating system type and version, installed applications along with their versions, contact information for the system administrator, and any other information needed to assess the risk from new vulnerabilities and patch the system. It is also important to inventory network devices such as printers and routers as these may also have vulnerabilities that will need to be patched. This inventory needs to be kept up-to-date and can be used to understand what threats are applicable to your environment by comparing new product vulnerabilities to the products that are present in your environment. In small organizations, this may be as simple as a spreadsheet that lists the various operating systems and application versions. In larger organizations, keeping these lists up-to-date becomes more of a challenge. There are software tools that will assist in this. These tools can be run periodically on systems to update a master database of software and version information. When a new vulnerability is made public, this information can be used to determine which systems in the environment are vulnerable and which systems need to have patches applied.

The actual patching mechanism can be a manual process initiated by a systems administrator, or it could be an automated system utilized by a central organization. An automated process can either be a "push" or a "pull" model. A "pull" model waits for a system to check for a patch from a distribution point. Systems will be configured to check at regular intervals. The advantage of a "pull" model is that new systems can be added with little effort, as the distribution point may not need to know anything about the clients it serves. A "push" model will initiate the patch from a central location. The advantage of a "push" system is that it has the ability to distribute a patch faster because it does not have to wait for a system to check for a patch. One disadvantage is that it requires more administration to add new machines as well as remove obsolete machines from the environment. Automated patching processes tend to be more effective when used for large numbers of end-user machines. End users do not tend to be aware of when they need to patch and may also not have the technical skills to apply the patches properly. Automation can ensure that the patches get applied in a timely manner because they can be controlled by a central organization. Automation also works best in environments that are standardized so that the machines are configured the same way. In more diverse environments, automated processes tend to become more complex because they require patching packages that take into account each separate variation of the environment.

Server administrators usually prefer manual patch processes because they prefer more control over the changes that are made to their servers. Applying patches presents a risk of breaking the applications that run on servers, and server applications tend to be more complicated than ones that run on end-user machines. Production servers will usually require back-out procedures in case the patch causes errors in applications. Automated patch processes may not be easily backed out because the automation may hide the actual changes being made to the server.

One other issue that needs to be addressed is the deployment of new machines into the infrastructure. As part of the deployment process, these machines will need to be patched to the appropriate level before being utilized. It is also important that they be integrated into the patch management inventory process to ensure that they receive future patches. Failure to do this will slowly erode the effectiveness of the patch management process.

Patch Management and Incident Response

Patching systems is a key part of responding to virus and worm incidents. Updating anti-virus software or applying fixes to systems may be the only way to halt the spread of a network worm or e-mail virus. Having an automated patch system can greatly decrease the amount of time it takes to patch large numbers of systems and mitigate these threats. However, provisions still need to be made for patching systems without the automated systems. The automated process can be disabled by the worm or virus, or the nature of the incident may require that the system be taken off the network and patched before being put back on the network. This was evident during the Code Red worm, in which systems were rebooted to clean the worm out of memory but, if left connected to the network, were almost immediately re-infected. This meant that network-based services, like file shares or Web sites that contained the patch, could not be used. So instead of using an automated patching system, a manual method using removable media had to be employed. This greatly increased the time needed to recover from this worm. If automated patching processes are used, care should be taken to protect them because, if these systems become incapacitated during an incident, then they cannot be used to help resolve it.

Compliance

Enforcing proper patch management is an important part of the process. This is especially true if non-automated methods are utilized. Periodic checks for compliance should be utilized to measure the effectiveness of the process. These checks can include:

- *Periodic system audits.* Manually checking your information systems can uncover systems that are not in compliance. This checking can be more complete than network scanning but is more personnel resource intensive.
- *Network vulnerability scanning.* This method is useful in checking large numbers of systems without much effort. This can be done on a regular basis and the information used to determine which systems

have been patched and which have not. One drawback is that this method will not be able to check for vulnerabilities that are not detectable over the network.

- *Login scripts.* Some network operating systems support login scripts that run when they authenticate to network resources. These scripts can be used to check the system compliance with patch management policy and have the benefit of being run every time a system authenticates. This check can also optionally deny access until the system is patched to the proper level. This has the benefit of forcing systems into compliance, or else they will not be able to access network resources. This can be especially useful when there is an outbreak of a new worm or virus. Systems with the particular vulnerability being exploited can be denied access until they are patched.

Compliance processes are very important when dealing with remote access users — especially those that utilize virtual private network (VPN) access via the Internet. Because these machines are on the Internet, they may be exposed to more threats than machines that are on an internal network behind a firewall. While connected to the Internet, machines may become infected by a worm or a virus and when they then connect to the internal network, they may carry this infection to internal systems. Having a process that checks the patch levels of these systems before allowing them access to internal network resources can greatly reduce the threat from such machines.

Exceptions

Systems may not always be capable of being patched to a secure level. One reason may be because the software vendor may no longer support the software. Most vendors will support software only for a certain length of time and then they will drop it from their supported software list. Having unsupported software in an organization can be very risky, as any new vulnerability will not be mitigated. Most organizations will transition to the newer versions of the software before this happens. In some cases, they will not be able to do this. The new version may require new hardware or it may not include necessary features. When this happens, the organization needs to understand the risk and balance it against the needs of the business as well as consider the cost of moving to a supported system. Other controls or precautions can be put in place to help mitigate the risk. These include additional network controls through the use of firewalls or host-based intrusion detection, which can detect configuration changes. Anti-virus software may also provide some measure of protection. It is also important to continually evaluate newly discovered vulnerabilities. If a high-risk vulnerability is found, it may be necessary to reevaluate the need for the system and to either adjust the controls or decommission the system. Systems that are exempt from patching can represent a significant risk to an organization and need to be managed appropriately.

Conclusion

Patch management is one of the key pieces of securing your information technology infrastructure. Patching against known vulnerabilities can reduce the known threats against your information systems. It provides a proactive way to reduce risk and can make a difference in ensuring the integrity and availability of your information systems. It is a straightforward way to make your systems more secure and reduce the threat to your information technology infrastructure.

Purposes of Information Security Management

Harold F. Tipton

Managing computer and network security programs has become an increasingly difficult and challenging job. Dramatic advances in computing and communications technology during the past five years have redirected the focus of data processing from the computing center to the terminals in individual offices and homes. The result is that managers must now monitor security on a more widely dispersed level. These changes are continuing to accelerate, making the security manager's job increasingly difficult.

The information security manager must establish and maintain a security program that ensures three requirements: the confidentiality, integrity, and availability of the company's information resources. Some security experts argue that two other requirements may be added to these three: utility and authenticity (i.e., accuracy). In this discussion, however, the usefulness and authenticity of information are addressed within the context of the three basic requirements of security management.

CONFIDENTIALITY

Confidentiality is the protection of information in the system so that unauthorized persons cannot access it. Many believe this type of protection is of most importance to military and government organizations that need to keep plans and capabilities secret from potential enemies. However, it can also be significant to businesses that need to protect proprietary trade secrets from competitors or prevent unauthorized persons from accessing the company's sensitive information (e.g., legal, personnel, or medical information). Privacy issues, which have received an increasing amount of attention in the past few years, place the importance of confidentiality on protecting personal information maintained in automated systems by both government agencies and private-sector organizations.

Confidentiality must be well defined, and procedures for maintaining confidentiality must be carefully implemented, especially for standalone computers. A crucial aspect of confidentiality is user identification and authentication. Positive identification of each system user is essential to ensuring the effectiveness of policies that specify who is allowed access to which data items.

Threats to Confidentiality

Confidentiality can be compromised in several ways. The following are some of the most commonly encountered threats to information confidentiality:

- Hackers.
- Masqueraders.
- Unauthorized user activity.
- Unprotected downloaded files.
- Local area networks (LANs).
- Trojan horses.

Hackers. A hacker is someone who bypasses the system's access controls by taking advantage of security weaknesses that the systems developers have left in the system. In addition, many hackers are adept at discovering the passwords of authorized users who fail to choose passwords that are difficult to guess or not included in the dictionary. The activities of hackers represent serious threats to the confidentiality of information in computer systems. Many hackers have created copies of inadequately protected files and placed them in areas of the system where they can be accessed by unauthorized persons.

Masqueraders. A masquerader is an authorized user of the system who has obtained the password of another user and thus gains access to files available to the other user. Masqueraders are often able to read and copy confidential files. Masquerading is a common occurrence in companies that allow users to share passwords.

Unauthorized User Activity. This type of activity occurs when authorized system users gain access to files that they are not authorized to access. Weak access controls often enable unauthorized access, which can compromise confidential files.

Unprotected Downloaded Files. Downloading can compromise confidential information if, in the process, files are moved from the secure environment of a host computer to an unprotected microcomputer for local processing. While on the microcomputer, unattended confidential information could be accessed by authorized users.

Local Area Networks. LANs present a special confidentiality threat because data flowing through a LAN can be viewed at any node of the network, whether or not the data is addressed to that node. This is particularly significant because the unencrypted user IDs and secret passwords of users logging on to the host are subject to compromise as this data travels from the user's node through the LAN to the host. Any confidential information not intended for viewing at every node should be protected by encryption.

Trojan Horses. Trojan horses can be programmed to copy confidential files to unprotected areas of the system when they are unknowingly executed by users who have authorized access to those files. Once executed, the Trojan horse becomes resident on the user's system and can routinely copy confidential files to unprotected resources.

Confidentiality Models

Confidentiality models are used to describe what actions must be taken to ensure the confidentiality of information. These models can specify how security tools are used to achieve the desired level of confidentiality.

The most commonly used model for describing the enforcement of confidentiality is the Bell-LaPadula model. It defines the relationships between objects (i.e., the files, records, programs, and equipment that contain or receive information) and subjects (i.e., the persons, processes, or devices that cause information to flow between the objects). The relationships are described in terms of the subject's assigned level of access or privilege and the object's level of sensitivity. In military terms, these would be described as the security clearance of the subject and security classification of the object.

Subjects access objects to read, write, or read and write information. The Bell-LaPadula model enforces the lattice principle, which specifies that subjects are allowed write access to objects at the same or higher level as the subject, read access to objects at the same or lower level, and read/write access to only those objects at the same level as the subject. This prevents the ability to write higher-classified information into a lower-classified file or to disclose higher-classified information to a lower-classified individual. Because an object's level indicates the security level of data it contains, all the data within a single object must be at the same level. This type of model is called flow model, because it ensures that information at a given security level flows only to an equal or higher level.

Another type of model that is commonly used is the access control model, which organizes a system into objects (i.e., resources being acted on), subjects (i.e., the persons or programs doing the action), and operations (i.e., the process of the interaction). A set of rules specifies which

operations can be performed on an object by which subjects. This type of model has the additional benefit of ensuring the integrity of information as well as the confidentiality; the flow model supports only confidentiality.

Implementing Confidentiality Models

The trusted system criteria provide the best guidelines for implementing confidentiality models. These criteria were developed by the National Computer Security Center and are published in the *Department of Defense Trusted Computer System Evaluation Criteria* (commonly referred to as the Orange Book), which discusses information confidentiality in considerable detail. In addition, the National Computer Security Center has developed a Trusted Network Interpretation that applies the Orange Book criteria to networks; the network interpretation is described in the *Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria* (commonly referred to as the Red Book).

INTEGRITY

Integrity is the protection of system data from intentional or accidental unauthorized changes. The challenge of the security program is to ensure that data is maintained in the state that users expect. Although the security program cannot improve the accuracy of data that is put into the system by users, it can help ensure that any changes are intended and correctly applied.

An additional element of integrity is the need to protect the process or program used to manipulate the data from unauthorized modification. A critical requirement of both commercial and government data processing is to ensure the integrity of data to prevent fraud and errors. It is imperative, therefore, that no user be able to modify data in a way that might corrupt or lose assets or financial records or render decision-making information unreliable. Examples of government systems in which integrity is crucial include air traffic control systems, military fire control systems (which control the firing of automated weapons), and Social Security and welfare systems. Examples of commercial systems that require a high level of integrity include medical prescription systems, credit reporting systems, production control systems, and payroll systems.

As with the confidentiality policy, identification and authentication of users are key elements of the information integrity policy. Integrity depends on access controls; therefore, it is necessary to positively and uniquely identify all persons who attempt access.

Protecting Against Threats to Integrity

Like confidentiality, integrity can be compromised by hackers, masqueraders, unauthorized user activity, unprotected downloaded files, LANs, and unauthorized programs (e.g., Trojan horses and viruses), because

each of these threats can lead to unauthorized changes to data or programs. For example, authorized users can corrupt data and programs accidentally or intentionally if their activities on the system are not properly controlled.

Three basic principles are used to establish integrity controls:

1. granting access on a need-to-know basis,
2. separation of duties,
3. rotation of duties.

Need-to-Know Access. Users should be granted access only to those files and programs that they need in order to perform their assigned job functions. User access to production data or source code should be further restricted through use of well-formed transactions, which ensure that users can change data only in controlled ways that maintain the integrity of data. A common element of well-formed transactions is the recording of data modifications in a log that can be reviewed later to ensure that only authorized and correct changes were made. To be effective, well-formed transactions must ensure that data can be manipulated only by a specific set of programs. These programs must be inspected for proper construction, installation, and controls to prevent unauthorized modification.

Because users must be able to work efficiently, access privileges should be judiciously granted to allow sufficient operational flexibility; need-to-know access should enable maximum control with minimum restrictions on users. The security program must employ a careful balance between ideal security and practical productivity.

Separation of Duties. To ensure that no single employee has control of a transaction from beginning to end, two or more people should be responsible for performing it — for example, anyone allowed to create or certify a well-formed transaction should not be allowed to execute it. Thus, a transaction cannot be manipulated for personal gain unless all persons responsible for it participate.

Rotation of Duties. Job assignments should be changed periodically so that it is more difficult for users to collaborate to exercise complete control of a transaction and subvert it for fraudulent purposes. This principle is effective when used in conjunction with a separation of duties. Problems in effectively rotating duties usually appear in organizations with limited staff resources and inadequate training programs.

Integrity Models

Integrity models are used to describe what needs to be done to enforce the information integrity policy. There are three goals of integrity, which the models address in various ways:

1. Preventing unauthorized users from making modifications to data or programs.
2. Preventing authorized users from making improper or unauthorized modifications.
3. Maintaining internal and external consistency of data and programs.

The first step in creating an integrity model for a system is to identify and label those data items for which integrity must be ensured. Two procedures are then applied to these data items. The first procedure verifies that the data items are in a valid state (i.e., they are what the users or owners believe them to be because they have not been changed). The second procedure is the transformation procedure or well-formed transaction, which changes the data items from one valid state to another. If only a transformation procedure is able to change data items, the integrity of the data is maintained. Integrity enforcement systems usually require that all transformation procedures be logged, to provide an audit trail of data item changes.

Another aspect of preserving integrity relates to the system itself rather than only the data items in the system. The system must perform consistently and reliably — that is, it must always do what the users or owners expect it to do.

National Computer Security Center Report 79–91, "Integrity in Automated Information Systems" (September 1991), discusses several integrity models. Included are five models that suggest different approaches to achieving integrity:

1. Biba,
2. Goguen-Meseguer,
3. Sutherland,
4. Clark-Wilson,
5. Brewer-Nash.

The Biba Model. The first model to address integrity in computer systems was based on a hierarchical lattice of integrity levels defined by Biba in 1977. The Biba integrity model is similar to the Bell-LaPadula model for confidentiality in that it uses subjects and objects; in addition, it controls object modification in the same way that Bell-LaPadula controls disclosure.

Biba's integrity policy consists of three parts. The first part specifies that a subject cannot execute objects that have a lower level of integrity than the subject. The second part specifies that a subject cannot modify objects that have a higher level of integrity. The third part specifies that a subject may not request service from subjects that have a higher integrity level.

The Goguen-Meseguer Model. The Goguen-Meseguer model, published in 1982, is based on the mathematical principle governing automatons (i.e., a control mechanism designed to automatically follow a predetermined sequence of operations or respond to encoded instructions) and includes domain separation. In this context, a domain is the list of objects that a user can access; users can be grouped according to their defined domains. Separating users into different domains ensures that users cannot interfere with each other's activities. All the information about which activities users are allowed to perform is included in a capabilities table.

In addition, the system contains information not related to permissions (e.g., user programs, data, and messages). The combination of all this information is called the state of the system. The automaton theory used as a basis for this model predefines all of the states and transitions between states, which prevents unauthorized users from making modifications to data or programs.

The Sutherland Model. The Sutherland model, published in 1986, approaches integrity by focusing on the problem of inference (i.e., the use of covert channels to influence the results of a process). This model is based on a state machine and consists of a set of states, a set of possible initial states, and a transformation function that maps states from the initial state to the current state.

Although the Sutherland model does not directly invoke a protection mechanism, it contains access restrictions related to subjects and information flow restrictions between objects. Therefore, it prevents unauthorized users from modifying data or programs.

The Clark-Wilson Model. The Clark-Wilson model, published in 1987 and updated in 1989, involves two primary elements for achieving data integrity — the well-formed transaction and separation of duties. Well-formed transactions, as previously mentioned, prevent users from manipulating data, thus ensuring the internal consistency of data. Separation of duties prevents authorized users from making improper modifications, thus preserving the external consistency of data by ensuring that data in the system reflects the real-world data it represents.

The Clark-Wilson model differs from the other models that are subject and object oriented by introducing a third access element — programs — resulting in what is called an access triple, which prevents unauthorized users from modifying data or programs. In addition, this model uses integrity verification and transformation procedures to maintain internal and external consistency of data. The verification procedures confirm that the data conforms to the integrity specifications at the time the verification is performed. The transformation procedures are designed to take the system

from one valid state to the next. The Clark-Wilson model is believed to address all three goals of integrity.

The Brewer-Nash Model. The Brewer-Nash model, published in 1989, uses basic mathematical theory to implement dynamically changing access authorizations. This model can provide integrity in an integrated data base. In addition, it can provide confidentiality of information if the integrated data base is shared by competing companies; subjects can access only those objects that do not conflict with standards of fair competition.

Implementation involves grouping data sets into discrete classes, each class representing a different conflict of interest (e.g., classified information about a company is not made available to a competitor). Assuming that a subject initially accesses a data set in each of the classes, the subject would be prevented from accessing any other data set in each class. This isolation of data sets within a class provides the capability to keep one company's data separate from a competitor's in an integrated data base, thus preventing authorized users from making improper modifications to data outside their purview.

Implementing Integrity Models

The integrity models may be implemented in various ways to provide the integrity protection specified in the security policy. National Computer Security Center Report 79–91 discusses several implementations, including those by Lipner, Boebert and Kain, Lee and Shockley, Karger, Jueneman, and Gong. These six implementations are discussed in the following sections.

The Lipner Implementation. The Lipner implementation, published in 1982, describes two ways of implementing integrity. One uses the Bell-LaPadula confidentiality model, and the other uses both the Bell-LaPadula model and the Biba integrity model. Both methods assign security levels and functional categories to subjects and objects. For subjects, this translates into a person's clearance level and job function (e.g., user, operator, applications programmer, or systems programmer). For objects, the sensitivity of the data or program and its functions (e.g., test data, production data, application program, or system program) are defined.

Lipner's first method, using only Bell-LaPadula model, assigns subjects to one of two sensitivity levels — system manager and anyone else — and to one of four job categories. Objects (i.e., file types) are assigned specific levels and categories. Most of the subjects and objects are assigned the same level; therefore, categories become the most significant integrity (i.e., access control) mechanism. The applications programmers, systems programmers, and users are confined to their own domains according to their

assigned categories, thus preventing unauthorized users from modifying data.

Lipner's second method combines Biba's integrity model with the Bell-LaPadula basic security implementation. This combination of models helps prevent contamination of high-integrity data by low-integrity data or programs. The assignment of levels and categories to subjects and objects remains the same as for Lipner's first method. Integrity levels are used to avoid the unauthorized modification of system programs; integrity categories are used to separate domains that are based on functional areas (e.g., production or research and development). This method prevents unauthorized users from modifying data and prevents authorized users from making improper data modifications.

Lipner's methods were the first to separate objects into data and programs. The importance of this concept becomes clear when viewed in terms of implementing the Clark-Wilson integrity model; because programs allow users to manipulate data, it is necessary to control which programs a user may access and which objects a program can manipulate.

The Boebert and Kain Implementations. Boebert and Kain independently proposed (in 1985 and 1988, respectively) implementations of the Goguen-Meseguer integrity model. This implementation uses a subsystem that cannot be bypassed; the actions performed on this subsystem cannot be undone and must be correct. This type of subsystem is featured in the system's logical coprocessor kernel, which checks every access attempt to ensure that the access is consistent with the security policy being invoked.

Three security attributes are related to subjects and objects in this implementation. First, subjects and objects are assigned sensitivity levels. Second, subjects are identified according to the user in whose behalf the subject is acting, and objects are identified according to the list of users who can access the object and the access rights users can execute. Third, the domain (i.e., subsystem) that the program is a part of is defined for subjects, and the object type is defined according to the information contained within the object.

When the system must determine the kind of access a subject is allowed, all three of these security attributes are used. Sensitivity levels of subjects and objects are compared to enforce the mandatory access control policy. To enforce discretionary access control, the access control lists are checked. Finally, access rights are determined by comparing the subject domain with the object type.

By isolating the action rather than the user, the Boebert and Kain implementation ensures that unauthorized users cannot modify data. The use of

domains requires that actions be performed in only one location and in only one way; a user who cannot access the domain cannot perform the action.

The Lee and Shockley Implementations. In 1988, Lee and Shockley independently developed implementations of the Clark-Wilson integrity model using Biba's integrity categories and trusted subjects. Both of these implementations were based on sensitivity levels constructed from independent elements. Each level represents a sensitivity to disclosure and a sensitivity to modification.

Data is manipulated by certified transactions, which are trusted subjects. The trusted subject can transform data from a specific input type to a specific output type. The Biba lattice philosophy is implemented so that a subject may not read above its level in disclosure or below its level in integrity. Every subject and object has both disclosure and integrity levels for use in this implementation. The Lee and Shockley implementations prevent unauthorized users from modifying data.

The Karger Implementation. In 1988, Karger proposed another implementation of the Clark-Wilson integrity model, augmenting it with his secure capabilities architecture (developed in 1984) and a generic lattice security model. In this implementation, audit trails play a much more prominent part in the enforcement of security than in other implementations. The capabilities architecture combined with access control lists that represent the security lattice provide for improved flexibility in implementing integrity.

In addition, the Karger implementation requires that the access control lists contain the specifics of the Clark-Wilson triples (i.e., the names of the subjects and objects the user is requesting access to and the names of the programs that provide the access), thereby enabling implementation of static separation of duties. Static separation of duties prevents unauthorized users from modifying data and prevents authorized users from making improper modifications.

The part of Karger's implementation that uses capabilities with access control lists limits actions to particular domains. The complex access control lists not only contain the triples but specify the order in which the transactions must be executed. These lists are used with audit-based capabilities to enforce dynamic separation of duties.

The Karger implementation provides three levels of integrity protection. First, triples in the access control lists allow for basic integrity (i.e., static separation of duties). Second, the capabilities architecture can be used with access control lists to provide faster access and domain separation. Third, access control lists and the capabilities architecture support both dynamic separation of duties and well-formed transactions.

The Jueneman Implementation. In 1989, Jueneman proposed a defensive detection implementation for use on dynamic networks of interconnected trusted computers communicating through unsecured media. This implementation was based on mandatory and discretionary access controls, encryption, checksums, and digital signatures. It prevents unauthorized users from modifying data.

The control mechanisms in this implementation support the philosophy that the originator of an object is responsible for its confidentiality and that the recipient is responsible for its integrity in a network environment. The mandatory access controls prevent unauthorized modification within the trusted computers and detect modifications external to the trusted computers. The discretionary access controls prevent the modification, destruction, or renaming of an object by a user who qualifies under mandatory control but lacks the owner's permission to access the object. The encryption mechanism is used to avoid unauthorized disclosure of the object. The encryption mechanism is used to avoid unauthorized disclosure of the object. Checksums verify that the communication received is the communication that was sent, and digital signatures are evidence of the source of the communication.

The Gong Implementation. The Gong implementation, developed in 1989, is an identity-based and capability-oriented security system for distributed systems in a network environment. Capabilities identify each object and specify the access rights (i.e., read, write and update) to be allowed each subject that is authorized access. Access authorizations are provided in an access list.

The Gong implementation consists of subjects (i.e., users), objects, object servers, and a centralized access control server. The access control server contains the access control lists, and the object server contains the capability controls for each object.

This implementation is very flexible because it is independent of the protection policy (i.e., the Bell-LaPadula disclosure lattice, the Biba integrity lattice, the Clark-Wilson access triples, or the Lee-Shockley nonhierarchical categories). The Gong implementation can be used to prevent unauthorized users from modifying data and to prevent authorized users from making unauthorized modifications.

AVAILABILITY

Availability is the assurance that a computer system is accessible by authorized users whenever needed. Two facets of availability are typically discussed:

1. Denial of service.
2. Loss of data processing capabilities as a result of natural disasters (e.g., fires, floods, storms, or earthquakes) or human actions (e.g., bombs or strikes).

Denial of service usually refers to actions that tie up computing services in a way that renders the system unusable by authorized users. For example, the Internet worm overloaded about 10% of the computer systems on the network, causing them to be nonresponsive to the needs of users.

The loss of data processing capabilities as a result of natural disasters or human actions is perhaps more common. Such losses are countered by contingency planning, which helps minimize the time that a data processing capability remains unavailable. Contingency planning — which may involve business resumption planning, alternative-site processing, or simply disaster recovery planning — provides an alternative means of processing, thereby ensuring availability.

Physical, technical, and administrative issues are important aspects of security initiatives that address availability. The physical issues include access controls that prevent unauthorized persons from coming into contact with computing resources, various fire and water control mechanisms, hot and cold sites for use in alternative-site processing, and off-site backup storage facilities. The technical issues include fault-tolerance mechanisms (e.g., hardware redundancy, disk mirroring, and application checkpoint restart), electronic vaulting (i.e., automatic backup to a secure, off-site location), and access control software to prevent unauthorized users from disrupting services. The administrative issues include access control policies, operating procedures, contingency planning, and user training. Although not obviously an important initiative, adequate training of operators, programmers, and security personnel can help avoid many computing stages that result in the loss of availability. In addition, availability can be restricted if a security office accidentally locks up an access control data base during routine maintenance, thus preventing authorized users access for an extended period of time.

Considerable effort is being devoted to addressing various aspects of availability. For example, significant research has focused on achieving more fault-tolerant computing. Another sign that availability is a primary concern is that increasing investments are being made in disaster recovery planning combined with alternative-site processing facilities. Investments in antiviral products are escalating as well; denial of service associated with computer viruses, Trojan horses, and logic bombs is one of today's major security problems.

Known threats to availability can be expected to continue. New threats may emerge as technology evolves, making it quicker and easier for users to share information resources with other users, often at remote locations.

SUMMARY

The three basic purposes of security management — integrity, confidentiality, and availability — are present in all systems. Whether a system emphasizes one or the other of these purposes depends on the functions performed by the applications. For example, air traffic control systems do not require a high level of information confidentiality; however, a high degree of integrity is crucial to avoid disastrous misguiding of aircraft, and availability is important to avoid disruption of air traffic services.

Automobile companies, on the other hand, often go to extreme lengths to protect the confidentiality of new designs, whereas integrity and availability are of lesser concern. Military weapons systems also must have a high level of confidentiality to avoid enemy compromise. In addition, they must provide high levels of integrity (to ensure reliability) and availability (to ensure that the system operates as expected when needed).

Historically, confidentiality has received the most attention, probably because of its importance in military and government applications. As a result, capabilities to provide confidentiality in computer systems are considerably more advanced than those providing integrity or availability. Significant research efforts have recently been focused on the integrity issue. Still, little attention has been paid to availability, with the exception of building fault tolerance into vendor products and including hot and cold sites for backup processing in disaster recovery planning.

The combination of integrity, availability, and confidentiality in appropriate proportions to support the organization's goals can provide users with a trustworthy system — that is, users can trust it will consistently perform according to their expectations. Trustworthiness has a broader definition than security in that it combines security with safety and reliability as well as the protection of privacy (which is already considered to be a part of security). In addition, many of the mechanisms that provide security also make systems more trustworthy in general. These multipurpose safeguards should be exploited to the extent practicable.

The Building Blocks of Information Security

Ken M. Shaurette

INFORMATION SECURITY IS NOT JUST ABOUT TECHNOLOGICAL CONTROLS. SECURITY CANNOT BE ACHIEVED SOLELY THROUGH THE APPLICATION OF SOFTWARE OR HARDWARE. Any attempt to implement technology controls without considering the cultural and social attitudes of the corporation is a formula for disaster. The best approach to effective security is a layered approach that encompasses both technological and nontechnological safeguards. Ideally, these safeguards should be used to achieve an acceptable level of protection while enhancing business productivity. While the concept may sound simple, the challenge is to strike a balance between being too restrictive (overly cautious) or too open (not cautious enough).

Security technology alone cannot eliminate all exposures. Security managers must integrate themselves with existing corporate support systems. Together with their peers, they will develop the security policies, standards, procedures, and guidelines that form the foundation for security activities. This approach will ensure that security becomes a function of the corporation — not an obstacle to business.

A successful layered approach must look at all aspects of security. A layered approach concentrating on technology alone becomes like a house of cards. Without a foundation based on solid policies, the security infrastructure is just cards standing side by side, with each technology becoming a separate card in the house. Adding an extra card (technology layer) to the house (overall security) does not necessarily make the house stronger.

Without security policies, standards, procedures, and guidelines, there is no general security framework or foundation. Policies define the behavior that is allowed or not allowed. They are short because they do not explain how to achieve compliance; such is the purpose of procedures and

guidelines. Corporate policy seldom changes because it does not tie to technology, people, or specific processes. Policy establishes technology selection and how it will be configured and implemented. Policies are the consensus between people, especially important between all layers of corporate management. Policy can ensure that the Security Manager and his or her peers apply security technology with the proper emphasis and return on investment for the good of the business as a whole.

In most security audits or reviews, checking, maybe even testing, an organization's security policies, standards, procedures, and guidelines is often listed as the first element in assessing security risk. It is easy to see the published hard-copy policy; but to ensure that policy is practiced, it is necessary to observe the workplace in order to evaluate what is really in operation. Lack of general awareness or compliance with a security policy usually indicates a policy that was not developed with the participation of other company management.

Whether the organization is global or local, there is still expectation of levels of due diligence. As a spin on the golden rule: "Compute unto others as you would want them to compute unto you."

Define the Scope: Objective

> "The first duty of a human is to assume the right functional relationship to society — more briefly, to find your real job, and do it."
>
> — Charlotte Perkins Gilman

Define Security Domain

Every organization has a different perspective on what is within the domain of its Information Security department.

- Does the Information Security domain include both electronic and non-electronic information, printed versus the bytes stored on a computer?
- Does the Information Security department report to IS and have responsibility for only information policies, not telephone, copier, fax, and mail use?
- Does physical security and contingency planning fall into the Information Security Manager's domain?
- Is the Security Manager's responsibility corporate, regional, national, or global?

Information Security's mission statement must support the corporation's business objective. Very often, one can find a security mission stated something like:

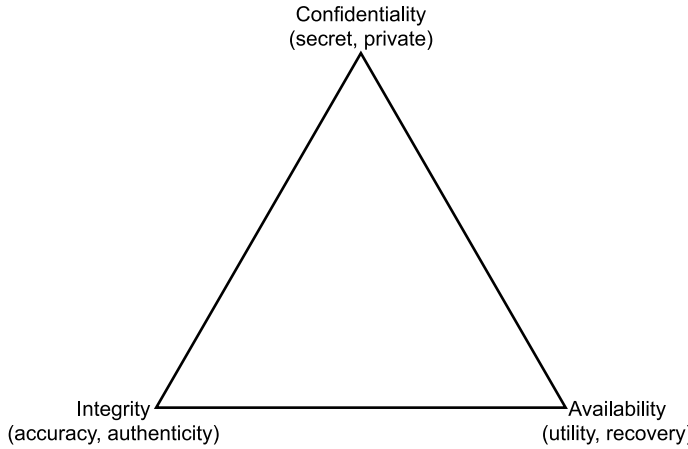


Exhibit 10-1. Basic security triad.

> The mission of the Information Security department is to protect the information assets, the information systems, and networks that deliver the information, from damage resulting from failures of confidentiality, integrity, and availability (CIA) (see Exhibit 10-1).

This mission is quite specific to Information Security and a specific department. A mission like this is a prime reason why defining the Security Manager's domain is critical to the success of policy formation.

Would the mission be more positive and clear by being tied to the business objectives with something like:

> Security's objective is to enhance the productivity of the business by reducing probability of loss through the design and implementation of policies, standards, procedures, and guidelines that enhance the protection of business assets.

Notice how this mission statement does not limit itself to "information." It does not limit the responsibilities to only computer systems and their processing of information. In addition, it ties the success of the mission to the business. It still provides the flexibility to define assets and assign owners to them for accountability. It is important to understand the objectives that security is going to deliver for the business. Exhibit 10-2 outlines some sample objectives.

What will be in the Security Manager's domain: physical security, contingency planning, telephones, copiers, faxes, or mail (especially e-mail)? These technologies process information too, so would they be covered by Information Security Policy? How far reaching will the Security Manager's responsibilities be: corporate, global, national, regional, or local? Is it the

Exhibit 10-2. Questions to help determine security philosophy.

- Do users have expectations relating to security?
- Is it possible to lose customers if security is too restrictive, not restrictive enough, or if controls and policy are so unreasonable that functionality is impaired?
- Is there a history for lost availability or monetary loss from security incidents in the past? What was the cost to the business?
- Who is the primary enemy — employees or outsiders?
- How much confidential information is online, and how is it accessed? What would be the loss if the information was compromised or stolen?
- Is it important to layer security controls for different parts of the organization?
- Are dangerous services that increase vulnerabilities supported by the organization? Is it required that networks and systems meet a security baseline?
- What security guidelines, procedures, regulations, or laws must be met?
- Is there a conflict between business objectives and security?
- Confidentiality, integrity, and availability: how crucial is each to the overall operation?
- Consider business needs and economic reality. What meets due diligence for like companies, the security industry, for this information in other environments?

Security Manager's responsibility to enforce compliance? Is contingency planning or business continuity planning (BCP) a function of physical security? Once the domain has been clearly defined, it becomes easy for responsible areas to form and begin to create their specific policies, standards, procedures, and guidelines.

Traditionally, organizations would refer to different departments for the responsibility of security on such things as telephones, copiers, faxes, or mail. An organization would have to climb quite high in the organizational structure — executive VP, COO, CEO — to find the common management point in the organizational structure where a person responsible for the security of all the disparate resources would come together for central accountability.

Hint: Policies written with the term "electronic" can cover e-mail, (electronic mail), EDI (electronic data interchange), or all the other "E-words" that are becoming popular (i.e., E-commerce, E-marketing, and E-business). Policies not using the term "electronic" can refer to information regardless of technology, storage media, or transportation methods.

In that regard, what used to be called datasecurity, today is referred to as information security. Information security often considers the security of data, information in both electronic and non-electronic forms. The role of the Information Security Manager has either expanded or information security personnel have begun assuming responsibilities in areas that are

often not clearly defined. Some organizations are recognizing the difficulty of separating information dealing with technology from non-technology. With that in mind, Corporate Security Officer (CSO) type positions are being created (other possible name: Chief Security Officer). These positions can be scoped to have responsibility for security, regardless of technology, and across the entire enterprise regardless of geography. This would not necessarily mean that all of the impacted areas report to this position, but this position would provide the enterprise or corporate vision of information security. It would coordinate the security accomplishments for the good of the entire organization, crossing domains and departments. Define "information"; what does it not include?

For years, security purists have argued for information security to report high in the organization as well as not necessarily within the information services (IS) division. Some organizations accomplished this by creating executive-level security positions reporting to the president, COO, or CEO. In differing ways, more organizations are finally making strides to at least put the "corporate" or "enterprise" spin on addressing the security issues of the organization, not just the issues (policy) of IS. An appointment of security personnel with accountability across the organization is a start. Giving them top management and line management support across the organization remains critical to their success, regardless of how high they report in the organization. An executive VP of information security will fail if the position is only a token position. On the other hand, the flunky of information security can be successful if everyone from top down is behind him and the concept of corporate information security.

In this structure, traditional areas can remain responsible for their parts of security and policy definition, their cards in the house, but a corporate entity coordinates the security efforts and brings it all together. That corporate entity is tasked with providing the corporate security vision and could report high in the organization, which is probably the best, or it could be assigned corporate responsibility by executive management. Total and very visual support by all management is obviously critical for success.

Sample roles and responsibilities for this structure include:

- The protection and safety department would continue to contract for guards, handle building access control, ID cards, and other physical building controls, including computer rooms.
- The telecommunications department is still be accountable for the security of phone systems and helps with establishment of policy addressing phone-mail and use of company telephones, probably including fax.
- A corporate mail department deals with internal and external mail, possibly including e-mail.

- IS has accountability for computer-based information processing systems and assists with the establishment of standards for use of them or policy dealing with information processing.
- The corporate legal department would help to ensure that policy meets regulations from a legal perspective and that proper wording makes them enforceable.
- A corporate compliance department can insure that regulatory and legislative concerns are addressed, such as the federal sentencing guidelines.
- Human resources (HR) is still a critical area in identifying employee policies and works closely with the Corporate Security Officer (CSO) on all policies, standards, procedures, and guidelines, as well as proper enforcement.
- The CSO works with all areas to provide high-level security expertise, coordinate and establish employee security awareness, security education programs, along with publication and communication of the security policies, standards, procedures, and guidelines.

SECURITY PHILOSOPHY

> No gain is possible without attendant outlay, but there will be no profit if the outlay exceeds the receipts.
>
> — Plautus

Return on Investment (ROI): What is the basis for security philosophy?

Security is often expected to provide a return on investment (ROI) to justify expenditures. How often is it possible for information security to generate a direct ROI? Which is more expensive, recover from an incident or prevent the incident in the first place? Computer security is often an intangible process. In many instances, the level of security is not evident until a catastrophe happens, at which time the lack of security is all too painfully evident.

Information security should be viewed in terms of the processes and goals of the business. Business risk is different from security risk, but poor security can put the business at risk, or make it risky doing business.

Example

- Would a wise company provide banking services, transmitting credit card numbers and account balances using an unsecured Internet connection? A properly secured infrastructure using encryption or certificates for nonrepudiation can provide the company with a business opportunity that it would not otherwise be likely to engage in. In that situation, the security is an integral part of that business opportunity, minimizing the business risk.

- How can a security manager justify control procedures over program changes or over developers with update access to production data? Assume that 20 percent of problems result from program errors or incorrect updates to data. Maybe inadequately tested code in a program is transferred to production. If controls can reduce the errors and resulting rework to say 10 percent, the payback would be only a few months. In a company that sells its programming services based on quality, this would directly relate to potential business opportunity and increased contracts.
- What about customer privacy? A Harris Poll showed that 53 percent of American adults are concerned about privacy threats from corporations. People have stated in surveys that they would rather do business with a company they feel is going to protect the privacy of their information. Increased business opportunity exists for the company that can show that it protects customer privacy better than its competition, even if it only generates the perception of better. Perception is 90 percent reality. Being able to show how the company enforces sound security policies, standards, and procedures would provide the business advantage.

  Although a mission statement may no longer refer directly to confidentiality, integrity, and availability, the security department cannot ignore CIA (see Exhibit 10-1). As discussed, the base security philosophy must now help improve business productivity. The real life situation is that we can never provide 100 percent security. We can, however, reduce the probability of loss or taking reasonable measures of due diligence consistent with industry norms for how like companies are dealing with like information. Going that extra step ahead to lead the industry can create business opportunity and minimize business risk.

To meet the security business objective, a better order for this triad is probably AIC, but that does not stir as much intrigue as CIA. Studies show AIC to be better matched to the order of priority for many security managers.

WHY?

- *Availability:* A corporation gathers endless amounts of information and in order to effectively produce product, that information must be available and usable when needed. This includes the concept of utility, or that the information must have the quality or condition of being useful. Just being available is not sufficient.
- *Integrity:* For the information to have any value and in order to produce quality product, the data must be protected against unauthorized or inadvertent modification. Its integrity must be of the highest quality and original. If the authenticity of the information is in doubt or compromised, the integrity is still jeopardized.

- *Confidentiality:* The privacy of customer information is becoming more and more important, if not to the corporation, to the customer. Legislation could one day mandate minimum protections for specific pieces of information like health records, credit card numbers, and bank account numbers. Ensuring that only the proper people have access to the information needed to perform their job or that they have been authorized to access it is often the last concern because it can impede business productivity.

MANAGEMENT MYTHS OF SECURITY

1. Security technology will solve all the problems.

Buy the software; now the company is secure. Management has signed the purchase order and the software has arrived. Is management's job finished and the company now secure? Management has done their due diligence, right? Wrong! Remember, software and security technologies are only a piece of the overall security program.

Management must have a concept or philosophy regarding how it wants to address information security, recognizing that technology and software are not 100 percent effective and are not going to magically eliminate all security problems. Does the security software restrict any access to a resource, provide everyone access, or just audit the access until someone steps forward with resources that need to be protected? The security job is not done once the software is installed or the technology is chosen.

Management support for proper security software implementation, configuration, continued maintenance, and the research and development of new security technologies is critical.

2. I have written the policy, so now we are done.

If policies or standards are written but never implemented, or not followed, not enforced, or enforced inconsistently it is worse than not having them at all. Federal Sentencing Guidelines require consistent application of policy and standards.

In an excerpt from the Federal Sentencing Guidelines, it states:

> The standards must have been consistently enforced through appropriate disciplinary mechanisms, including as appropriate, discipline of individuals responsible for the failure to detect an offense. Adequate discipline of individuals responsible for an offense is a necessary component of enforcement; however, the form of discipline that will be appropriate will be case specific.

Management must recognize that policy and standards implementation should be defined as a specific project receiving continued management

support. They may not have understood that there is a cost associated with implementing policy and thought this was only a policy development effort.

Strict enforcement of policy and standards must become a way of life in business. Corporate policy-making bodies should consider adherence to them a condition of employment. Never adopt a policy unless there is a good prospect that it will be followed. Make protecting the confidentiality, integrity, and availability of information "The Law."

3. Publish policy and standards and everyone will comply.

Not only is the job not done once the policy is written, but ensuring that every employee, customer, vendor, constituent, or stockholder knows and understands policy is essential. Training them and keeping records of the training on company policy are critical. Just publishing the policy does not encourage anyone to comply with it.

Simply training people or making them aware (security awareness) is also not sufficient; all one gets is shallow or superficial security. There needs to be motivation to carry out policy; only penalizing people for poor security does not always create positive motivation and is a militaristic attitude. Even child psychologists recommend positive reinforcement.

Security awareness alone can have a negative effect by teaching people how to avoid security in their work. Everyone knows it just slows them down, and they hate it anyway, especially if only penalties are associated with it. Positive reinforcement calls for rewards when people show actions and attitudes toward very good security. Do not eliminate penalties for poor security, but do not let them be the only motivator. Once rewards and penalties are identified, education can include how to achieve the rewards and avoid the penalties, just as for other work motivation. This requires an effectively applied security line item in salary and performance reviews and real rewards and penalties.

4. Follow the vendor's approach: it is the best way to make an organization secure.

An organization's goals should be to build the fences as high as it can. Protect everything; implement every feature of that new software. The organization has paid for those functions and the vendor must know the best way to implement them.

Often, an organization might be inclined to take a generic security product and fail to tailor it to fit its business objectives. Everyone can name an operating system that is not quite as secure as one would like it to be using the vendor defaults. The vendor's approach may go against organization

security philosophy. The product may come out of the box with limited security, open architecture, but the company security philosophy is to allow only access as appropriate, or vice versa.

Should one put all one's eggs in one basket or build one's house all from the same deck of cards? Does using only one security solution from a single vendor open vulnerability to the security architecture? Think about using the best-of-class solution from multiple vendors; this way, one's security architecture is not easily blueprinted by outsiders.

BUILDING THE BRIDGE: SECURITY CONTROLS REACH FOR BUSINESS NEEDS

An information security infrastructure is like a bridge built between the user with a business need to access information and at the other end of the bridge the information they wish to access. Creating gates between the end user and the data are the controls (technology) providing security protection or defining specific paths to the information. Forming the foundation for the security technology to be implemented are policies, standards, and procedures.

Guidelines are not required actions, but provide a map (suggestions of how to comply) or, like the railings of the bridge, help direct end users to their destination so they do not fall off the bridge. Just like the rails of a bridge, if the guidelines are not followed, it is still possible to fall off the bridge (not comply with policy and standards). The river represents unauthorized access, malicious elements (hackers), or unauthorized entities (disgruntled employees) that could affect the delivery of the payloads (information) across the bridge. The river (malicious access) is constantly flowing and often changing faster than security controls can be implemented. The security technology or software are locked gates, toll ways, or speed bumps on the bridge that control and audit the flow of traffic authorized to cross. Exposures or risks that have been accepted by management are represented by holes in the surface of the bridge that are not patched or are not covered by a security technology. Perhaps they are only covered with a see-through mesh, because ignorance is the only protection. The bigger the risk, the bigger the hole in the roadbed.

Build bridges that can get the organization from the "Wild Wild West" of the Internet to the future wars that are yet to be identified. William Hugh Murray of Deloitte and Touche once stated that one should build a solid infrastructure; the infrastructure should be a foundation that will last for 30 years. Work to build a bridge that will handle traffic for a long time and one will have the kind of infrastructure that can be depended upon for many years. Well-written and management-accepted policy should rarely change.

THE RIVER: UNDERSTANDING THE BUSINESS NEED

Understanding what one is protecting the business against is the first place to start. Too often, IS people will build a fantastic bridge — wide, double decked, all out of the best steel in the world — then they begin looking for a river to cross. This could also be called knowing the enemy or, in a more positive light to go with the business concept, understanding the business need.

If the Security Manager does not understand what objectives the end users of the information have, one will not know what is the best security philosophy to choose. One will not know whether availability is more important than integrity or confidentiality, nor which should get the primary focus. It will be difficult to leverage sufficient security technology with administrative procedures, policies, and standards. ROI will be impossible to guage. There will be no way of knowing what guidelines would help the end user follow policy or work best with the technology. Organizations often focus efforts on technical priorities that may not even be where the greatest exposures to the information are (see Exhibit 10-3). Problems for nonexistent exposures will be getting solved; a bridge will be getting erected across a dry river.

Exhibit 10-3. Case study: bank of the world savings.

CASE STUDY:

The Bank of the World Savings (BOWS) organization is dealing daily with financial information. BOWS has security technology fully implemented for protecting information from manipulation by unauthorized people and from people stealing credit card numbers, etc. to the best of its technical ability. Assuming this is equivalent to what all other banks do, BOWS has probably accomplished a portion of its due diligence.

Because no technology can provide 100 percent security, what happens if a person does get by the security technology? BOWS can be damaged just as severely by bad publicity as from the actual loss incurred by circumvention of the technology. Unless the bank has created procedures and policies for damage control, its loss could be orders of magnitude larger in lost business than the original loss.

BOWS does not process information using Internet technology; therefore, the outside element is of less concern. However, the company does have a high employee turnover rate and provides remote access via dial-up and remote control software. No policy exists to require unique user IDs, nor are there any procedures to ensure that terminated employees are promptly removed from system access.

The perpetrator (a terminated employee) is angry with BOWS and wants to get back at the company. He would not even need to use the information for his own financial gain. He could simply publish his ability to penetrate BOWS' defenses and create a consumer scare. The direct loss from the incident was $0, but overall damage to business was likely mega-dollars when the consumer community found out about BOWS bad security practices.

LAYING THE ROADBED: POLICY AND STANDARDS

The roadbed consists of policy and standards. Security policy and standards must have muscle. They must include strong yet enforceable statements, clearly written with no room for interpretation, and most importantly must be reasonable and supported by all levels of management. Avoid long or complex policies. As a rule of thumb, no policy should be more than one page in length; a couple of short paragraphs is preferable. Use words in the policy like must, shall, and will. If a policy is something that will not be supported or it is not reasonable to expect someone to follow it to do their job, it should not be published. (See also Exhibit 10-5.) Include somewhere in policy documentation of the disciplinary measures for anyone who does not comply. Procedures and guidelines can provide detail explaining how personnel can comply. To be valid, policy and standards must be consistently enforced. More information on the structure of policy and standards is available later in this article.

Enforcement procedures are the edges of the roadbed. Noncompliance might result in falling off the bridge, which many can relate to being in trouble, especially if one cannot swim. Enforcement provides the boundaries to keep personnel on the proper road. A sample of a simple enforcement procedure for a security violation might be:

1. On the first occurrence, the employee will be informed and given a warning of the importance to comply with policy.
2. On the next occurrence, the employee's supervisor will be contacted. The supervisor will discuss the indiscretion with the employee.
3. Further violations of the same policy will result in disciplinary actions that might consist of suspension or possible termination, depending on the severity of the incident.

In any case, it might be necessary to publish a disclaimer stating that depending on the severity of the incident, disciplinary actions can result in termination. Remember that, to some degree, common sense must come into the decisions regarding how enforcement procedures should be applied, but they should always be consistently enforced. Also, emphasize the fact that it is all management's responsibility to enforce policy, not just the Security Manager's.

Start with the basics, create baselines, and build on them until one has a corporate infrastructure that can stand years and years of traffic. Policy and standards form the benchmarks or reference points for audits. They provide the basis of evidence that management has acted with due diligence, thus reducing their liability.

THE GATE KEEPERS: TECHNOLOGY

Technology is everywhere. In the simplest terms, the security technology consists of specific software that will provide for three basic elements

of protection: authentication, accountability, and audit. Very specific standards provide the baselines for which technology is evaluated, purchased, and implemented. Technology provides the mechanism to enforce policies, standards, and procedures.

Authentication. Authentication is the process by which access is established and the system verifies that the end user requesting access to the information is who they claim to be. The process involves providing one's personal key at the locked gate to open it in order to be able to cross the bridge using the path guarded by that gate.

Accountability. Accountability is the process of assigning appropriate access and identification codes to users in order for them to access the information. Establishing audit trails is what establishes accountability.

An example of accountability in electronic commerce is the assignment of digital certificates that can provide varying levels of guaranteed accountability (trust). At the least trusted levels, the user has a credit card or cash to buy a certificate. At a middle degree of trust, there is more checking done to validate that the user really is the person who they claim to be. At the highest level of trust, an entity is willing to stand behind the accountability of the certificate assignment to make it legally binding. This would mean a signature document was signed in person with the registrant that assigns certificates for establishing the accountability.

Assigning a personal key to an individual who has provided beyond-doubt proof (DNA test) that they are who they say they are and that they have agreed to guard their key with their life and that any access by that key can only be by them.

Audit. This is the process, on which accountability depends that can verify using system events to show beyond a reasonable doubt, that specific activities, authorized or unauthorized, occurred in the system by a specific user identification at a given point in time. The information is available on request and used to report to management, internal and external auditors, and could be used as legal evidence in a criminal prosecution.

Having the necessary proof that the personal (authentication) key assigned (accountable) to Ken M. Shaurette was used to perform an unauthorized activity such as to modify the payroll system, adding bonus bucks to the salaries of all CISSP personnel.

PROVIDING TRANSPORTATION: COMMUNICATION

Communication is the #1 key to the success of any security infrastructure. Not only do policy, standards, procedures, and guidelines need to be communicated, but proper use and availability of the security technologies and processes also need to be communicated. Communications is like the

racecar or the bus that gets the user across the bridge faster from their business need to the information on the other side. Arguably, the most important aspect of security is informing everyone that they have a responsibility for its effectiveness.

CERT estimates that 80 percent of network security intrusions are a result of users selecting and using passwords that are easy to guess and as such are easy to compromise. If users are unaware that bad password selection is a risk, what incentive is there to make better selections? If they knew of guidelines that could help them pick a more difficult password to compromise, would they not be more inclined to do so? If users are unaware that guidelines exist to help them, how can they follow them?

What makes up communications? Communications involves integrating the policy into the organization using a successful security-training program consisting of such things as:

- new employee orientations
- periodic newsletters
- intranet Web site
- electronic announcements (i.e., banners, e-mail)
- CBT course
- technology lunches, dinners
- informal user group forums
- regular company publications
- security awareness days
- ethics and appropriate use agreements signed annually

EXPERT VERSUS FOOL: IMPLEMENTATION RECOMMENDATIONS

Before beginning policy and standard development, understand that in an established organization, policy and standards may exist in different forms. There is probably official, de jure, less official, *de facto* and proprietary, no choice. Official is the law; they are formal and already accepted. Less official consists of things that get followed but are not necessarily published, but maybe should be. Proprietary are the items that are dictated by an operating system; for example, MVS has limitations of eight-character user IDs and eight-character passwords.

Be the Expert: Implementation Recommendations

Form a team or committee that gets the involvement and cooperation of others. If the policies, standards, procedures, and guidelines are to become enterprisewide, supported by every layer of management, and be reasonable and achievable, representation from all areas — both technology and non-technology — will go a long way toward meeting that goal. Only a team

of the most wise and sage experts from all over the organization will know what may already exist and what might still be necessary.

As the security professional, efforts should be concentrated on providing high-level security expertise, coordination, recommendations, communication, and education in order to help the team come to a consensus. Be the engineer, not the builder; get the team to build the bridge.

Layering Security

Layer protection policies and standards. Support them with procedures and guidelines. Review and select security technology that can be standards. Create guidelines and procedures that help users comply with policy. Establishing policy and adequate standards provides the organization with control of its own destiny. Not doing so provides the potential for auditors (internal or external) or legal actions to set policy.

The following walks the reader through the layers outlined in Exhibit 10-4, from the top down.

Corporate Security Policy. This is the top layer of Exhibit 10-4. There should be as few policies as possible used to convey corporate attitude and the attitude from the top down. Policies will have very distinct characteristics. They should be short, enforceable, and seldom change. See Exhibit 10-5 for tips on writing security policy. Policy that gets in the way of business productivity will be ignored or eliminated. Corporate ethics are a form of policy at the top level. Proper use of computing resources or platforms is another example of high-level policy, such as the statement, "for business use only."

> SAMPLE POLICY:
> Information will be protected based on a need-to-know philosophy. Information will be classified and protected in a manner commensurate with its sensitivity, value, and criticality. Protection of information will apply regardless of the media where the information is stored (printed, electronic, etc.), the systems that process it (PC, mainframes, voice mail systems, etc.), or the transport mechanisms by which it is moved (fax, electronic mail, TCP/IP network, voice conversation, etc.).

Functional Standards

Functional standards (the second layer of Exhibit 10-4) are generally associated to a business area. The Loan department in a bank might have standards governing proper handling of certain loan information. For example, a loan department might have a standard with an associated procedure for the special handling of loans applied for by famous people, or executives of the company. Standards might require that information assigned sensitive classification levels is shredded, or an HR department

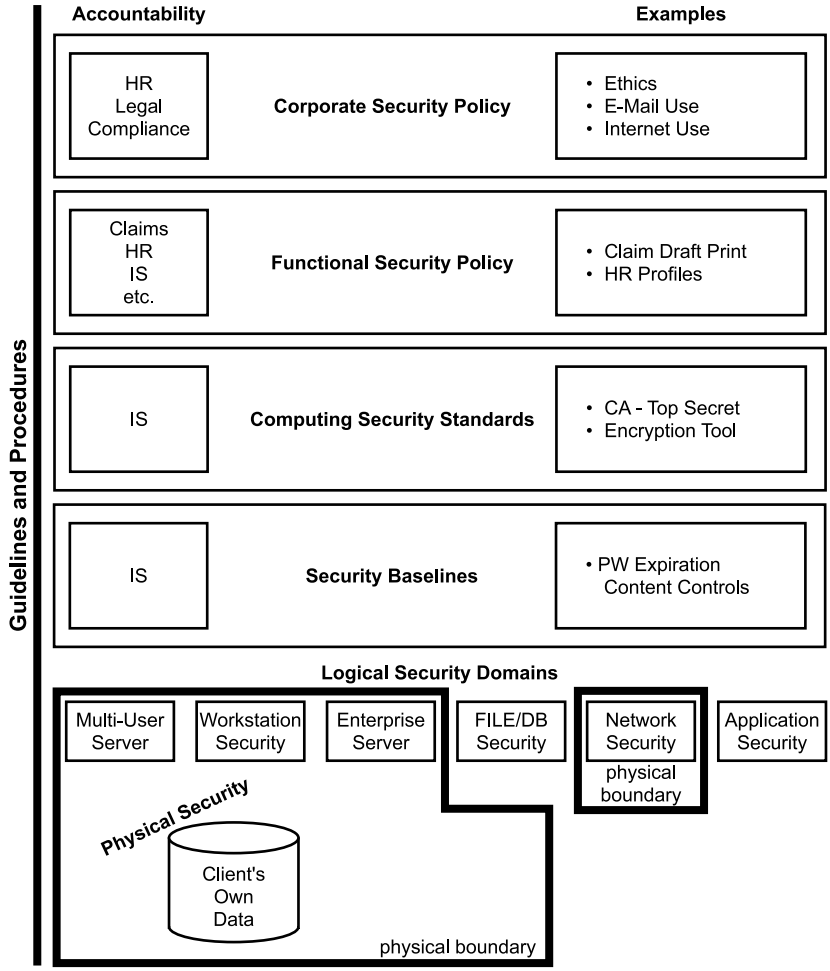


Exhibit 10-4. Layers of security: policies, standards, and procedures.

might require that employee profiles only be printed on secure printers, available and handled only by specific personnel. The Claims department in an insurance company may set standards that require the printing of claim checks on printers local to the office that is handling the claim.

Computing Policy

The computing policies (the third layer in Exhibit 10-4) are tied with technology. These standards establish computing environments such as identifying the standard security software for securing mainframe-computing environments (i.e., CA-Top Secret, RACF, or CA-ACF2), establishing an encryption standard (i.e., PGP, BLOWFISH, DES, 3DES) for every desktop/laptop, or

Exhibit 10-5. Tips on writing security policy.

- Make the policy easy to understand.
- Make it applicable. Does the policy really fit? Does it relate to what actually happens at the company? Does if fit the organizations culture?
- Make it do-able. Can the company still meet business objectives if the policy is implemented?
- Make it enforceable.
- Use a phased-in approach. Allow time for employees to read, digest, and respond to the policy.
- Be pro-active. State what must be done.
- Avoid absolutes; almost never say "never."
- Use wording such as "must," "will," or "shall" — not "would," "should," or "could."
- Meet business objectives. Allow the organization to identify an acceptable level of risk.
- Address all forms of information. (How were the machine names obtained)?
- Obtain appropriate management support.
- Conform. It is important that policy looks like other written company policies.
- Keep it short. Policies are shorter than procedures or practices, usually one or two pages in length maximum.
- What is to be protected?
- When does the policy take effect?
- Where within the organization does the policy reach? Remember the scope.
- To whom does the policy apply? Is there a limitation on the domain?
- Why was the policy developed?
- Who is responsible for enforcement?
- What are the ramifications of noncompliance?
- What, if any, deviations are allowed? If allowed, what are the deviation procedures?
- Are audit trails available and required?
- Who developed, approved, and authorized the policy?
- How will compliance be monitored?
- Are there only penalties for noncompliance, or are rewards available to motivate people toward good practices?
- Who has update and maintenance responsibility for the policies?
- How often will the policy be reviewed and updated if necessary?
- Are there documented approval procedures for new or updated policy?
- Is there an archive of policy, past to present? What was in effect last year at the time of the incident?
- What is the date of the last revision?

transmission of any sensitive information. Information services is most likely establishing the computing security standards that work with information owner requirements, and business needs.

Security Baselines

Security baselines (the fourth layer in Exhibit 10-4) can also be called the minimums. These are tied very closely to the operating environment

and day-to-day functioning of the business. Some baselines might be password expiration intervals, password content controls (six characters must be one numeric or special character), and minimum length of user ID. Another might be requiring that every computing system perform authentication based on a personal identity code that will be assigned to each user and that they use their personal password or alternative authentication (token, biometrics) before access is granted to perform any activities. Audit would also be another baseline requirement.

Technology and Physical Security

Technology and physical security are the components making up the bottom layer of Exhibit 10-4. This is the technology, the security software or hardware, that makes up the various computing platforms that comprise the information processing environment. It is the specific security within an NOS, an application, firewalls for the network, database security, or any other specific technology that provides the actual controls that allow the organization to enforce baselines and standards. An application program may have the security checking that restricts the printing of employee profiles and claim checks or provides alerts and special handling controls for loans by special people.

Procedures and Guidelines

Procedures and guidelines cross all layers of the information security infrastructure, as illustrated in Exhibit 10-4. Guidelines are not required actions, but procedures could fall into either something that must be done or provide help in compliance with security policy, standards, and technology. The best policy and standard can have minimal value if people do not have guidelines to follow. Procedures go that next step in explaining the why and how of policy in the day-to-day business operation to help ensure proper implementation and continued compliance. Policy can only be concise if the guidelines and procedures provide sufficient explanation of how to achieve the business objective. Enforcement is usually spelled out in the form of a procedure; procedures would tell how to and why it is necessary to print to specific printers or handle certain loans in a special way. Guidelines are the hints and tips; for example, sharing one's password does not eliminate one's accountability; choose passwords that are not easily guessed and give sample techniques for password selection. Help personnel find the right path and they will follow it; reminders of the consequences are good incentives.

THE POLICE ARE COMING!

In conclusion, what are the measures that can be taken to protect the company or management from litigation? Security cannot provide 100 percent

protection. There will be a need to accept some risk. Recognize due care methods to reduce and limit liability by minimizing how much risk must be accepted. Computer security is often an intangible process. In many instances, the level of security is not evident until a catastrophe happens, at which time the lack of security is all too painfully evident. Make the protection of corporate information assets "the law." Make adherence to policy and standards a condition of employment. Policy, standards, and procedures must become part of the corporation's living structure, not just a policy development effort. Information security's objective is to enhance the productivity of the business by reducing probability of loss through the design and implementation of policies, standards, procedures, and guidelines that enhance the protection of business assets.

- Information security is not just about technological controls such as software or hardware. Establishing policy and adequate standards provide an organization with control over its own destiny.
- Information security should be viewed in terms of the processes and goals of the business. Business risk is different than security risk, but poor security can put the business at risk; or make it risky doing business.
- Security must become a function of the corporation, and not viewed as an obstacle to business. Policies support the business; put them in business terminology.
- Form a team. Only a team of the most wise and sage experts from all over the organization will know what policy may already exist and what might still be necessary.
- There should be as few policies as possible used to convey corporate attitude and the attitude from the top down. Policies will have very distinct characteristics. They should be short, enforceable, and seldom altered. They must include strong yet enforceable statements, be clearly written with no room for interpretation, and most importantly, must be reasonable and supported by all levels of management. Use words in the policy like must, shall, and will.
- Policy can only be concise if the guidelines and procedures provide sufficient explanation of how to achieve the business objective.
- Test policy and standards; it is easy to know what is published, but is that what is really in operation?
- To be valid, policy and standards must be consistently enforced.
- Carefully define the Security Manager's domain, responsibility, and accountabilities. Clearly identify the scope of their job.
- Communication is the #1 key to the success of any security infrastructure.

> To defeat a strong enemy: Deploy forces to defend the strategic points; exercise vigilance in preparation, do not be indolent. Deeply investigate the true situation, secretly await their laxity. Wait until they leave their strongholds, then seize what they love.
>
> — Sun Tzu

Information security is a team effort; all members in an organization must support the business objectives; and information security is an important part of that objective.

55

The Human Side of Information Security

Kevin Henry, CISA, CISSP

We often hear that people are the weakest link in any security model. That statement brings to mind the old adage that a chain is only as strong as its weakest link. Both of these statements may very well be true; however, they can also be false and misleading.

Throughout this chapter we are going to define the roles and responsibilities of people, especially in relation to information security. We are going to explore how people can become our strongest asset and even act as a compensating strength for areas where mechanical controls are ineffective. We will look briefly at the training and awareness programs that can give people the tools and knowledge to increase security effectiveness rather than be regarded as a liability and a necessary evil.

The Role of People in Information Security

First, we must always remember that systems, applications, products, etc. were created for people — not the other way around. As marketing personnel know, the end of any marketing plan is when a product or service is purchased for, and by, a person. All of the intermediate steps are only support and development for the ultimate goal of providing a service that a person is willing, or needs, to purchase. Even though many systems in development are designed to reduce labor costs, streamline operations, automate repetitive processes, or monitor behavior, the system itself will still rely on effective management, maintenance upgrades, and proper use by individuals. Therefore, one of the most critical and useful shifts in perspective is to understand how to get people committed to and knowledgeable about their roles and responsibilities as well as the importance of creating, enforcing, and committing to a sound security program.

Properly trained and diligent people can become the strongest link in an organization's security infrastructure. Machines and policy tend to be static and limited by historical perspectives. People can respond quickly, absorb new data and conditions, and react in innovative and emotional ways to new situations. However, while a machine will enforce a rule it does not understand, people will not support a rule they do not believe in. The key to strengthening the effectiveness of security programs lies in education, flexibility, fairness, and monitoring.

The Organization Chart

A good security program starts with a review of the organization chart. From this administrative tool, we learn hints about the structure, reporting relationships, segregation of duties, and politics of an organization. When we map out a network, it is relatively easy to slot each piece of equipment into its proper place, show how data flows from one place to another, show linkages, and expose vulnerabilities. It is the same with an organization chart. Here we can see the structure of an organization, who reports to whom, whether authority is distributed or centralized, and who has the ability or placement to make decisions — both locally and throughout the enterprise.

Why is all of this important? In some cases, it is not. In rare cases, an ideal person in the right position is able to overcome some of the weaknesses of a poor structure through strength or personality. However, in nearly all cases, people fit into their relative places in the organizational structure and are constrained by the limitations and boundaries placed around them. For example, a security department or an emergency planning group may be buried deep within one *silo* or branch of an organization. Unable to speak directly with decision makers, financial approval teams, or to have influence over other branches, their efforts become more or less philosophical and ineffective. In such an environment the true experts often leave in frustration and are replaced by individuals who thrive on meetings and may have limited vision or goals.

Do We Need More Policy?

Many recent discussions have centered on whether the information security community needs more policy or to simply get down to work. Is all of this talk about risk assessment, policy, roles and responsibilities, disaster recovery planning, and all of the other *soft* issues that are a part of an information security program only expending time and effort with few results? In most cases, this is probably true. Information security must be a cohesive, coordinated action, much like planning any other large project. A house can be built without a blueprint, but endless copies of blueprints and modifications will not build a house. However, proper planning and methodologies will usually result in a project that is on time, meets customer needs, has a clearly defined budget, stays within its budget, and is almost always run at a lower stress level. As when a home is built, the blueprints almost always change, modifications are done, and, together with the physical work, the administrative effort keeps the project on track and schedules the various events and subcontractors properly.

Many firms have information security programs that are floundering for lack of vision, presentation, and coordination. For most senior managers, information security is a gaping dark hole into which vast amounts of cash are poured with few outcomes except further threats, fear-mongering, and unseen results.

To build an effective program requires vision, delegation, training, technical skills, presentation skills, knowledge, and often a thick skin — not necessarily in that order.

The program starts with a vision. What do we want to accomplish? Where would we like to be? Who can lead and manage the program? How can we stay up-to-date, and how can we do it with limited resources and skills?

A vision is the perception we have of the goal we want to reach. A vision is not a fairy tale but a realistic and attainable objective with clearly defined parameters. A vision is not necessarily a roadmap or a listing of each component and tool we want to use; rather, it is a strategy and picture of the functional benefits and results that would be provided by an effective implementation of the strategic vision.

How do we define our vision? This is a part of policy development, adherence to regulations, and risk assessment. Once we understand our security risks, objectives, and regulations, we can begin to define a practical approach to addressing these concerns.

A recent seminar was held with security managers and administrators from numerous agencies and organizations. The facilitator asked the group to define four major technical changes that were on the horizon that would affect their agencies. Even among this knowledgeable group, the response indicated that most were unaware of the emerging technologies. They were knowledgeable about current developments and new products but were unaware of dramatic changes to existing technologies that would certainly have a major impact on their operations and technical infrastructures within the next 18 months. This is a weakness among many organizations. Strategic planning has been totally overwhelmed by the need to do operational and tactical planning.

Operational or day-to-day planning is primarily a response mechanism — how to react to today's issues. This is kindly referred to as crisis management; however, in many cases the debate is whether the managers are managing the crisis or the crisis is managing the managers.

Tactical planning is short- to medium-term planning. Sometimes, tactical planning is referred to in a period of up to six months. Tactical planning is forecasting developments to existing strategies, upgrades, and operational process changes. Tactical planning involves understanding the growth, use, and risks of the environment. Good tactical plans prevent performance impacts from over-utilization of hardware resources, loss of key personnel, and market changes. Once tactical planning begins to falter, the impact is felt on operational activity and planning within a short timeframe.

Strategic planning was once called long-term planning, but that is relative to the pace of change and volatility of the environment. Strategic planning is preparing for totally new approaches and technologies. New projects,

marketing strategies, new risks, and economic conditions are all a part of a good strategic plan. Strategic planning is looking ahead to entirely new solutions for current and future challenges — seeing the future and how the company or organization can poise itself to be ready to adopt new technologies. A failure to have a strategic plan results in investment in technologies that are outdated, have a short life span, are ineffective, do not meet the expectations of the users, and often result in a lack of confidence by senior management (especially from the user groups) in the information technology or security department.

An information security program is not only a fire-fighting exercise; yet for many companies, that is exactly what they are busy with. Many system administrators are averaging more than five patch releases a week for the systems for which they are responsible. How can they possibly keep up and test each new patch to ensure that it does not introduce other problems? Numerous patches have been found to contain errors or weaknesses that affect other applications or systems. In October 2001, anti-virus companies were still reporting that the LoveLetter virus was accounting for 2.5 percent of all help desk calls — more than a year after patches were available to prevent infection.[1]

What has gone wrong? How did we end up in the position we are in today? The problem is that not any one person can keep up with this rapidly growing and developing field. Here, therefore, is one of the most critical reasons for delegation: the establishment of the principles of responsibility and accountability in the correct departments and with the proper individuals.

Leadership and placement of the security function is an ongoing and never-to-be-resolved debate. There is not a one-size-fits-all answer; however, the core concern is whether the security function has the influence and authority it needs to fulfill its role in the organization.

The role of security is to inform, monitor, lead, and enforce best practice. As we look further at each individual role and responsibility in this chapter, we will define some methods of passing on information or awareness training.

Security Placement

The great debate is where the security department should reside within an organization. There are several historical factors that apply to this question. Until recently, physical security was often either outsourced or considered a less-skilled department. That was suitable when security consisted primarily of locking doors and patrolling hallways. Should this older physical security function be merged into the technical and cyber-security group?

To use our earlier analogy of security being a chain, and the risk that one weak link may have a serious impact on the entire chain, it is probable that combining the functions of physical and technical security is appropriate. Physical access to equipment presents a greater risk than almost any other vulnerability. The trend to incorporate security, risk management, business continuity, and sometimes even audit under one group led by a chief risk officer is recognition both of the importance of these various functions and the need for these groups to work collaboratively to be effective.

The position of chief risk officer (CRO) is usually as a member of the senior management team. From this position, the CRO can ensure that all areas of the organization are included in risk management and disaster recovery planning. This is an extremely accountable position. The CRO must have a team of diligent and knowledgeable leaders who can identify, assess, analyze, and classify risks, data, legislation, and regulation. They must be able to convince, facilitate, coordinate, and plan so that results are obtained; workable strategies become tactical plans; and all areas and personnel are aware, informed, and motivated to adhere to ethics, best practices, policy, and emergency response.

As with so many positions of authority, and especially in an area where most of the work is administrative such as audit, business continuity planning, and risk management, the risk of gathering a team of paper pushers and "yes men" is significant. The CRO must resist this risk by encouraging the leaders of the various departments to keep each other sharp, continue raising the bar, and striving for greater value and benefits.

The Security Director

The security director should be able to coordinate the two areas of physical and technical security. This person has traditionally had a law enforcement background, but these days it is important that this person have a good understanding of information systems security. This person ideally should have certification such as the

CISSP (Certified Information Systems Security Professional administered by ISC² [www.isc2.org]) and experience in investigation and interviewing techniques. Courses provided by companies like John E. Reid and Associates can be an asset for this position.

Roles and Responsibilities

The security department must have a clearly defined mandate and reporting structure. All of its work should be coordinated with the legal and human resources departments. In extreme circumstances it should have access directly to the board of directors or another responsible position so that it can operate confidentially anywhere within the organization, including the executive management team. All work performed by security should be kept confidential in order to protect information about ongoing investigations or erroneously damage the reputation of an individual or a department.

Security should also be a focus point to which all employees, customers, vendors, and the public can refer questions or threats. When an employee receives an e-mail that he suspects may contain a virus or that alleges a virus is on the loose, he should know to contact security for investigation — and not to send the e-mail to everyone he knows to warn them of the perceived threat.

The security department enforces organizational policy and is often involved in the crafting and implementation of policy. As such, this department needs to ensure that policy is enforceable, understandable, comprehensive, up-to-date, and approved by senior management.

Training and Awareness

The security director has the responsibility of promoting education and awareness as well as staying abreast of new developments, threats, and countermeasures. Association with organizations such as SANS (www.sans.org), ISSA (www.issa.org), and CSI (www.gocsi.org) can be beneficial. There are many other groups and forums out there; and the director must ensure that the most valued resources are used to provide alerts, trends, and product evaluation.

The security department must work together with the education and training departments of the organization to be able to target training programs in the most effective possible manner. Training needs to be relevant to the job functions and risks of the attendees. If the training can be imparted in such a way that the attendees are learning the concepts and principles without even realizing how much they have learned, then it is probably ideal. Training is not a "do not do this" activity — ideally, training does not need to only define rules and regulations; rather, training is an activity designed to instill a concept of best practice and understanding to others. Once people realize the reasons behind a guideline or policy, they will be more inclined to better standards of behavior than they would if only pressured into a firm set of rules.

Training should be creative, varied, related to real life, and frequent. Incorporating security training into a ten-minute segment of existing management and staff meetings, and including it as a portion of the new employee orientation process, is often more effective than a day-long seminar once a year. Using examples can be especially effective. The effectiveness of the training is increased when an actual incident known to the staff can be used as an example of the risks, actions, retribution, and reasoning associated with an action undertaken by the security department. This is often called *dragging the wolf into the room*. When a wolf has been taking advantage of the farmer, bringing the carcass of the wolf into the open can be a vivid demonstration of the effectiveness of the security program. When there has been an incident of employee misuse, bringing this into the open (in a tactful manner) can be a way to prevent others from making the same mistakes. Training is not fear mongering. The attitude of the trainers should be to raise the awareness and behavior of the attendees to a higher level, not to explain the rules as if to criminals that they had "better behave or else."

This is perhaps the greatest strength of the human side of information security. Machines can be programmed with a set of rules. The machine then enforces these rules mechanically. If prople are able to slightly modify their activity or use a totally new attack strategy, they may be able to circumvent the rules and attack the machine or network. Also — because machines are controlled by people — when employees feel unnecessarily constrained by a rule, they may well disable or find a way to bypass the constraint and leave a large hole in the rule base. Conversely, a security-conscious person may be able to detect an aberration in behavior or even attitude that could be a precursor to an attack that is well below the detection level of a machine.

Reacting to Incidents

Despite our best precautions and controls, incidents will arise that test the strength of our security programs. Many incidents may be false alarms that can be resolved quickly; however, one of the greatest fears with false alarms is the tendency to become immune to the alarms and turn off the alarm trigger. All alarms should be logged and resolved. This may be done electronically, but it should not be overlooked. Alarm rates can be critical indicators of trends or other types of attacks that may be emerging; they can also be indicators of additional training requirements or employees attempting to circumvent security controls.

One of the tools used by security departments to reduce nuisance or false alarms is the establishment of clipping levels or thresholds for alarm activation. The clipping level is the acceptable level of error before triggering the alarm. These are often used for password lockout thresholds and other low-level activity. The establishment of the correct clipping level depends on historical events, the sensitivity of the system, and the granularity of the system security components. Care must be exercised to ensure that clipping levels are not set too high such that a low-level attack can be performed without bringing in an alarm condition.

Many corporations use a tiered approach to incident response. The initial incident or alarm is recognized by a help-desk or low-level technical person. This person logs the alarm and attempts to resolve the alarm condition. If the incident is too complex or risky to be resolved at this level, the technician refers the alarm to a higher-level technical expert or to management. It is important for the experts to routinely review the logs of the alarms captured at the initial point of contact so that they can be assured that the alarms are being handled correctly and to detect relationships between alarms that may be an indication of further problems.

Part of good incident response is communication. To ensure that the incident is handled properly and risk to the corporation is minimized, a manner of distributing the information about the incident needs to be established. Pagers, cell phones, and e-mail can all be effective tools for alerting key personnel. Some of the personnel that need to be informed of an incident include senior management, public relations, legal, human resources, and security.

Incident handling is the expertise of a good security team. Proper response will contain the damage; assure customers, employees, and shareholders of adequate preparation and response skills; and provide feedback to prevent future incidents.

When investigating an incident, proper care must be taken to preserve the information and evidence collected. The victims or reporting persons should be advised that their report is under investigation.

The security team is also responsible for reviewing past incidents and making recommendations for improvements or better controls to prevent future damage. Whenever a business process is affected, and the business continuity plan is enacted, security should ensure that all assets are protected and controls are in place to prevent disruption of recovery efforts.

Many corporations today are using managed security service providers (MSSPs) to monitor their systems. The MSSP accumulates the alarms and notifies the corporation when an alarm or event of significant seriousness occurs. When using an MSSP, the corporation should still have contracted measurement tools to evaluate the appropriateness and effectiveness of the MSSP's response mechanisms. A competent internal resource must be designated as the contact for the MSSP.

If an incident occurs that requires external agencies or other companies to become involved, a procedure for contacting external parties should be followed. An individual should not contact outside groups without the approval and notification of senior management. Policy must be developed and monitored regarding recent laws requiring an employee to alert police forces of certain types of crimes.

The IT Director — The Chief Information Officer (CIO)

The IT director is responsible for the strategic planning and structure of the IT department. Plans for future systems development, equipment purchase, technological direction, and budgets all start in the office of the IT director. In most cases, the help desk, system administrators, development departments, production support, operations, and sometimes even telecommunications departments are included in his jurisdiction.

The security department should not report to the IT director because this can create a conflict between the need for secure processes and the push to develop new systems. Security can often be perceived as a roadblock for operations and development staff, and having both groups report to the same manager can cause conflict and jeopardize security provisioning.

The IT director usually requires a degree in electrical engineering or computer programming and extensive experience in project planning and implementation. This is important for an understanding of the complexities and challenges of new technologies, project management, and staffing concerns. The IT director or CIO should sit on the senior management team and be a part of the strategic planning process for the organization. Facilitating business operations and requirements and understanding the direction and technology needs of the corporation are critical to ensuring that a gulf does not develop between IT and the sales, marketing, or production shops. In many cases, corporations have been limited in their flexibility due to the cumbersome nature of legacy systems or poor communications between IT development and other corporate areas.

The IT Steering Committee

Many corporations, agencies, and organizations spend millions of dollars per year on IT projects, tools, staff, and programs and yet do not realize adequate benefits or return on investment (ROI) for the amounts of money spent. In many cases this is related to poor project planning, lack of a structured development methodology, poor requirements definition, lack of foresight for future business needs, or lack of close interaction between the IT area and the business units. The IT steering committee is comprised of leaders from the various business units of the organization and the director of IT. The committee has the final approval for any IT expenditures and project prioritization. All proposed IT projects should be presented to the committee along with a thorough business case and forecast expenditure requirements. The committee then determines which projects are most critical to the organization according to risk, opportunities, staffing availability, costs, and alignment with business requirements. Approval for the projects is then granted.

One of the challenges for many organizations is that the IT steering committee does not follow up on ongoing projects to ensure that they meet their initial requirements, budget, timeframes, and performance. IT steering committee members need to be aware of business strategies, technical issues, legal and administrative requirements, and economic conditions. They need the ability to overrule the IT director and cancel or suspend any project that may not provide the functionality required by the users, adequate security, or is seriously over budget. In such cases the IT steering committee may require a detailed review of the status of the project and reevaluate whether the project is still feasible.

Especially in times of weakening IT budgets, all projects should undergo periodic review and rejustification. Projects that may have been started due to hype or the proverbial bandwagon — "everyone must be E-business or they are out of business" — and do not show a realistic return on investment should be cancelled. Projects that can save money must be accelerated — including in many cases a piecemeal approach to getting the most beneficial portions implemented rapidly. Projects that will result in future savings, better technology, and more market flexibility need to be continued, including projects to simplify and streamline IT infrastructure.

Change Management — Certification and Accreditation

Change management is one of the greatest concerns for many organizations today. In our fast-paced world of rapid development, short time to market, and technological change, change management is the key to ensuring that a "sober second thought" is taken before a change to a system goes into production. Many times, the pressure to make a change rapidly and without a formal review process has resulted in a critical system failure due to inadequate testing or unanticipated or unforeseen technical problems.

There are two sides to change management. The most common definition is that change management is concerned with the certification and accreditation process. This is a control set in place to ensure that all changes that are proposed to an existing system are properly tested, approved, and structured (logically and systematically planned and implemented).

The other aspect of change management comes from the project management and systems development world. When an organization is preparing to purchase or deploy a new system, or modify an existing system, the organization will usually follow a project management framework to control the budget, training, timing, and staffing requirements of the project. It is common (and often expected, depending on the type of development life cycle employed) that such projects will undergo significant changes or decision points throughout the project lifetime. The decision points are times when evaluations of the project are made and a choice to either continue or halt the project may be required. Other changes may be made to a project due to external factors — economic climate, marketing forces, and availability of skilled personnel — or to internal factors

such as identification of new user requirements. These changes will often affect the scope of the project (the amount of work required and the deliverables) or timing and budgeting. Changes made to a project in midstream may cause the project to become unwieldy, subject to large financial penalties — especially when dealing with an outsourced development company — or delayed to the point of impacting business operations. In this instance, change management is the team of personnel that will review proposed changes to a project and determine the cutoff for modifications to the project plan. Almost everything we do can be improved and as the project develops, more ideas and opportunities arise. If uncontrolled, the organization may well be developing a perfect system that never gets implemented. The change control committee must ensure that a time comes when the project timeline and budget are set and followed, and refuse to allow further modifications to the project plan — often saving these ideas for a subsequent version or release.

Change management requires that all changes to hardware, software, documentation, and procedures are reviewed by a knowledgeable third party prior to implementation. Even the smallest change to a configuration table or attaching a new piece of equipment can cause catastrophic failures to a system. In some cases a change may open a security hole that goes unnoticed for an extended period of time. Changes to documentation should also be subject to change management so that all documents in use are the same version, the documentation is readable and complete, and all programs and systems have adequate documentation. Furthermore, copies of critical documentation need to be kept off-site in order to be available in the event of a major disaster or loss of access to the primary location.

Certification

Certification is the review of the system from a user perspective. The users review the changes and ensure that the changes will meet the original business requirements outlined at the start of the project or that they will be compatible with existing policy, procedures, or business objectives. The other user group involved is the security department. This group needs to review the system to ensure that it is adequately secured from threats or risks. In this they will need to consider the sensitivity of the data within the system or that the system protects, the reliance of the business process on the system (availability), regulatory requirements such as data protection or storage (archival) time, and documentation and user training.

Accreditation

Once a system has been certified by the users, it must undergo accreditation. This is the final approval by management to permit the system, or the changes to a component, to move into production. Management must review the changes to the system in the context of its operational setting. They must evaluate the certification reports and recommendations from security regarding whether the system is adequately secured and meets user requirements and the proposed implementation timetable. This may include accepting the residual risks that could not be addressed in a cost-effective manner.

Change management is often handled by a committee of business analysts, business unit directors, and security and technical personnel. They meet regularly to approve implementation plans and schedules. Ideally, no change will go into production unless it has been thoroughly inspected and approved by this committee. The main exceptions to this, of course, are changes required to correct system failures. To repair a major failure, a process of emergency change management must be established. The greatest concern with emergency changes is ensuring that the correct follow-up is done to ensure that the changes are complete, documented, and working correctly.

In the case of volatile information such as marketing programs, inventory, or newsflashes, the best approach is to keep the information stored in tables or other logically separated areas so that these changes (which may not be subject to change management procedures) do not affect the core system or critical functionality.

Technical Standards Committee

Total cost of ownership (TCO) and keeping up with new or emerging tools and technologies are areas of major expenditure for most organizations today. New hardware and software are continuously marketed. In many cases a new operating system may be introduced before the organization has completed the rollout of the previous version. This often means supporting three versions of software simultaneously. Often this has resulted

in the inability of personnel still using the older version of the software to read internal documents generated under the newer version. Configurations of desktops or other hardware can be different, making support and maintenance complex. Decisions have to be made about which new products to purchase — laptops instead of desktops, the minimum standards for a new machine, or type of router or network component. All of these decisions are expensive and require a long-term view of what is coming onto the horizon.

The technical standards committee is an advisory committee and should provide recommendations (usually to the IT steering committee or another executive-level committee) for the purchase, strategy, and deployment of new equipment, software, and training. The members of the technical standards committee must be aware of the products currently available as well as the emerging technologies that may affect the viability of current products or purchases. No organization wants to make a major purchase of a software or hardware product that will be incompatible with other products the organization already has or will require within the next few months or years. The members of the technical standards committee should consist of a combination of visionaries, technical experts, and strategic business planners. Care should be taken to ensure that the members of this committee do not become unreasonably influenced by or restricted to one particular vendor or supplier.

Central procurement is a good principle of security management. Often when an organization is spread out geographically, there is a tendency for each department to purchase equipment independently. Organizations lose control over standards and may end up with incompatible VPNs, difficult maintenance and support, loss of savings that may have been available through bulk purchases, cumbersome disaster recovery planning through the need to communicate with many vendors, and loss of inventory control. Printers and other equipment become untraceable and may be subject to theft or misuse by employees. One organization recently found that tens of thousands of dollars' worth of equipment had been stolen by an employee that the organization never realized was missing. Unfortunately for the employee, a relationship breakdown caused an angry partner to report the employee to corporate security.

The Systems Analyst

There are several definitions for a systems analyst. Some organizations may use the term *senior analyst* when the person works in the IT development area; other organizations use the term to describe the person responsible for systems architecture or configuration.

In the IT development shop, the systems analyst plays a critical role in the development and leadership of IT projects and the maintenance of IT systems. The systems analyst may be responsible for chairing or sitting on project development teams, working with business analysts to determine the functional requirements for a system, writing high-level project requirements for use by programmers to write code, enforcing coding standards, coordinating the work of a team of programmers and reviewing their work, overseeing production support efforts, and working on incident handling teams.

The systems analyst is usually trained in computer programming and project management skills. The systems analyst must have the ability to review a system and determine its capabilities, weaknesses, and workflow processes.

Systems analysts should not have access to change production data or programs. This is important to ensure that they cannot inadvertently or maliciously change a program or organizational data. Without such controls, the analyst may be able to introduce a Trojan horse, circumvent change control procedures, and jeopardize data integrity.

Systems analysts in a network or overall systems environment are responsible for ensuring that secure and reliable networks or systems are developed and maintained. They are responsible for ensuring that the networks or systems are constructed with no unknown gaps or backdoors, that there are few single points of failure, that configurations and access control procedures are set up, and that audit trails and alarms are monitored for violations or attacks.

The systems analyst usually requires a technical college diploma and extensive in-depth training. Knowledge of system components, such as the firewalls in use by the organization, tools, and incident handling techniques, is required.

Most often, the systems analyst in this environment will have the ability to set up user profiles, change permissions, change configurations, and perform high-level utilities such as backups or database reorganizations. This creates a control weakness that is difficult to overcome. In many cases the only option an organization has is to trust the person in this position. Periodic reviews of their work and proper management controls are

some of the only compensating controls available. The critical problem for many organizations is ensuring that this position is properly backed up with trained personnel and thorough documentation, and that this person does not become technically stagnant or begin to become sloppy about security issues.

The Business Analyst

The business analyst is one of the most critical roles in the information management environment. A good business analyst has an excellent understanding of the business operating environment, including new trends, marketing opportunities, technological tools, current process strengths, needs, and weaknesses, and is a good team member. The business analyst is responsible for representing the needs of the users to the IT development team. The business analyst must clearly articulate the functional requirements of a project early on in the project life cycle in order to ensure that information technology resources, money, personnel, and time are expended wisely and that the final result of an IT project meets user needs, provides adequate security and functionality, and embraces controls and separation of duties. Once outlined, the business analyst must ensure that these requirements are addressed and documented in the project plan. The business analyst is then responsible for setting up test scenarios to validate the performance of the system and verify that the system meets the original requirements definitions.

When testing, the business analyst should ensure that test scenarios and test cases have been developed to address all recognized risks and test scenarios. Test data should be sanitized to prevent disclosure of private or sensitive information, and test runs of programs should be carefully monitored to prevent test data and reports from introduction into the real-world production environment. Tests should include out-of-range tests, where numbers larger or smaller than the data fields are attempted and invalid data formats are tried. The purpose of the tests is to try to see if it is possible to make the system fail. Proper test data is designed to stress the limitations of the system, the edit checks, and the error handling routines so that the organization can be confident that the system will not fail or handle data incorrectly once in production. The business analyst is often responsible for providing training and documentation to the user groups. In this regard, all methods of access, use, and functionality of the system from a user perspective should be addressed. One area that has often been overlooked has been assignment of error handling and security functionality. The business analyst must ensure that these functions are also assigned to reliable and knowledgeable personnel once the system has gone into production.

The business analyst is responsible for reviewing system tests and approving the change as the certification portion of the change management process. If a change needs to be made to production data, the business analyst will usually be responsible for preparing or reviewing the change and approving the timing and acceptability of the change prior to its implementation. This is a proper segregation of duties, whereby the person actually making the change in production — whether it is the operator, programmer, or other user — is not the same person who reviews and approves the change. This may prevent either human error or malicious changes.

Once in production, business analysts are often the second tier of support for the user community. Here they are responsible to check on inconsistencies, errors, or unreliable processing by the system. They will often have a method of creating trouble tickets or system failure notices for the development and production support groups to investigate or take action.

Business analysts are commonly chosen from the user groups. They must be knowledgeable in the business operations and should have good communication and teamwork skills. Several colleges offer courses in business analysis, and education in project management can also be beneficial.

Because business analysts are involved in defining the original project functional requirements, they should also be trained in security awareness and requirements. Through a partnership with security, business analysts can play a key role in ensuring that adequate security controls are included in the system requirements.

The Programmer

This chapter is not intended to outline all of the responsibilities of a programmer. Instead, it focuses on the security components and risks associated with this job function. The programmer, whether in a mainframe, client/server, or Web development area, is responsible for preparing the code that will fulfill the requirements of the

users. In this regard, the programmer needs to adhere to principles that will provide reliable, secure, and maintainable programs without compromising the integrity, confidentiality, or availability of the data. Poorly written code is the source of almost all buffer overflow attacks. Because of inadequate bounds, parameter checking, or error handling, a program can accept data that exceeds its acceptable range or size, thereby creating a memory or privilege overflow condition. This is a potential hole either for an attacker to exploit or to cause system problems due to simple human error during a data input function.

Programs need to be properly documented so that they are maintainable, and the users (usually business analysts) reviewing the output can have confidence that the program handles the input data in a consistent and reliable manner.

Programmers should never have access to production data or libraries. Several firms have experienced problems due to disgruntled programmers introducing logic bombs into programs or manipulating production data for their own benefit. Any changes to a program should be reviewed and approved by a business analyst and moved into production by another group or department (such as operators), and not by the programmer directly. This practice was established during the mainframe era but has been slow to be enforced on newer Web-based development projects. This has meant that several businesses have learned the hard way about proper segregation of duties and the protection it provides a firm. Often when a program requires frequent updating, such as a Web site, the placement of the changeable data into tables that can be updated by the business analysts or user groups is desirable.

One of the greatest challenges for a programmer is to include security requirements in the programs. A program is primarily written to address functional requirements from a user perspective, and security can often be perceived as a hindrance or obstacle to the fast execution and accessibility of the program. The programmer needs to consider the sensitivity of the data collected or generated by the program and provide secure program access, storage, and audit trails. Access controls are usually set up at the initiation of the program; and user IDs, passwords, and privilege levels are checked when the user first logs on to the system or program. Most programs these days have multiple access paths to information — text commands, GUI icons, and drop-down menus are some of the common access methods. A programmer must ensure that all access methods are protected and that the user is unable to circumvent security by accessing the data through another channel or method.

The programmer needs training in security and risk analysis. The work of a programmer should also be subject to peer review by other systems analysts or programmers to ensure that quality and standard programming practices have been followed.

The Librarian

The librarian was a job function established in a mainframe environment. In many cases the duties of the librarian have now been incorporated into the job functions of other personnel such as system administrators or operators. However, it is important to describe the functions performed by a librarian and ensure that these tasks are still performed and included in the performance criteria and job descriptions of other individuals.

The librarian is responsible for the handling of removable media — tapes, disks, and microfiche; the control of backup tapes and movement to off-site or near-line storage; the movement of programs into production; and source code control. In some instances the librarian is also responsible for system documentation and report distribution.

The librarian duties need to be described, assigned, and followed. Movement of tapes to off-site storage should be done systematically with proper handling procedures, secure transport methods, and proper labeling. When reports are generated, especially those containing sensitive data, the librarian must ensure that the reports are distributed to the correct individuals and no pages are attached in error to other print jobs. For this reason, it is a good practice to restrict the access of other personnel from the main printers.

The librarian accepts the certified and accredited program changes and moves them into production. These changes should always include a back-out plan in case of program or system problems. The librarian should take a backup copy of all programs or tables subject to change prior to moving the new code into production. A librarian should always ensure that all changes are properly approved prior to making a change.

Librarians should not be permitted to make changes to programs or tables; they should only enact the changes prepared and approved by other personnel. Librarians also need to be inoculated against social engineering or pressure from personnel attempting to make changes without going through the proper approval process.

The Operator

The operator plays a key role in information systems security. No one has greater access or privileges than the operator. The operator can be a key contributor to system security or a gaping hole in a security program. The operator is responsible for the day-to-day operations, job flow, and often the scheduling of the system maintenance and backup routines. As such, an operator is in a position that may have serious impact on system performance or integrity in the event of human error, job-sequencing mistakes, processing delays, backup execution, and timing. The operator also plays a key role in incident handling and error recovery. The operator should log all incidents, abnormal conditions, and job completions so that they can be tracked and acted upon, and provide input for corrective action. Proper tracking of job performance, storage requirements, file size, and database activity provides valuable input to forecasting requirements for new equipment or identification of system performance issues and job inefficiencies before they become serious processing impairments.

The operator should never make changes to production programs or tables except where the changes have been properly approved and tested by other personnel. In the event of a system failure, the operator should have a response plan in place to notify key personnel.

The System Owner and the Data Owner

History has taught us that information systems are not owned by the information technology department, but rather by the user group that depends on the system. The system owner therefore is usually the senior manager in the user department. For a financial system this may be the vice president of finance; for a customer support system, the vice president of sales. The IT department then plays the role of supporting the user group and responding to the needs of the user. Proper ownership and control of systems may prevent the development of systems that are technically sound but of little use to the users. Recent studies have shown that the gap between user requirements and system functionality was a serious detriment to business operations. In fact, several government departments have had to discard costly systems that required years of development because they were found to be inadequate to meet business needs.[2]

The roles of system owner and data owner may be separate or combined, depending on the size and complexity of the system. The system owner is responsible for all changes and improvements to a system, including decisions regarding the overall replacement of a system. The system owner sits on the IT steering committee, usually as chair, and provides input, prioritization, budgeting, and high-level resource allocation for system maintenance and development. This should not conflict with the role of the IT director and project leaders who are responsible for the day-to-day operations of production support activity, development projects, and technical resource hiring and allocation. The system owner also oversees the accreditation process that determines when a system change is ready for implementation. This means the system owner must be knowledgeable about new technologies, risks, threats, regulations, and market trends that may impact the security and integrity of a system.

The responsibility of the data owner is to monitor the sensitivity of the data stored or processed by a system. This includes determining the appropriate levels of information classification, access restrictions, and user privileges. The data owner should establish or approve the process for granting access to new users, increasing access levels for existing users, and removing access in a timely manner for users who no longer require access as a part of their job duties. The data owner should require an annual report of all system users and determine whether the level of access each user has is appropriate. This should include a review of special access methods such as remote access, wireless access, reports received, and ad hoc requests for information.

Because these duties are incidental to the main functions of the persons acting as data or system owners, it is incumbent upon these individuals to closely monitor these responsibilities while delegating certain functions to other persons. The ultimate responsibility for accepting the risks associated with a system rests with the system and data owners.

The User

All of the systems development, the changes, modifications, and daily operations are to be completed with the objective of addressing user requirements. The user is the person who must interact daily with the system and

relies on the system to continue business operations. A system that is not designed correctly may lead to a high incidence of user errors, high training costs or extended learning curves, poor performance and frustration, and overly restrictive controls or security measures. Once users notice these types of problems, they will often either attempt to circumvent security controls or other functionality that they find unnecessarily restrictive or abandon the use of the system altogether.

Training for a user must include the proper use of the system and the reasons for the various controls and security parameters built into the system. Without divulging the details of the controls, explaining the reasons for the controls may help the users to accept and adhere to the security restrictions built into the system.

Good Principles — Exploiting the Strengths of Personnel in Regard to a Security Program

A person should never be disciplined for following correct procedures. This may sound ridiculous, but it is a common weakness exploited by people as a part of social engineering. Millions of dollars' worth of security will be worthless if our staff is not trained to resist and report all social engineering attempts. Investigators have found that the easiest way to gather corporate information is through bribery or relationships with employees.

There are four main types of social engineering: intimidation, helpfulness, technical, and name-dropping. The principle of intimidation is the threat of punishment or ridicule for following correct procedures. The person being "engineered" is bullied by the attacker into granting an exception to the rules — perhaps due to position within the company or force of character. In many instances the security-minded person is berated by the attacker, threatened with discipline or loss of employment, or otherwise intimidated by a person for just trying to do their job. Some of the most serious breaches of secure facilities have been accomplished through these techniques. In one instance the chief financial officer of a corporation refused to comply with the procedure of wearing an ID card. When challenged by a new security person, the executive explained in a loud voice that he should never again be challenged to display an ID card. Such intimidation unnerved the security person to the point of making the entire security procedure ineffective and arbitrary. Such a "tone at the top" indicates a lack of concern for security that will soon permeate through the entire organization.

Helpfulness is another form of social engineering, appealing to the natural instinct of most people to want to provide help or assistance to another person. One of the most vulnerable areas for this type of manipulation is the help desk. Help desk personnel are responsible for password resets, remote access problem resolution, and system error handling. Improper handling of these tasks may result in an attacker getting a password reset for another legitimate user's account and creating either a security gap or a denial-of-service for the legitimate user.

Despite the desires of users, the help desk, and administrators to facilitate the access of legitimate users to the system, they must be trained to recognize social engineering and follow established secure procedures.

Name-dropping is another form of social engineering and is often facilitated by press releases, Web page ownership or administrator information, discarded corporate documentation, or other ways that an attacker can learn the names of individuals responsible for research, business operations, administrative functions, or other key roles. By using the names of these individuals in conversation, a hacker can appear to be a legitimate user or have a legitimate affiliation with the corporation. It has been quoted that "The greater the lie, the easier it is to convince someone that it is true." This especially applies to a name-dropping type of attack. Despite the prior knowledge of the behaviors of a manager, a subordinate may be influenced into performing some task at the request of an attacker although the manager would never have contemplated or approved such a request.

Technology has provided new forms of social engineering. Now an attacker can e-mail or fax a request to a corporation for information and receive a response that compromises security. This may be from a person alleging to represent law enforcement or some other government department demanding cooperation or assistance. The correct response must be to have an established manner of contact for outside agencies and train all personnel to route requests for information from an outside source through proper channels.

All in all, the key to immunizing personnel against social-engineering attacks is to emphasize the importance of procedure, the correctness of following and enforcing security protocols, and the support of management for personnel who resist any actions that attempt to circumvent proper controls and may be an incidence of social engineering. All employees must know that they will never lose their job for enforcing corporate security procedures.

Job Rotation

Job rotation is an important principle from a security perspective, although it is often seen as a detriment by project managers. Job rotation moves key personnel through the various functional roles in a department or even between departments. This provides several benefits, such as cross-training of key personnel and reducing the risks to a system through lack of trained personnel during vacations or illnesses. Job rotation also serves to identify possible fraudulent activity or shortcuts taken by personnel who have been in the job for an extended time period. In one instance, a corporation needed to take disciplinary action against an employee who was the administrator for a critically important system, not only for the business but also for the community. Because this administrator had sole knowledge of the system and the system administrator password, they were unable to take action in a timely manner. They were forced to delay any action until the administrator left for vacation and gave the password to a backup person.

When people stay in a position too long, they may become more attached to the system than to the corporation, and their activity and judgment may become impaired.

Anti-Virus and Web-Based Attacks

The connectivity of systems and the proliferation of Web-based attacks have resulted in significant damage to corporate systems, expenses, and productivity losses. Many people recognize the impact of Code Red and Nimda; however, even when these attacks were taken out of the calculations, the incidence of Web-based attacks rose more than 79 percent in 2001.[3] Some studies have documented more attacks in the first two months of 2002 than were detected in the previous year and a half.[4]

Users have heard many times not to open e-mail attachments; however, this has not prevented many infections and security breaches from happening. More sophisticated attacks — all of which can appear to come from trusted sources — are appearing, and today's firewalls and anti-virus products are not able to protect an organization adequately. Instead, users need to be more diligent to confirm with a sender whether they intended to send out an attachment prior to opening it. The use of instant messaging, file sharing, and other products, many of which exploit open ports or VPN tunnels through firewalls, is creating even more vulnerabilities. The use of any technology or new product should be subject to analysis and review by security before the users adopt it. This requires the security department to react swiftly to requests from users and be aware of the new trends, technologies, and threats that are emerging.

Segregation of Duties

The principle of segregation of duties breaks an operation into separate functions so that no one person can control a process from initiation through to completion. Instead, a transaction would require one person to input the data, a second person to review and reconcile the batch totals, and another person (or perhaps the first individual) to confirm the final portion of the transaction. This is especially critical in financial transactions or error handling procedures.

Summary

This is neither a comprehensive list of all the security concerns and ways to train and monitor the people in our organizations, nor is it a full list of all job roles and functions. Hopefully it is a tool that managers, security personnel, and auditors can use to review some of the procedures they have in place and create a better security infrastructure. The key objective of this chapter is to identify the primary roles that people play in the information security environment. A security program is only as good as the people implementing it, and a key realization is that tools and technology are not enough when it comes to protecting our organizations. We need to enlist the support of every member of our companies. We need to see the users, administrators, managers, and auditors as partners in security. Much of this is accomplished through understanding. When the users understand why we need security, the security people understand the business, and everyone respects the role of the other departments, then the atmosphere and environment will lead to greater security, confidence, and trust.

References

1. www.viruslist.com as reported in *SC INFOSECURITY* magazine, December 2001, p. 12.
2. www.oregon.gov, Secretary of State Audit of the Public Employees Benefit Board — also California Department of Motor Vehicles report on abandoning new system.
3. Cyber security, Claudia Flisi, *Newsweek,* March 18, 2002.
4. Etisalat Academy, March 2002.

56

Security Management

Ken Buszta, CISSP

It was once said, "Information is king." In today's world, this statement has never rung more true. As a result, information is now viewed as an asset; and organizations are willing to invest large sums of money toward its protection. Unfortunately, organizations appear to be overlooking one of the weakest links for protecting their information — the information security management team. The security management team is the one component in our strategy that can ensure our security plan is working properly and takes corrective actions when necessary. In this chapter, we address the benefits of an information security team, the various roles within the team, job separation, job rotation, and performance metrics for the team, including certifications.

Security Management Team Justification

Information technology departments have always had to justify their budgets. With the recent global economic changes, the pressures of maintaining stockholder values have brought IT budgets under even more intense scrutiny. Migrations, new technology implementations, and even staff spending have been either been delayed, reduced, or removed from budgets. So how is it that an organization can justify the expense, much less the existence, of an information security management team? While most internal departments lack the necessary skill sets to address security, there are three compelling reasons to establish this team:

1. *Maintain competitive advantage.* An organization exists to provide a specialized product or service for its clients. The methodologies and trade secrets used to provide these services and products are the assets that establish our competitive advantage. An organization's failure to properly protect and monitor these assets can result in the loss of not only a competitive advantage but also lost revenues and possible failure of the organization.
2. *Protection of the organization's reputation.* In early 2000, several high-profile organizations' Web sites were attacked. As a result, the public's confidence was shaken in their ability to adequately protect their clients. A security management team will not be able to guarantee or fully prevent this from happening, but a well-constructed team can minimize the opportunities made available from your organization to an attacker.
3. *Mandates by governmental regulations.* Regulations within the United States, such as the Health Insurance Portability and Accountability Act (HIPAA) and the Gramm-Leach-Bliley Act (GLBA) and those abroad, such as the European Convention on Cybercrime, have mandated that organizations protect their data. An information security management team, working with the organization's legal and auditing teams, can focus on ensuring that proper safeguards are utilized for regulatory compliance.

Executive Management and the IT Security Management Relationship

The first and foremost requirement to help ensure the success of an information security management team relies on its relationship with the organization's executive board. Commencing with the CEO and then working downward, it is essential for the executive board to support the efforts of the information security team. Failure

of the executive board to actively demonstrate its support for this group will gradually become reflected within the rest of the organization. Apathy toward the information security team will become apparent, and the team will be rendered ineffective. The executive board can easily avoid this pitfall by publicly signing and adhering to all major information security initiatives such as security policies.

Information Security Management Team Organization

Once executive management has committed its support to an information security team, a decision must be made as to whether the team should operate within a centralized or decentralized administration environment.

In a centralized environment, a dedicated team is assigned the sole responsibility for the information security program. These team members will report directly to the information security manager. Their responsibilities include promoting security throughout the organization, implementing new security initiatives, and providing daily security administration functions such as access control.

In a decentralized environment, the members of the team have information security responsibilities in addition to those assigned by their departments. These individuals may be network administrators or reside in such departments as finance, legal, human resources, or production.

This decision will be unique to each organization. Organizations that have identified higher risks deploy a centralized administration function. A growing trend is to implement a hybrid solution utilizing the best of both worlds. A smaller dedicated team ensures that new security initiatives are implemented and oversees the overall security plan of the organization, while a decentralized team is charged with promoting security throughout their departments and possibly handling the daily department-related administrative tasking.

The next issue that needs to be addressed is how the information security team will fit into the organization's reporting structure. This is a decision that should not be taken lightly because it will have a long-enduring effect on the organization. It is important that the organization's decision makers fully understand the ramifications of this decision. The information security team should be placed where its function has significant power and authority. For example, if the information security manager reports to management that does not support the information security charter, the manager's group will be rendered ineffective. Likewise, if personal agendas are placed ahead of the information security agenda, it will also be rendered ineffective. An organization may place the team directly under the CIO or it may create an additional executive position, separate from any particular department. Either way, it is critical that the team be placed in a position that will allow it to perform its duties.

Roles and Responsibilities

When planning a successful information security team, it is essential to identify the roles, rather than the titles, that each member will perform. Within each role, their responsibilities and authority must be clearly communicated and understood by everyone in the organization.

Most organizations can define a single process, such as finance, under one umbrella. There is a manager, and there are direct reports for every phase of the financial life cycle within that department. The information security process requires a different approach. Regardless of how centralized we try to make it, we cannot place it under a single umbrella. The success of the information security team is therefore based on a layered approach. As demonstrated in Exhibit 56.1, the core of any information security team lies with the executive management because they are ultimately responsible to the investors for the organization's success or failure. As we delve outward into the other layers, we see there are roles for which an information security manager does not have direct reports, such as auditors, technology providers, and the end-user community, but he still has an accountability report from or to each of these members.

It is difficult to provide a generic approach to fit everyone's needs. However, regardless of the structure, organizations need to assign security-related functions corresponding to the selected employees' skill sets. Over time, eight different roles have been identified to effectively serve an organization:

1. *Executive management.* The executive management team is ultimately responsible for the success (or failure) of any information security program. As stated earlier, without their active support, the information security team will struggle and, in most cases, fail in achieving its charter.
2. *Information security professionals.* These members are the actual members trained and experienced in the information security arena. They are responsible for the design, implementation, management, and review of the organization's security policy, standards, measures, practices, and procedures.

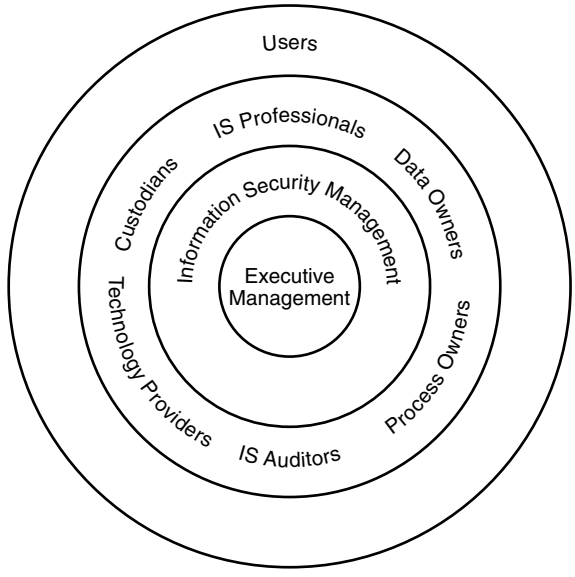


EXHIBIT 56.1 Layers of information security management team.

3. *Data owners.* Everyone within the organization can serve in this role. For example, the creator of a new or unique data spreadsheet or document can be considered the data owner of that file. As such, they are responsible for determining the sensitivity or classification levels of the data as well as maintaining the accuracy and integrity of the data while it resides in the system.
4. *Custodians.* This role may very well be the most under-appreciated of all. Custodians act as the owner's delegate, with their primary focus on backing up and restoring the data. The data owners dictate the schedule at which the backups are performed. Additionally, they run the system for the owners and must ensure that the required security controls are applied in accordance with the organization's security policies and procedures.
5. *Process owners.* These individuals ensure that the appropriate security, consistent with the organization's security policy, is embedded in the information systems.
6. *Technology providers.* These are the organization's subject matter experts for a given set of information security technologies and assist the organization with its implementation and management.
7. *Users.* As almost every member of the organization is a user of the information systems, they are responsible for adhering to the organization's security policies and procedures. Their most vital responsibility is maintaining the confidentiality of all usernames and passwords, including the program upon which these are established.
8. *Information systems auditor.* The auditor is responsible for providing independent assurance to management on the appropriateness of the security objectives and whether the security policies, standards, measures, practices, and procedures are appropriate and comply with the organization's security objectives. Because of the responsibility this role has in the information security program, organizations may shift this role's reporting structure directly to the auditing department as opposed to within the information security department.

Separation of Duties and the Principle of Least Privilege

While it may be necessary for some organizations to have a single individual serve in multiple security roles, each organization will want to consider the possible effects of this decision. By empowering one individual, it is possible for that person to manipulate the system for personal reasons without the organization's knowledge. As such, an information security practice is to maintain a separation of duties. Under this philosophy, pieces of a task are assigned to several people. By clearly identifying the roles and responsibilities, an organization

will be able to also implement the Principle of Least Privilege. This idea supports the concept that the users and the processes in a system should have the least number of privileges and for the shortest amount of time needed to perform their tasks.

For example, the system administrator's role may be broken into several different functions to limit the number of people with complete control. One person may become responsible for the system administration, a second person for the security administration, and a third person for the operator functions.

Typical system administrator/operator functions include:

- Installing system software
- Starting up and shutting down the system
- Adding and removing system users
- Performing backups and recovery
- Mounting disks and tapes
- Handling printers

Typical security administrator functions include:

- Setting user clearances, initial passwords, and other security clearances for new users, and changing security profiles for existing users
- Setting or changing the sensitivity file labels
- Setting security characteristics of devices and communication channels
- Reviewing audit data

The major benefit of both of these principles is to provide a *two-person control* process to limit the potential damage to an organization. Personnel would be forced into collusion in order to manipulate the system.

Job Rotation

Arguably, training may provide the biggest challenge to management, and many view it as a double-edged sword. On the one edge, training is viewed as an expense and is one of the first areas depreciated when budget cuts are required. This may leave the organization with stale skill sets and disgruntled employees. On the other edge, it is not unusual for an employee to absorb as much training from an organization as possible and then leave for a better opportunity. Where does management draw the line?

One method to address this issue is job rotation. By routinely rotating the job a person is assigned to perform, we can provide cross-training to the employees. This process provides the team members with higher skill sets and increased self-esteem; and it provides the organization with backup personnel in the event of an emergency.

From the information security point of view, job rotation has its benefits. Through job rotation, the collusion fostered through the separation of duties is broken up because an individual is not performing the same job functions for an extended period. Further, the designation of additionally trained workers adds to the personnel readiness of the organization's disaster recovery plan.

Performance Metrics

Each department within an organization is created with a charter or mission statement. While the goals for each department should be clearly defined and communicated, the tools that we use to measure a department's performance against these goals are not always as clearly defined, particularly in the case of information security. It is vital to determine a set of metrics by which to measure its effectiveness. Depending upon the metrics collected, the results may be used for several different purposes, such as:

- *Financial.* Results may be used to justify existing or increasing future budget levels.
- *Team competency.* A metric, such as certification, may be employed to demonstrate to management and the end users the knowledge of the information security team members. Additional metrics may include authorship and public speaking engagements.
- *Program efficiency.* As the department's responsibilities are increased, its ability to handle these demands while limiting its personnel hiring can be beneficial in times of economic uncertainty.

While in the metric planning stages, the information security manager may consider asking for assistance from the organization's auditing team. The auditing team can provide an independent verification of the metric results to both the executive management team and the information security department. Additionally, by getting the auditing department involved early in the process, it can assist the information security department in defining its metrics and the tools utilized to obtain them.

Determining performance metrics is a multi-step process. In the first step, the department must identify its process for metric collection. Among the questions an organization may consider in this identification process are:

- Why do we need to collect the statistics?
- What statistics will we collect?
- How will the statistics be collected?
- Who will collect the statistics?
- When will these statistics be collected?

The second step is for the organization to identify the functions that will be affected. The functions are measured as time, money, and resources. The resources can be quantified as personnel, equipment, or other assets of the organization.

The third step requires the department to determine the drivers behind the collection process. In the information security arena, the two drivers that affect the department's ability to respond in a timely manner are the number of system users and the number of systems within its jurisdiction. The more systems and users an organization has, the larger the information security department.

With these drivers in mind, executive management could rely on the following metrics with a better understanding of the department's accomplishments and budget justifications:

- Total systems managed
- Total remote systems managed
- User administration, including additions, deletions, and modifications
- User awareness training
- Average response times

For example, Exhibit 56.2 shows an increase in the number of system users over time. This chart alone could demonstrate the efficiency of the department as it handles more users with the same number of resources.

Exhibit 56.3 shows an example of the average information security response times. Upon review, we are clearly able to see an upward trend in the response times. This chart, when taken by itself, may pose some concerns by senior management regarding the information security team's abilities. However, when this metric is used in conjunction with the metrics found in Exhibit 56.2, a justification could be made to increase the information security personnel budget.

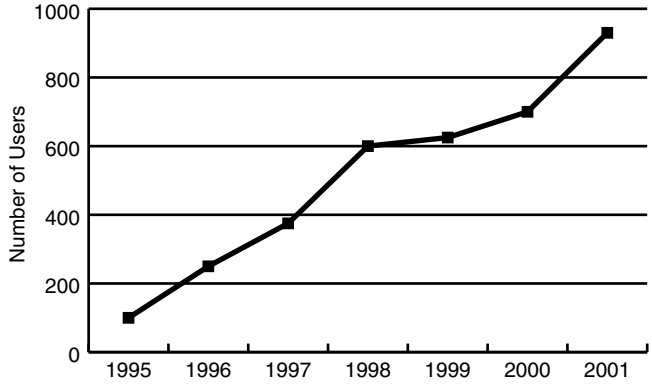


EXHIBIT 56.2 Users administered by information security department.

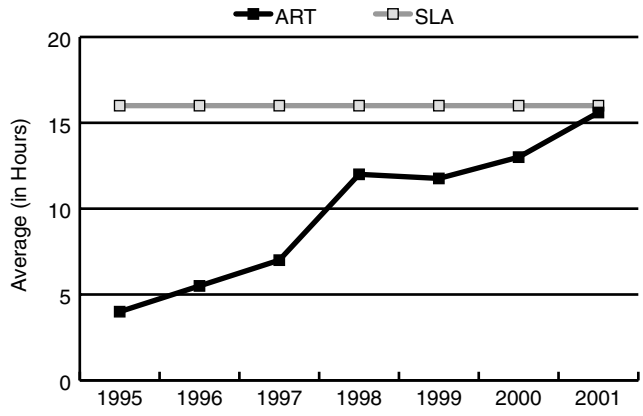


EXHIBIT 56.3 Average information security response times.

While it is important for these metrics to be gathered on a regular basis, it is even more important for this information to be shared with the appropriate parties. For example, by sharing performance metrics within the department, the department will able to identify its strong and weak areas. The information security manager will also want to share these results with the executive management team to perform a formal annual metric review and evaluation of the metrics.

Certifications

Using the various certification programs available is an effective tool for management to enhance the confidence levels in its security program while providing the team with recognition for its experience and knowledge. While there are both vendor-centric and vendor-neutral certifications available in today's market, we will focus only on the latter. (Note: The author does not endorse any particular certification program.)

Presently there is quite a debate about which certification is best. This is a hard question to answer directly. Perhaps the more important question is, "What do I want to accomplish in my career?" If based upon this premise, certification should be tailored to a set of objectives and therefore is a personal decision.

Certified Information Systems Security Professional (CISSP)

The CISSP Certification is an independent and objective measure of professional expertise and knowledge within the information security profession. Many regard this certification as an information security management certification. The credential, established over a decade ago, requires the candidate to have three years' verifiable experience in one or more of the ten domains in the Common Body of Knowledge (CBK) and pass a rigorous exam. The CBK, developed by the International Information Systems Security Certification Consortium $(ISC)^2$, established an international standard for IS security professionals. The CISSP multiple-choice certification examination covers the following ten domains of the CBK:

Domain 1: Access Control Systems and Methodology
Domain 2: Telecommunications and Network Security
Domain 3: Security Management Practices
Domain 4: Applications and Systems Development Security
Domain 5: Cryptography
Domain 6: Security Architecture and Models
Domain 7: Operations Security
Domain 8: Business Continuity Planning (BCP) and Disaster Recovery Planning (DRP)
Domain 9: Law, Investigations and Ethics
Domain 10: Physical Security

More information on this certification can be obtained by contacting (ISC)² through its e-mail address, info@isc2.org.

Systems Security Certified Practitioner (SSCP)

The SSCP certification focuses on information systems security practices, roles, and responsibilities defined by experts from major industries. Established in 1998, it provides network and systems security administrators with independent and objective measures of competence and recognition as a knowledgeable information systems security practitioner. Certification is only available to those individuals who have at least one year's experience in the CBK, subscribe to the (ISC)² Code of Ethics, and pass the 125-question SSCP certification examination, based on seven CBK knowledge areas:

1. Access Controls
2. Administration
3. Audit and Monitoring
4. Risk, Response and Recovery
5. Cryptography
6. Data Communications
7. Malicious Code/Malware

GIAC

In 1999, the SANS (System Administration, Networking, and Security) Institute founded the Global Information Assurance Certification (GIAC) Program to address the need to validate the skills of security professionals. The GIAC certification provides assurance that a certified individual holds an appropriate level of knowledge and skill necessary for a practitioner in key areas of information security. This is accomplished through a twofold process: practitioners must pass a multiple-choice exam and then complete a practical exam to demonstrate their ability to apply their knowledge. GIAC certification programs include:

- *GIAC Security Essentials Certification (GSEC).* GSEC graduates have the knowledge, skills, and abilities to incorporate good information security practice in any organization. The GSEC tests the essential knowledge and skills required of any individual with security responsibilities within an organization.
- *GIAC Certified Firewall Analyst (GCFW).* GCFWs have the knowledge, skills, and abilities to design, configure, and monitor routers, firewalls, and perimeter defense systems.
- *GIAC Certified Intrusion Analyst (GCIA).* GCIAs have the knowledge, skills, and abilities to configure and monitor intrusion detection systems and to read, interpret, and analyze network traffic and related log files.
- *GIAC Certified Incident Handler (GCIH).* GCIHs have the knowledge, skills, and abilities to manage incidents; to understand common attack techniques and tools; and to defend against or respond to such attacks when they occur.
- *GIAC Certified Windows Security Administrator (GCWN).* GCWNs have the knowledge, skills, and abilities to secure and audit Windows systems, including add-on services such as Internet Information Server and Certificate Services.
- *GIAC Certified UNIX Security Administrator (GCUX).* GCUXs have the knowledge, skills, and abilities to secure and audit UNIX and Linux systems.
- *GIAC Information Security Officer (GISO).* GISOs have demonstrated the knowledge required to handle the Security Officer responsibilities, including overseeing the security of information and information resources. This combines basic technical knowledge with an understanding of threats, risks, and best practices. Alternately, this certification suits those new to security who want to demonstrate a basic understanding of security principles and technical concepts.
- *GIAC Systems and Network Auditor (GSNA).* GSNAs have the knowledge, skills, and abilities to apply basic risk analysis techniques and to conduct a technical audit of essential information systems.

Certified Information Systems Auditor (CISA)

CISA is sponsored by the Information Systems and Audit Control Association (ISACA) and tests a candidate's knowledge of IS audit principles and practices, as well as technical content areas. It is based on the results of a practice analysis. The exam tests one process and six content areas (domains) covering those tasks that are routinely performed by a CISA. The process area, which existed in the prior CISA practice analysis, has been expanded to provide the CISA candidate with a more comprehensive description of the full IS audit process. These areas are as follows:

- Process-based area (domain)
- The IS audit process
- Content areas (domains)
- Management, planning, and organization of IS
- Technical infrastructure and operational practices
- Protection of information assets
- Disaster recovery and business continuity
- Business application system development, acquisition, implementation, and maintenance
- Business process evaluation and risk management

For more information, contact ISACA via e-mail: certification@isaca.org.

Conclusion

The protection of the assets may be driven by financial concerns, reputation protection, or government mandate. Regardless of the reason, well-constructed information security teams play a vital role in ensuring organizations are adequately protecting their information assets. Depending upon the organization, an information security team may operate in a centralized or decentralized environment; but either way, the roles must be clearly defined and implemented. Furthermore, it is crucial to develop a set of performance metrics for the information security team. The metrics should look to identify issues such as budgets, efficiencies, and proficiencies within the team.

References

Hutt, Arthur E. et al., *Computer Security Handbook*, 3rd ed., John Wiley & Sons, Inc., New York, 1995.

International Information Systems Security Certification Consortium (ISC)^2, www.isc2.org.

Information Systems and Audit Control Association (ISACA), www.isaca.org.

Kabay, Michel E., *The NCSA Guide to Enterprise Security: Protecting Information Assets*, McGraw-Hill, New York, 1996.

Killmeyer Tudor, Jan, *Information Security Architecture: An Integrated Approach to Security in the Organization*, Auerbach Publications, Boca Raton, FL, 2001.

Kovacich, Gerald L., *Information Systems Security Officer's Guide: Establishing and Managing an Information Protection Program,* Butterworth-Heinemann, Massachusetts, 1998.

Management Planning Guide for Information Systems Security Auditing, National State Auditors Association and the United States General Accounting Office, 2001.

Russell, Deborah and Gangemi, G.T. Sr., *Computer Security Basics*, O'Reilly & Associates, Inc., California, 1991.

System Administration, Networking, and Security (SANS) Institute, www.sans.org.

Stoll, Clifford, *The Cuckoo's Egg*, Doubleday, New York, 1989

Wadlow, Thomas A., *The Process of Network Security: Designing and Managing a Safe Network*, Addison-Wesley, Massachusetts, 2000.

Securing New Information Technology

Louis Fried

Payoff

New information technologies mean new information security risks. This article helps data center managers to keep up with new information technology and the security risks this technology presents.

Introduction

The job of the IS security specialist has gone from protecting information within the organization to protecting information in the extended enterprise. Controlled offices and plants have given way to a porous, multiconnected, global environment. The pace at which new information technology capabilities are being introduced in the corporate setting also creates a situation in which the potential of new security risks isn't well thought out. Data center managers and must be aware of these threats before adopting new technologies so that they can take adequate countermeasures.

Information security is concerned with protecting:

- The availability of information and information processing resources.

- The integrity and confidentiality of information.

Unless adequate protection is in place when new business applications are developed, one or both of these characteristics of information security may be threatened. Availability alone is a major issue. Among US companies, the cost of systems downtime has been placed by some estimates at $4 billion a year, with a loss of 37 million hours in worker productivity.

The application of information security methods has long been viewed as insurance against potential losses. Senior management has applied the principle that it should not spend more for insurance than the potential loss could cost. In many cases, management is balancing information security costs against the potential for a single loss incident, rather than multiple occurrences of loss. This fallacious reasoning can lead to a failure to protect information assets continuously or to upgrade that protection as technology changes and exposes new opportunities for losses.

Those who would intentionally damage or steal information also follow some basic economic principles. Amateur hackers may not place a specific value on their time and thus may be willing to put substantial effort into penetrating information systems. A professional clearly places an implicit value on time by seeking the easiest way to penetrate a system or by balancing potential profit against the time and effort necessary to carry out a crime. New technologies that create new (and possibly easier) ways to penetrate a system invite such professionals and fail to deter the amateurs.

This article describes some of the potential threats to information security that may arise in the next few years. The article concludes by pointing out the opportunities for employing new countermeasures.

Document Imaging Systems

The capabilities of document imaging systems include:

- Reading and storing images of paper documents.
- Character recognition of text for abstracting or indexing.
- Retrieval of stored documents by index entry.
- Manipulation of stored images.
- Appending notes to stored images (either text or voice).
- Workflow management tools to program the distribution of documents as action steps are needed.

Workflow management is critical to taking full advantage of image processing for business process applications in which successive or parallel steps are required to process the document. Successful applications include loan processing, insurance application or claims processing, and many others that depend on the movement of documents through review and approval steps.

Image processing usually requires a mainframe or minicomputer for processing any serious volume of information, though desktop and workstation versions also exist for limited use. In addition, a full image processing system requires document readers (i.e., scanners), a local area network (LAN), workstations or personal computers, and laser printer as output devices. It is possible to operate image processing over a Wide Area Network; however, because of the bandwidth required for reasonable response times, this is not usually done. As a result, most configurations are located within a single building or building complex.

Two years ago, an insurance company installed an imaging application for processing claims. The system was installed on a LAN linked to a minicomputer in the claims processing area. A manager who had received a layoff notice accessed the parameter-driven work-flow management system and randomly realigned the processing steps into new sequences, reassigning the process steps in an equally random fashion to the hundred or so claims processing clerks using the system. He then took the backup tapes, which were rotated weekly, and backed up the revised system files on all the tapes, replacing them in the tape cabinet. The individual did not steal any information or delete any information from the system. The next morning, he called the personnel department and requested that his final paycheck be sent to his home.

The cost to the insurance company? Tens of thousands of dollars in clerical time wasted and professional and managerial time lost in finding and correcting the problem. Even worse, there were weeks of delays in processing claims and handling the resultant complaint letters. No one at the company can estimate the loss of goodwill in the customer base.

Workflow management's weaknesses.

The very techniques of workflow management that make image processing systems so effective are also their Achilles' heel. Potential threats to image processing systems may come from disruption of the workflow by unauthorized changes to sequence or approval levels in workflow management systems or from the disruption of the workflow by component failure or damage. Information contained on documents may be stolen by the unauthorized copying (downloading of the image to the workstation)and release of document images by users of workstations.

These potential threats raise issues that must be considered in the use of image processing technology. The legal status of stored images may be questioned in court because of the potential for undetectable change. In addition, there are the threats to the business from loss of confidentiality of documents, loss of availability of the system during working hours, damage to the integrity of the images and notes appended to them, and questions about authenticity of stored documents.

Minisupercomputers

Massively parallel minisupercomputers are capable of providing relatively inexpensive, large computational capacity for such applications as signal processing, image recognition processing, or neural network processing.

Massively parallel processors are generally designed to work as attached processors or in conjunction with workstations. Currently available minisupercomputers can provide 4,096 processors for$85,000 or 8,192 processors for $150,000. They can interface to such devices as workstations, file servers, and LANs.

These machines can be an inexpensive computational resource for cracking encryption codes or computer-access codes; consequently, organizations that own them are well advised to limit access control for resource use to authorized users. This is especially true if the processor is attached to a mainframe with wide area network (WAN) connectivity. Such connectivity may allow unauthorized users to obtain access to the attached processor through the host machine.

Even without using a minisupercomputer but by simply stealing unauthorized time on conventional computers, a European hacker group bragged that it had figured out the access codes to all the major North American telephone switches. This allows them to make unlimited international telephone calls at no cost (or, if they are so inclined, to destroy the programming in the switches and deny service to millions of telephone users).

Neural Network Systems

Neural network systems are software (or hardware/software combinations) capable of heuristic learning within limited domains. These systems are an outgrowth of artificial intelligence research and are currently available at different levels of capacity on systems ranging from personal computers to mainframes.

With their heuristic learning capabilities, neural networks can learn how to penetrate a network or computer system. Small systems are already in the hands of hobbyists and hackers. The capability of neural networks programs will increase as greater amounts of main memory and processing power become easily affordable for desktop machines.

Wireless Local Area Networks

Wireless LANs support connectivity of devices by using radio frequency (RF) or infrared (IR) transmission between devices located in an office or office building. Wireless LANs consist of a LAN controller and signal generators or receivers that are either attached to devices or embedded in them. Wireless LANs have the advantage of allowing easy movement of connected devices so that office space can be reallocated or modified without the constraints of hard wiring. They can connect all sizes of computers and some peripherals. As portable computers become more intensively used, they can be easily connected to PCs or workstations in the office for transmission of files in either direction.

Wireless LANs may be subject to signal interruption or message capture by unauthorized parties. Radio frequency LANs operate throughout a transmitting area and are therefore more vulnerable than infrared transmission, which is line-of-sight only.

Among the major issues of concern in using this technology are retaining confidentiality and privacy of transmissions and avoiding business interruption in the event of a failure. The potential also exists, however, for other kinds of damage to wireless LAN users. For example, supermarkets are now experimenting with wireless terminals affixed to supermarket shopping carts that broadcast the price specials on that aisle to the shopper. As this technology is extended to the inventory control function and eventually to other functions in the store, it will not be long before some clever persons find a way to reduce their shopping costs and share the method over the underground networks.

WAN Radio Communications

WAN radio communications enable handheld or portable devices to access remote computers and exchange messages (including fax messages). Wireless wide area network (WAN) may use satellite transmission through roof-mounted antennas or regional radiotelephone technology. Access to wireless Wide Area Network is supported by internal radio modems in notebook and handheld computers or wireless modems/pagers on Personal Computer Memory Card International Association cards for optional use.

Many users think that telephone land lines offer some protection from intrusion because wiretaps can often be detected and tapping into a fiberoptic line is impossible without temporarily interrupting the service. Experience shows that most intrusions results from logical—not physical—attacks on networks. Hackers usually break in through remote maintenance ports on Private Branch eXchange, voice-mail systems, or remote-access features that permit travelers to place outgoing calls.

The threat to information security from the use of wireless wide area network (WAN) is that direct connectivity is no longer needed to connect to networks. Intruders may be able to fake legitimate calls once they have been able to determine access codes. Users need to consider such protective means as encryption for certain messages, limitations on the use of wireless wide area network (WAN) transmission for confidential material, and enforcement for encrypted password and user authentication controls.

Videoconferencing

Travel costs for nonsales activities is of growing concern to many companies. Companies are less concerned about the costs of travel and subsistence than they are about the costs to the company of having key personnel away from their jobs. Crossing the US or traveling to foreign countries for a one-day meeting often requires a key employee to be

away from the job for three days. Videoconferencing is increasingly used to reduce travel to only those trips that are essential for hands-on work.

The capabilities of videoconferencing include slow-scan video for sharing documents or interactive video for conferencing. Videoconferencing equipment is now selling for as little as$30,000 per installation. At that price, saving a few trips a year can quickly pay off. However, videoconferencing is potentially vulnerable to penetration of phone switches to tap open lines and receive both ends of the conferencing transmissions.

Protection against tapping lines requires additional equipment at both ends to scramble communications during transmission. It further requires defining when to scramble communications, making users aware of the risks, and enforcing rules.

Embedded Systems

Embedding computers into mechanical devices was pioneered by the military for applications ranging from autopilots on aircraft to smart bombs and missiles. In the civilian sector, process controls, robots, and automated machine tools were early applications. Manufacturers now embed intelligence and communications capabilities in products ranging from automobiles to microwave ovens. Computers from single-chip size to minicomputers are being integrated into the equipment that they direct. In factory automation systems, embedded systems are linked through LANs to area computers and to corporate hosts.

One security concern is that penetration of host computers can lead to penetration of automated factory units, which could interrupt productive capacity and create potential hazards for workers. In the past, the need for information security controls rarely reached the factory floor or the products that were produced because there was no connection to computers that resided on wide area network (WAN). Now, however, organizations must use techniques that enforce access controls and segment LANs on the factory floor to minimize the potential for unauthorized access through the company's host computers.

Furthermore, as computers and communications devices are used more in products, program bugs or device failure could endanger the customers who buy these products. With computer-controlled medical equipment or automobiles, for example, potential liability from malfunction may be enormous. Information security techniques must extend to the environment in which embedded systems software is developed to protect this software from corruption and the company from potential liability resulting from product failures.

PCMCIA Cards

PCMCIA cards are essentially small computer boards on which chips are mounted to provide memory and processing capacity. They can be inserted (i.e., docked) into slots on portable computers to add memory capacity, processing capacity, data base capacity, or communications functions such as pagers, electronic mail, or facsimile transmission. PCMCIA cards now contain up to 4M bytes of storage; by 1997, they can be expected to provide up to 20M bytes of storage in a 1.8-inch drive, can be inserted into portable devices with double Personal Computer Memory Card International Association card slots.

The small format of PCMCIA cards and their use in portable devices such as notebook or handheld computers makes them especially vulnerable to theft or loss. Such theft or loss can cause business interruption or breach of confidentiality through loss of the information contained on the card. In addition, poor work habits, such as failing to back up the data on another device, can result in the loss of data if the card fails or if the host device fails in a

manner that damages the card. Data recovery methods are notoriously nonexistent for small portable computers.

Smart Cards

Smart cards, consisting of a computer chip mounted on a plastic card similar to a credit card, have limited intelligence and storage compared to Personal Computer Memory Card International Association cards. Smart cards are increasingly used for health records, debit cards, and stored value cards. When inserted into an access device (reader), they may be used in pay telephones, transit systems, retail stores, health care providers, and Asynchronous Transfer Mode, as well as being used to supplement memory in handheld computers.

The risks in using this technology are the same as those for PCMCIA cards but may be exacerbated by the fact that smart cards can be easily carried in wallets along with credit cards. Because smart cards are used in stored value card systems, loss or damage to the card can deprive the owner of the value recorded. Both PCMCIA cards and smart cards must contain means for authenticating the user in order to protect against loss of confidentiality, privacy, or monetary value.

Notebook and Palmtop Computers

Notebook and palmtop computers are small portable personal computers, often supporting wireless connection to LANs and wide area network (WAN) or modems and providing communications capability for docking to desktop computers for uploading or downloading of files (either data or programs).

These devices have flat panel displays and may include 1.8-inch microdisks with 20M- to 80M-byte capacity. Some models support handwriting input. Smart cards, Personal Computer Memory Card International Association cards, or flashcards may be used to add functionality or memory. By the end of the decade, speech recognition capability should be available as a result of more powerful processors and greater memory capacity.

As with the cards that may be inserted into these machines, portable computers are vulnerable to loss or theft—both of the machine and of the information contained in its memory. In addition, their use in public places (such as on airplanes) may breach confidentiality or privacy.

It is vital that companies establish information security guidelines for use of these machines as they become ubiquitous. Guidelines should include means for authentication of the user to the device before it can be used, etching or otherwise imprinting the owner's name indelibly onto the machine, and rules for protected storage of the machine when it is not in the user's possession(as in travel or at hotel stays). One problem is that most hotel safes do not have deposit boxes large enough to hold notebook computers.

Portable computers combined with communications capability may create the single largest area of information security exposure in the future. Portable computers can go wherever the user goes. Scenarios of business use are stressing advantages but not security issues. Portable computers are used in many business functions including marketing, distribution field service, public safety, health care, transportation, financial services, publishing, wholesale and retail sales, insurance sales, and others. As the use of portable computers spreads, the opportunities for information loss or damage increase.

Portable computers, combined with communications that permit access to company data bases, require companies to adopt protective techniques to protect information bases from external access and prevent intelligence from being collected by repeated access. In

addition, techniques are needed for avoiding loss of confidentiality and privacy by device theft and business interruption through device failure.

New uses create new business vulnerabilities. New hospitals, for example, are being designed with patient-centered systems in which the services are brought to the patient (to the extent possible) rather than having the patient moved from one laboratory to another. This approach requires the installation of LANs throughout the hospital so that specialized terminals or diagnostic devices can be connected to the computers processing the data collected. Handheld computers may be moved with the patient or carried by attendants and plugged into the LAN to access patient records or doctors' orders. It is easy to anticipate abuses that range from illegal access to patient information to illegal dispensing of drugs to unauthorized persons.

New Opportunities for Defense

New technology should not, however, be seen solely as a security threat. New technology also holds opportunities for better means of protection and detection. Many capabilities provided by the IT department can support defensive techniques for information or information processing facilities.

Expert systems, neural networks, and minisupercomputers.
Used individually or in combination, these technologies may enable intrusion detection of information systems. These technologies can be used to recognize unusual behavior patterns on the part of the intruder, configure the human interface to suit individual users and their permitted accesses, detect physical intrusion or emergencies by signal analysis of sensor input and pattern recognition, and reconfigure networks and systems to maintain availability and circumvent failed components. In the future, these techniques may be combined with closed-circuit video to authenticate authorized personnel by comparing digitally stored images of persons wishing to enter facilities.

Smart cards or PCMCIA cards.
Used with card readers and carrying their own software data, data cards may enable authentication of a card owner through various means, including recognition of pressure, speed, and patterns of signatures; questions about personal history(the answers to which are stored on the card); use of a digitized picture of the owner; or cryptographic codes, access keys, and algorithms. Within five years, signature recognition capabilities may be used to limit access to penbased handheld computers to authorized users only, by recognizing a signature on log-in.

Personal computer networks (PCNs).
PCNs, enabled by nationwide wireless data communications networks, will permit a personal phone number to be assigned so that calls may reach individuals wherever they (and the instrument) are located in the US. PCNs will permit additional authentication methods and allow call-back techniques to work in a portable device environment.

Voice recognition.
When implemented along with continuous speech understanding, voice recognition may be used to authenticate users of voice input systems—for example, for inquiry systems in banking and brokerages. By the end of this decade voice recognition may be used to limit access to handheld computers to authorized users only by recognizing the owner's voice on log-in.

Wireless tokens.
Wireless tokens used as company identity badges can pinpoint the location of employees on plant sites and monitor restricted plant areas and work check-in and check-out. They may also support paging capability for messages or hazard warnings.

Reducing password risks.
The Obvious Password Utility System (OPUS)project at Purdue University has created a file compression technique that makes it possible to quickly check a proposed password against a list of prohibited passwords. With this technique, the check takes the same amount of time no matter how long the list. OPUS may allow prohibited password lists to be placed on small servers and improve password control so that systems are harder to crack.

Third-party authentication methods.
Systems like Kerberos and Sesame provide a third-party authentication mechanism that operates in an open network environment but does not permit access unless the user and the application are authenticated to each other by a separate, independent computer. (Third-party refers to a separate computer, not a legal entity.) Such systems may be a defense for the threats caused by portable systems and open networks. Users of portable computers may call the third-party machine and request access to a specific application on the remote host. The Kerberos or Sesame machine authenticates the user to the application and the application to the user before permitting access.

Conclusion

To stay ahead of the threats, data center managers must maintain a knowledge of technology advances, anticipate the potential threats and vulnerabilities, and develop the protective measures in advance. In well-run systems development functions, information security specialists are consulted during the systems specification and design phases to ensure that adequate provisions are made for the security of information in applications. Data center managers must be aware of the potential threats implicit in the adoption of new technologies and the defensive measures available in order to critique the design of new applications and to inform their senior management of hazards.

The combination of advanced computer capabilities and communications is making information available to corporate executives and managers on an unprecedented scale. The availability of information mandates its use by decision makers. Corporate officers could find that they are no longer just liable for prudent protection of the company's information assets but that they are liable for prudent use of the information available to the company in order to protect its customers and employees. Such conditions may alter the way systems are designed and information is used and the way the company chooses to protect its information assets.

Author Biographies

Louis Fried
Louis Fried is vice president of IT consulting at SRI International, Menlo Park CA.

59

Configuration Management: Charting the Course for the Organization

Mollie E. Krehnke, CISSP, IAM and David C. Krehnke, CISSP, CISM, IAM

Configuration management (CM) supports consistency, completeness, and rigor in implementing security. It also provides a mechanism for determining the current security posture of the organization with regard to technologies being utilized, processes and practices being performed, and a means for evaluating the impact of change on the security stance of the organization. If a new technology is being considered for implementation, an analysis can determine the effects from multiple standpoints:

- Costs to purchase, install, maintain, and monitor
- Positive or negative interactions with existing technologies or architectures
- Performance
- Level of protection
- Ease of use
- Management practices that must be modified to implement the technology
- Human resources who must be trained on the correct use of the new technology, as a user or as a provider

CM functions serve as a vital base for controlling the present — and for charting the future for an organization in meeting its goals. But looking at CM from a procedural level exclusively might result in the omission of significant processes that could enhance the information security stance of an organization and support mission success.

The Systems Security Engineering Capability Maturity Model (SSE-CMM)[1] will serve as the framework for the discussion of CM, with other long-standing, well-accepted references used to suggest key elements, policies, and procedural examples.

An Overview of the SSE-CMM

The SSE-CMM describes the essential characteristics of an organization's security engineering process that must exist to ensure good security engineering and thereby protect an organization's information resources, including hardware, software, and data. The SSE-CMM model addresses:

- The entire system life cycle, including concept definition, requirements analysis, design, development, integration, installation, operations, maintenance, and decommissioning activities

- The entire organization, including management, organizational, and engineering activities, and their staffs, including developers and integrators, that provide security services
- Concurrent interactions with other disciplines, such as systems, software, hardware, human factors, and testing engineering; system management, operation, and maintenance
- Interactions with other functions, including acquisition, system management, certification, accreditation, and evaluation
- All types and sizes of security engineering organizations — commercial, government, and academia[2]

SSE-CMM Relationship to Other Initiatives

Exhibit 59.1 shows how the SSE-CMM process relates to other initiatives working to provide structure, consistency, assurance, and professional stature to information systems security and security engineering.

EXHIBIT 59.1 Information Security Initiatives

Effort	Goal	Approach	Scope
SSE-CMM	Define, improve, and assess security engineering capability	Continuous security engineering maturity model and appraisal method	Security engineering organizations
SE-CMM	Improve the system or product engineering process	Continuous maturity model of systems engineering practices and appraisal method	Systems engineering organizations
SEI CMM for software	Improve the management of software development	Staged maturity model of software engineering and management practices	Software engineering organizations
Trusted CMM	Improve the process of high-integrity software development and its environment	Staged maturity model of software engineering and management practices, including security	High-integrity software organizations
CMM1	Combine existing process improvement models into a single architectural framework	Sort, combine, and arrange process improvement building blocks to form tailored models	Engineering organizations
Sys. Eng. CM (EIA731)	Define, improve, and assess systems engineering capability	Continuous system engineering maturity model and appraisal method	System engineering organizations
Common criteria	Improve security by enabling reusable protection profiles for classes of technology	Set of functional and assurance requirements for security, along with an evaluation process	Information technology
CISSP	Make security professional a recognized discipline	Security body of knowledge and certification test for security profession	Security practitioners
Assurance frameworks	Improve security assurance by enabling a broad range of evidence	Structured approach for creating assurance arguments and efficiently producing evidence	Security engineering organizations
ISO 9001	Improve organizational quality management	Specific requirements for quality management process	Service organizations
ISO 15504	Improve software process and assessment	Software process improvement model and appraisal methodology	Software engineering organizations
ISO 13335	Improve management of information technology security	Guidance on process used to achieve and maintain appropriate levels of security for information and services	Security engineering organizations

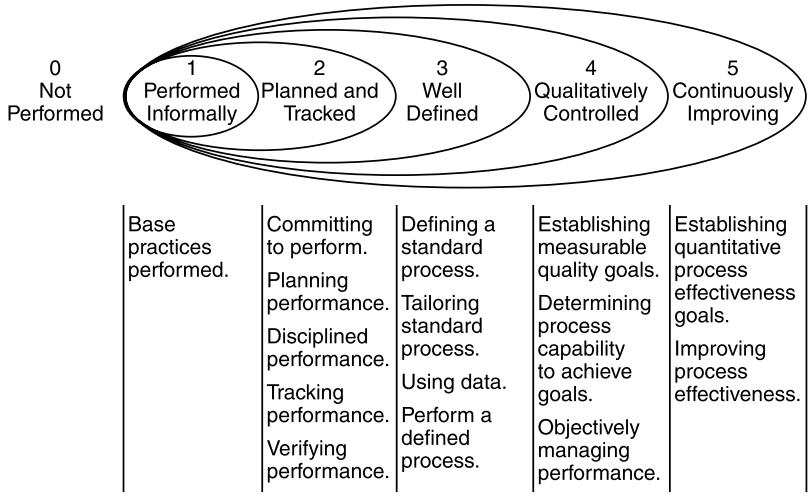


EXHIBIT 59.2 Capability levels of a security engineering organization.

CMM Framework

A CMM is a framework for evolving an security engineering organization from an ad hoc, less organized, less effective state to a highly structured effective state. Use of such a model is a means for organizations to bring their practices under statistical process control in order to increase their process capability. The SSE-CMM was developed with the anticipation that applying the concepts of statistical process control to security engineering will promote the development of secure systems and trusted products within anticipated limits of cost, schedule, and quality.

— SSE-CMM, Version 2.0, April 1, 1999

A process is a set of activities performed to achieve a given purpose. A well-defined process includes activities, input and output artifacts of each activity, and mechanisms to control performance of the activities. A defined process is formally described for or by an organization for use by its security professionals and indicates what actions are supposed to be taken. The performed process is what the security professionals actually do.…[P]rocess maturity indicates the extent to which a specific process is explicitly defined, managed, measured, controlled, and effective. Process maturity implies a potential for growth in capability and indicates both the richness of an organization's process and the consistency with which it is applied throughout the organization.

— SSE-CMM, Version 2.0, April 1, 1999, p. 21

Capability Levels Associated with Security Engineering Maturity

There are five capability levels associated with the SSE-CMM maturity model (see Exhibit 59.2) that represent increasing organizational capability. The levels are comprised of generic practices ordered according to maturity. Therefore, generic practices that indicate a higher level of process capability are located at the top of the capability dimension.

The SSE-CMM does not imply specific requirements for performing the generic practices. An organization is generally free to plan, track, define, control, and improve their processes in any way or sequence they choose. However, because some higher level generic practices are dependent on lower level generic practices, organizations are encouraged to work on the lower level generic practices before attempting to achieve higher levels.

— SSE-CMM, Version 2.0, April 1, 1999

CMM Institutionalization

Institutionalization is the building of an infrastructure and corporate culture that establishes methods, practices, and procedures, even after those who originally defined them are gone. The process capability side of the SSE-CMM supports institutionalization by providing practices and a path toward quantitative management and continuous improvement.[3] A mature, and continually improving, CM process and the associated base practices can result in activities with the following desirable qualities.

- C*ontinuity*: knowledge acquired in previous efforts is used in future efforts
- R*epeatability*: a way to ensure that projects can repeat a successful effort
- E*fficiency*: a way to help both developers and evaluators work more efficiently
- A*ssurance*: confidence that security needs are being addressed[4]

Security Engineering Model Goals

The SSE-CMM is a compilation of the best-known security engineering practices and is an evolving discipline. However, there are some general goals that can be presented. Many of these goals are also supported by the other organizations noted in Exhibit 59.1 that are working to protect an organization's information resources.

- Gain an understanding of the security risks associated with an enterprise.
- Establish a balanced set of security needs in accordance with identified risks.
- Transform security needs into security guidance to be integrated into the activities of other disciplines employed on a project and into descriptions of a system configuration or operation.
- Establish confidence or assurance in the correctness and effectiveness of security mechanisms.
- Determine that operational impacts due to residual security vulnerabilities in a system or its operation are tolerable (acceptable risks).
- Integrate the efforts of all security engineering disciplines and specialties into a combined understanding of the trustworthiness of a system.[5]

Security Engineering

While information technology security is often the driving discipline in the current security and business environment, the more traditional security disciplines should not be overlooked. These other security disciplines include the following:

- Operations security
- Information security
- Network security
- Physical security
- Personnel security
- Administrative security
- Communications security
- Emanation security
- Computer security

Security Engineering Process Overview

The security engineering process is composed of three basic areas: risk management, engineering, and assurance. The risk management process identifies and prioritizes dangers inherent in the developed product or system. The security engineering process works with the other engineering disciplines to determine and implement solutions to the problems presented by the dangers. The assurance process establishes confidence in the security solutions and conveys this confidence to customers or to management. These areas work together to ensure that the security engineering process results achieve the defined goals.

Risk Management

Risk management involves threats, vulnerabilities, and impacts. As an SSE-CMM process, risk management is the process of identifying and quantifying risk, and establishing an acceptable level of risk for the organization. The security practice areas in support of the risk management process are assess security risk, assess impact, and assess vulnerability.[6]

Engineering

Security engineers work with the customer to identify the security needs based on the identified risks, relevant laws, organizational policies, and existing information configurations. Security engineering is a process that proceeds through concept, design, implementation, test, deployment, operation, maintenance, and decommission. This process requires close cooperation and communication with other parts of the system engineering team to coordinate activities in the accomplishment of the required objectives, ensuring that security is an integral part of the process. Once the security needs are identified, security engineers identify and track specific requirements.[7]

The security practice areas in support of the engineering process are specify security needs, provide security input, administer security controls, and monitor security posture. Later in the life cycle, the security engineer is called on to ensure that products and systems are properly configured in relation to the perceived risks, ensuring that new risks do not make the system unsafe to operate.[8]

Assurance

Assurance is the degree of confidence that the security needs are satisfied. The controls have been implemented, will function as intended, and will reduce the anticipated risk. Often, this assurance is communicated in the form of an argument and is evident in documentation that is developed during the normal course of security engineering activities.

Security Engineering Basic Process Areas

The SSE-CMM contains approximately 60 security base practices, organized into 11 process areas that cover all major areas of security engineering, and represent the best existing practices of the security engineering community. Base practices apply across the life cycle of the enterprise, do not overlap with other base practices, represent a best practice of the security community (not a state-of-the-art technique), apply using multiple methods in multiple business contexts, and do not specify a particular method or tool. The 11 SSE-CMM process areas are listed below in alphabetical order to discourage the association of a practice with a life cycle phase.

- Administer security controls
- Assess impact
- Assess security risk
- Assess threat
- Assess vulnerability
- Build assurance argument
- Coordinate security
- Monitor security posture
- Provide security input
- Specify security needs
- Verify and validate security

Security Engineering Project and Organizational Practices

There are also 11 process areas related to project and organizational practices:

- Ensure quality
- Manage configuration
- Manage project risk
- Monitor and control technical effort
- Plan technical effort

- Define organization's system engineering process
- Improve organization's system engineering process
- Manage product line evolution
- Manage systems engineering support environment
- Provide ongoing skills and knowledge
- Coordinate with suppliers[9]

The base practices and the project and organizational practices were presented to provide the reader with a perspective for the focus of this chapter on the utilization and implementation configuration management — the topic of this chapter.

Configuration Management

This chapter follows the base practices associated with SSE-CMM PA 13 — Configuration Management to discuss policies, procedures, and resources that support this process in the establishment, implementation, and enhancement of security of an organization's information resources.

Configuration Management Description

The purpose of CM is to maintain data on and status of identified configuration units, and to analyze and control changes to the system and its configuration units. Managing the system configuration involves providing accurate and current configuration data and status to developers and customers. The goal is to maintain control over the established work product configurations.[10]

Configuration Management Base Practices

The following are the base practices considered essential elements of good security engineering CM:

- Establish CM methodology
- Identify configuration units
- Maintain work product baselines
- Control changes to established configuration units
- Communicate configuration status[11]

Each of these base practices is discussed below. The format presents the SSE-CMM description, example work products, and notes. Then a discussion of other references and resources that can be utilized to implement the base practice is presented.

Establish Configuration Management Methodology

Relationship to Other Security References

Choosing a CM tool to support the CM process will depend on the business processes being supported and the associated resources to be configured (see Exhibit 59.3). "Any information which may impact safety, quality, schedule, cost, or the environment must be managed. Each activity within the supply chain must be involved in the management process…. The best CM process is one that can best accommodate change and assure that all affected information remains clear, concise, and valid."[12]

Electronic Industries Alliance (EIA-649)

The Department of Defense and the Internal Revenue Service have adopted EIA-649 as their CM standard.

The CM process must relate to the context and environment in which it is to be implemented. Related activities include assignment of responsibilities, training of personnel, and determination of performance measurements. The Configuration Management Plan (CMP) can help to correlate CM to the International Standards Organization (ISO) 9000 series of quality systems criteria. The plan can also facilitate the justification of required resources and facilities, including automated tools.[13]

EXHIBIT 59.3 BP.13.01 — Establish CM Methodology

Description

Three primary trade-off considerations will have an impact on the structure and cost of CM, including:

- Level of detail at which the configuration units are identified
- Time when the configuration units are placed under CM
- Level of formalization required for the CM process

Example of Work Products

- Guidelines for identifying configuration units
- Timeline for placing configuration units under CM
- Selected CM process
- Selected CM process description

Notes

Selection criteria for configuration units should address interface maintenance, unique user requirements, new versus modified designs, and expected rate of change.

SSE-CMM, Version 2.0, April 1, 1999, p. 213–214.

Automated Tools

Institute of Configuration Management

There are several tools that have been certified by the Institute of Configuration Management (ICM)[14] because they can support a (new) configuration methodology (indicated as CMII) as defined by the ICM. The tools are listed in Exhibit 59.4.

The ICM certification signifies that:

- The tool supports achievement of the core elements of CMII functionality.
- The tool has the potential to be robust in all areas of functionality needed by that type of tool.
- The developer understands and agrees with the tool's strengths and weaknesses relative to CMII.
- The developer plans to make enhancements that will overcome those weaknesses.
- ICM agrees with the developer's priorities for doing so.[15]

Other Automated Tools

Another automated software management tool that is used in the IBM mainframe environment is ENDEVOR. The product can automate the transfer of all program source code, object code, executable code (load modules), interpretable code, control information, and the associated documentation to run a system. This includes source programs written in high-level programming language, job control or other control language, data dictionary, operating system, database components, online teleprocessing system, and job procedures.[16]

Two other commercially available online CM tools are UNIX's Source Code Control System (SCCS) and Revision Control System (RCS).[17]

EXHIBIT 59.4 ICM's CMII Certified Automated Tools

System Type	System Name	Release/ Version	Provider Name/Site	Date Certified
PDM	Metaphase	3.2	SDRD/Methphase www.SDRD.com	May 12, 2000
PDM	Axalant-CM	1.4	Usb/Eigner + Partner www.usbmuc.com www.ep-ag.com	December 8, 2000

Configuration Management Plan and Configuration Control Board as "Tools"

Computer Security Basics

This reference states that a manual tracking system can also be used for CM throughout a system's life cycle. Policies associated with CM implementation include:

- Assigning a unique identifier to each configuration item
- Developing a CMP
- Recording all changes to configuration items (either online or offline)
- Establishing a Configuration Control Board (CCB)[17]

EIA-649

Configuration identification is the basis of unique product identification, definition, and verification; product and document identification marking; change management; and accountability. The process enables a user to distinguish between product versions and supports release control of documents for baseline management.[18]

Information Systems Security Engineering Handbook

CM is a process for controlling all changes to a system (software, hardware, firmware, documentation, support/testing equipment, and development/maintenance equipment). A CCB should be established to review and approve any and all changes to the system. Reasons for performing CM throughout the life cycle of the information system include:

- Maintaining a baseline at a given point in the system life cycle
- Natural evolution of systems over time — they do not remain static
- Contingency planning for catastrophes (natural or human)
- Keeping track of all certification and accreditation evidence
- Use of the system's finite set of resources will grow through the system's life cycle
- Configuration item identification
- Configuration control
- Configuration accounting
- Configuration auditing[19]

NCSC-TG-006, A Guide to Understanding Configuration Management in Trusted Systems

The CMP and the human resources that support the CM process via the CCB should also be considered "tools." Effective CM should include a well-thought-out plan that should be prepared immediately after project initiation. The CMP should describe, in simple, positive statements, what is to be done to implement CM in the system.[20] CCB participants' roles should also be defined in the CMP. The responsibilities required by all those involved with the system should be established and documented in the CMP to ensure that the human element functions properly during CM.[21] A portion of the CMP should also address required procedures, and include routine CM procedures and any existing "emergency" procedures. Because the CMP is a living document, it should have the capability for additions and changes, but should be carefully evaluated and approved and then completely implemented to provide the appropriate assurances.

Any tools that will be used for CM should be documented in the CMP. These tools should be "maintained under strict configuration control." These tools can include forms used for change control, conventions for labeling configuration items, software libraries, as well as any automated tools that may be available to support the CM process. Samples of any documents to be used for reporting should also be contained in the CMP, along with a description of each.[21]

Information Systems Security Engineering Handbook, National Security Agency, Central Security Service.

Ensuring that a CM process is in place to prevent modifications that can cause an increase in security risk to occur without the proper approval is a consideration in the information system's life cycle, certification/accreditation, and recertification/reaccreditation activities after system activation.[22]

Identify Configuration Units

See Exhibits 59.5 and Exhibit 59.6.

EXHIBIT 59.5 BP.13.02 — Identify Configuration Units

Description

A configuration unit is one or more work products that are baselined together. The selection of work products for CM should be based on criteria established in the selected CM strategy. Configuration units should be selected at a level that benefits the developers and customers, but that does not place an unreasonable administrative burden on the developers.

Example of Work Products

- Baselined work product configuration
- Identified configuration units

Notes

Configuration units for a system that has requirements on field replacement should have an identified configuration unit at the field-replacement unit level.

SSE-CMM, Version 2.0, April 1, 1999, p. 215.

EXHIBIT 59.6 Examples of Configuration Units

The following examples of configuration units are cited in BP.01.02 — Manage Security Configuration:

- *Records of all software updates*: tracks licenses, serial numbers, and receipts for all software and software updates to the system, including date, person responsible, and a description of the change.
- *Records of all distribution problems*: describes any problem encountered during software distribution and how it was resolved.
- *System security configurations*: describes the current state of the system hardware, software, and communications, including their location, the individual assigned, and related information.
- *System security configuration changes*: describes any changes to the system security configuration, including the name of the person making the change, a description of the change, the reason for the change, and when the change was made.
- *Records of all confirmed software updates*: tracks software updates, which includes a description of the change, the name of the person making the change, and the date made.
- *Periodic summaries of trusted software distribution*: describes recent trusted software distribution activity, noting any difficulties and action items.
- *Security changes to requirements*: tracks any changes to system requirements made for security reasons or having an effect on security, to help ensure that changes and their effects are intentional.
- *Security changes to design documentation*: tracks any changes to the system design made for security reasons or having an effect on security, to help ensure that changes and their effects are intentional.
- *Control implementation*: describes the implementation of security controls within the system, including configuration details.
- *Security reviews*: describes the current state of the system security controls relative to the intended control implementation.
- *Control disposal*: describes the procedure for removing or disabling security controls, including a transition plan.

SSE-CMM, Version 2.0, April 1, 1999, p. 115–116.

Relationship to Other Security References

AR25-3, Army Life Cycle Management of Information Systems

CM focuses on four areas: configuration identification, configuration control, configuration status accounting, and configuration audit. CM should be applied throughout the life cycle of configuration items to control and improve the reliability of information systems.[23]

British Standards (BS7799), Information Security Management, Part 1, Code of Practice for Information Security Management Systems

A lack of change control is said to be a "common cause of system or security failures." Formal management and practice of change control are required for equipment, software, or procedures.[24]

Computer Security Basics

CM items also include documentation, test plans, and other security-related system tools and facilities.[25]

DOD-STD-2167A, Defense System Software Development.

Although this military standard has been canceled, the configuration identification units are a familiar concept to many system developers: computer software configuration items (CSCIs) and the corresponding computer software components (CSCs) and the computer software units (CSUs). Documentation established the Functional, Allocated, and Product Baselines. Each deliverable item had a version, release, change status, and other identification details. Configuration control was implemented through an established plan that was documented and then communicated through the implementation of configuration status accounting.

EIA-649

Unique identifiers support the correlation of the unit to a process, date, event, test, or document. Even documents must be uniquely identified to support association with the proper product configuration. The baseline represents an agreed-upon description of the product at a point in time with a known configuration. Intermediate baselines can be established for complex products. Baselines are the tools to match the need for consistency with the authority to approve changes. Baselines can include requirements, design releases, product configurations, operational, and disposal phase baselines.[26]

"Information Classification: A Corporate Implementation Guide," *Handbook of Information Security Management*

Maintaining an audit/history information that documents the software changes, "such as the work request detailing the work to be performed, who performed the work, and other pertinent documentation required by the business" is a vital software control.[17]

Maintain Work Product Baselines

See Exhibit 59.7.

Relationship to Other Security References

EIA-649

Recovery of a configuration baseline (or creation after the fact, with no adequate documentation) will be labor intensive and expensive. Without design and performance information, configuration must be determined via inspection, and this impacts operational and maintenance decisions. Reverse-engineering is a very expensive process.[26]

"Information Classification: A Corporate Implementation Guide," *Handbook of Information Security Management*

This chapter emphasizes the importance of version and configuration control, including "versions of software checked out for update, or being loaded to staging or production libraries. This would include the monitoring of error reports associated with this activity and taking appropriate corrective action."[28]

EXHIBIT 59.7 BP13.03 — Maintain Work Product Baselines

Description

This practice involves establishing and maintaining a repository of information about the work product configuration. …capturing data or describing the configuration units … including an established procedure for additions, deletions, and modifications to the baseline, as well as procedures for tracking/monitoring, auditing, and the accounting of configuration data … to provide an audit trail back to source documents at any point in the system life cycle.

Example of Work Products

- Decision database
- Baselined configuration
- Traceability matrix

Notes

Configuration data can be maintained in an electronic format to facilitate updates and changes to supporting documentation.[38]

SSE-CMM, Version 2.0, April 1, 1999, p. 216.

New Alliance Partnership Model (NAPM)

NAPM is a partnership model that combines security, configuration management, and quality assurance functions with an overall automated information system (AIS) security engineering process. NAPM provides insight into the importance of CM to the AISs of the organization and the implementation of an effective security program.

> CM provides management with the assurance that changes to an existing AIS are performed in an identifiable and controlled environment and that these changes do not adversely affect the integrity or availability properties of secure products, systems, and services. CM provides additional security assurance levels in that all additions, deletions, or changes made to a system do not compromise its integrity, availability, or confidentiality. CM is achieved through proceduralization and unbiased verification, ensuring that changes to an AIS and/or all supporting documentation are updated properly, concentrating on four components: identification, change control, status accounting, and auditing.[29]

Control Changes To Established Configuration Units

See Exhibit 59.8.

Relationship to Other Security References

British Standards (BS7799), Information Security Management, Part 1, Code of Practice for Information Security Management Systems

The assessment of the potential impact of a change, adherence to a procedure for approval of proposed changes, and procedures for aborting and recovering from unsuccessful changes play a significant role in the operational change process.[30] Policies and procedures to support software control and reduce the risk of operational systems corruption include:

- Program library updates by the nominated librarian with IT approval
- Exclusion of nonexecutable code
- In-depth testing and user acceptance of new code
- Updating of program source libraries
- Maintenance of an update audit log for all operational program libraries
- Retention of previous versions of software for contingencies[31]

EXHIBIT 59.8 BP.13.04 — Control Changes to Established Configuration Units

Description

Control is maintained over the configuration of the baselined work product. This includes tracking the configuration of each of the configuration units, approving a new configuration, if necessary, and updating the baseline. Identified problems with the work product or requests to change the work product are analyzed to determine the impact that the change will have on the work product, program schedule and cost, and other work products. If, based on analysis, the proposed change to the work product is accepted, a schedule is identified for incorporating the change into the work product and other affected areas. Changed configuration units are released after review and formal approval of configuration changes. Changes are not official until they are released.

Example of Work Products

- New work product baselines

Notes

Change control mechanisms can be tailored to categories of change. For example, the approval process should be shorter for component changes that do not affect other components.

SSE-CMM, Version 2.0, April 1, 1999, p. 217.

British Standards (BS7799), Information Security Management, Part 2, Specification for Information Security Management Systems

Formal change control procedures should be implemented for all stages of a system's life cycle, and these changes should be strictly controlled.[32]

EIA-649

> The initial baseline for change management consists of the configuration documentation defining the requirements that the performing activity (i.e., the product developer or product supplier) has agreed to meet. The design release baseline for change management consists of the detail design documentation used to manufacture, construct, build, or code the product. The product configuration baseline for change management consists of the detailed design documentation from the design release baseline which defines the product configuration that has been proven to meet the requirements for the product. The product configuration is considered [to be] a mature configuration. Changes to the current requirements, design release, or product configuration baselines may result from discovery of a problem, a suggestion for product improvement or enhancement, a customer request, or a condition dictated by the marketplace or by public law.
>
> Changes should be classified as major or minor to support the determination of the appropriate levels of review and approval. A major change is a change to the requirements of baselined configuration documentation (requirements, design release or product configuration baselines) that has significant impact. It requires coordination and review by all affected functional groups or product development teams and approval by a designated approval authority…. A minor change corrects or modifies configuration documentation (released design information), processes or parts but does not impact…customer requirements.
>
> To adequately evaluate a request for change, the change request must be clearly documented. It is important to accurately describe even minor changes so that an audit trail can be constructed in the event that there are unanticipated consequences or unexpected product failures. Saving the cost of the research involved in one such incident by having accurate accessible records may be sufficient to fully offset diligent, disciplined change processing.[33]

Technical, support, schedule, and cost impacts of a requested change must also be considered prior to approval and implementation. The organizational areas that will be impacted by the change or have the responsibility for implementing the change must be involved in the change process. Those organizations may have significant information (not available to other organizations) that could impact the successful implementation of a change. Change considerations must include the timeline and resource requirements of support organizations, as well as those of the primary client organization (e.g., update of support software, availability of spare and repair parts,

or revisions to operating and maintenance instructions) and urgency of the change. The change must be verified to ensure that the product, its documentation, and the support elements are consistent. The extent to which the verification is implemented will depend on the quantity of units changed and the type of change that is implemented. Records must be maintained regarding the verification of changes and implementation of required support functions. Variances to required configuration must be approved and documented.[33]

FIPS PUB 102, Guideline for Computer Security Certification and Accreditation

The change control process is an implicit form of recertification and reaccreditation. It is required during both development and operation. For sensitive applications, change control is needed for requirements, design, program, and procedural documentation, as well as for the hardware and software itself.

The process begins during development via the establishment of baselines for the products listed above. Once a baseline is established, all changes require a formal change request and authorization. Every change is reviewed for its impact on prior certification evidence.

An entity sometimes formed to oversee change control is the CCB. During development, the CCB is a working group subsidiary to the Project Steering Committee or its equivalent. Upon completion of development, CCB responsibility is typically transferred to an operations and maintenance office. There should be a security representative on the CCB responsible for the following:

- Deciding whether a change is security relevant
- Deciding on a required security review and required levels of recertification and reaccreditation
- Deciding on a threshold that would trigger recertification activity
- Serving as technical security evaluator, especially for minor changes that might receive no other security review

For very sensitive applications, it is appropriate to require approval and testing for all changes. However minor, a record must be kept of all changes as well as such pertinent certification evidence as test results. This record is reviewed during recertification.[33]

> As security features of a system or its environment change, recertification and reaccreditation are needed…. CM is a suitable area in which to place the monitoring activity for these changes.[34]

Information Systems Security Engineering Handbook

> A change or upgrade in the system, subsystem, or component configuration (e.g., incorporation of new operating system releases, modification of an applications program for data management, installation of a new commercial software package, hardware upgrades or swapouts, new security products, change to the interface characteristics of a 'trusted' component) … may violate its security assumptions.[35] The strongest configuration control procedures will include provisions for periodic physical and functional audit on the actual system in its operational environment. They will not rely solely on documentation or known or proposed changes. Changes frequently occur that are either not well known, or not well documented. These will only be detected by direct inspection of the system hardware, software, and resident data.[36]

NCSC-TG-006, A Guide to Configuration Management in Trusted Systems. CM maintains control of a system throughout its life cycle, ensuring that the system in operation is the correct system, and implementing the correct security policy. The Assurance Control Objective can be applied to configuration management as follows:

> Computer systems that process and store sensitive or classified information depend on the hardware and software to protect that information. It follows that the hardware and software themselves must be protected against unauthorized changes that could cause protection mechanisms to malfunction or be bypassed entirely. Only in this way can confidence be provided that the hardware and software interpretation of the security policy is maintained accurately and without distortion.[36]

Communicate Configuration Status

The status of the configuration is vital to the success of the organization (see Exhibit 59.9). The information that an organization uses must be accurate. "What is the sense of building the product to Six Sigma[37] when the blueprint is wrong?"[38] Changes must be documented and communicated in an expeditious and consistent manner.

EXHIBIT 59.9 BP.13.05 — Communicate Configuration Status

Description

Inform affected groups of the status of configuration data whenever there are any status changes. The status reports should include information on when accepted changes to configuration units will be processed, and the associated work products that are affected by the change. Access to configuration data and status should be provided to developers, customers, and other affected groups.

Example of Work Products

- Status reports

Notes

Examples of activities for communicating configuration status include providing access permissions to authorized users, and making baseline copies readily available to authorized users.

SSE-CMM, Version 2.0, April 1, 1999, p. 218.

Relationship to Other Security References

EIA-649

Configuration management information about a product is important throughout the entire life cycle, and the associated CM processes (planning and management, identification, change management, and verification and audit). "Configuration status accounting (CSA) correlates, stores, maintains, and provides readily available views of this organized collection of information…. CSA improves capabilities to identify, produce, inspect, deliver, operate, maintain, repair, and refurbish products."[39] CSA also provides "a source for configuration history of a product and all of its configuration documentation."

This CSA information must be disseminated to those who have a need to know throughout the product's life cycle. Examples of CSA life cycle documentation by phase include the following.

- *Conception phase*: requirements documents and their change history
- *Definition phase*: detailed configuration documents (e.g., specifications, engineering drawings, software design documents, software code, test plans and procedures) and their change history and variance status
- *Build phase*: additional product information (e.g., verified as-built unit configuration) and product changes, and associated variances
- *Distribution phase*: information includes customers and dates of delivery, installation configuration, warranty expiration dates, and service agreement types and expiration
- *Operation phase*: CSA varies, depending on the type of product and the contractual agreements regarding CM responsibilities, but could include product as-maintained and as-modified configurations, operation and maintenance information revision status, change requests and change notices, and restrictions
- *Disposal phase*: CSA information varies with the product and whether disposing of a product could have adverse implications, or if there are legal or contractual statues regarding retention of specific data[40]

"Systems Integrity Engineering," Handbook of Information Security Management

This chapter emphasizes the importance of configuration management plans to convey vital system-level information to the organization. Distributed system CM plans must document:

- System-level and site-level policies, standards, procedures, responsibilities, and requirements for the overall system control of the exchange of data
- The identification of each individual's site configuration
- Common data, hardware, and software
- The maintenance of each component's configuration

Distribution controls and audit checks to ensure common data and application versions are the same across the distributed system in which site-level CM plans are subordinate to distributed-level CM plans. The change control authority(ies) will need to establish agreements with all distributed systems on policies, standards,

procedures, roles, responsibilities, and requirements for distributed systems that are not managed by a single organizational department, agency, or entity.[41]

Conclusions

Change Is Inevitable

Change is inevitable in an organization. Changes in an information system, its immediate environment, or a wider organizational environment can (and probably will) impact the appropriateness of the information system's security posture and implemented security solutions. Routine business actions or events that can have a significant impact on security include:

- A mission or umbrella policy driven change in information criticality or sensitivity that causes a changes in the security needs or countermeasures required
- A change in the threat (e.g., changes in threat motivation, or new threat capabilities of potential attackers) that increases or decreases systems security risk
- A change in the application that requires a different security mode of operation
- A discovery of a new means of security attack
- A breach of security, a breach of system integrity, or an unusual situation or incident that appears to invalidate the accreditation by revealing a security flaw
- A security audit, inspection, or external assessment
- A change or upgrade in the system, subsystem, or component configurations
- The removal or degradation of a configuration item
- The removal or degradation of a system process countermeasure (i.e., human interface requirement or other doctrine/procedure components of the overall security solution)
- The connection to any new external interface
- Changes in the operational environment (e.g., relocation to other facilities, changes in infrastructure/environment-provided protections, changes in external operational procedures)
- Availability of new countermeasures technology that could, if used, improve the security posture or reduce operating costs
- Expiration of the information system's security accreditation statement[42]

Change Must Be Controlled

With the concept of control comes the concept of prior approval before changes are made. The approval is based on an analysis of the implications if the changes are made. It is possible that some changes may inadvertently change the security stance of the information system.

CM that is implemented according to an established plan can provide many benefits to an organization, including:

- Decisions based on knowledge of the complete change impact
- Changes limited to those that are necessary or offer significant benefits
- Effective cost-benefit analysis of proposed changes
- High levels of confidence in the product information or configurations
- Orderly communication of change information
- Preservation of customer interests
- Current system configuration baselines
- Configuration control at product interfaces
- Consistency between system configurations and associated documentation
- Ease of system maintenance after a change[43]

Change control must also be implemented within the computing facility. Every computing facility should have a policy regarding changes to operating systems, computing equipment, networks, environmental facilities (e.g., air conditioning, water, heat, plumbing, electricity, and alarms), and applications.[44]

Configuration Management as a Best Practice

The European Security Forum has been conducting systematic case studies of companies across various economic sectors for a number of years. A recent study addressed organizing and managing information technology (IT) security in a distributed environment. Change management for live systems was the fifth most important security practice worthy of additional study indicated by those organizations queried. Although the practice was well established and deemed of high importance by all respondents — as reported by the IT security manager, the IT manager, and a business manager of a functional area for each company — their comments resulted in the following finding. "While examples of successful practice exist, the general feeling that change management was an area where even the best organization recognized the need for improvement."[45]

Configuration Management as a Value-Adding Process

CM as a process enables an organization to tailor the process to address the context and environment in which the process will be implemented and add value to the resulting product. Multiple references reviewed for this chapter emphasized the need for consistency in how the process is implemented and its repeatability over time. It is better for an organization to consistently repeat a few processes over time than to inconsistently implement a multitude of activities once or twice. With standardization comes the knowledge of the status of that process. With knowledge of the status and the related benefits (and drawbacks), there can be a baseline of the process and its products. Effectively implementing configuration management can result in improved performance, reliability, or maintainability; extended life for the product; reduced development costs; reduced risk and liability; or corrected defects. The attributes of CM best practices include planned, integrated, consistent, rule/workflow-based, flexible, measured, and transparent.[46]

Security advantages of CM include protection against unintentional threats and malicious events. Not only does CM require a careful analysis of the implications of the proposed changes and approval of all changes before they are implemented, but it also provides a capability for reverting to a previous configuration (because previous versions are archived), if circumstances (e.g., a faulty change) require such an action. Once a reviewed program is accepted, a programmer is not permitted to make any malicious changes, such as inserting trapdoors, without going through the change approval process where such an action should be caught.[47]

Implementing Configuration Management

When implementing configuration management, the security professional should:

- Plan CM activities based on sound CM principles
- Choose a CM process that fits the environment, external constraints, and the product's life cycle phases
- Choose tools that support the CM process; tools can be simple and manual, or automated, or a combination of both
- Implement CM activities consistently across project and over time
- Use the CM plan as a training tool for personnel, and a briefing tool to explain the process to customers, quality assurance staff, and auditors
- Use enterprise CM plans to reduce the need for complete CM plans for similar products
- Ensure resources are available to support the process in a timely and accurate manner
- Ensure a security representative is on the CCB to evaluate the security implications of the proposed changes
- Ensure the changed system is tested and approved prior to deployment
- Ensure support/service areas are able to support the change
- Ensure configuration information is systematically recorded, safeguarded, validated, and disseminated
- Perform periodic audits to verify system configurations with the associated documentation, whether hardcopy or electronic in format

Notes

1. The Systems Security Engineering Capability Maturity Model (SSE-CMM) is a collaborative effort of Hughes Space and Communications, Hughes Telecommunications and Space, Lockheed Martin, Software Engineering Institute, Software Productivity Consortium, and Texas Instruments Incorporated.
2. SSE-CMM, Version 2.0, April 1, 1999, p. 2–3.

3. *Ibid.*, p. 22.
4. *Ibid.*, p. 6.
5. *Ibid.*, p. 26.
6. *Ibid.*, p. 31.
7. *Op cit.*
8. SSE-CMM, Version 2.0, April 1, 1999, p. 32.
9. *Ibid.*, p. 38.
10. *Ibid.*, p. 211.
11. *Ibid.*, p. 211.
12. To Fix CM Begins with Proper Training, *ICM Views,* ICM Web site, Institute of Configuration Management, P.O. Box 5656, Scottsdale, AZ 85261-5656, (840) 998-8600, info@icmhq.com.
13. EIA-649, National Consensus Standard for Configuration Management, Electronic Industries Alliance, August 1998, p. 9–12.
14. Institute of Configuration Management, P.O. Box 5656, Scottsdale, AZ 85261-5656, (840) 998-8600, info@icmhq.com.
15. *Configuration Management (CM) Resource Guide*, edited by Steve Easterbrook, is available at http://www.quality.org/config/cm-guide.html.
16. *CISSP Examination Textbooks, Volume 1: Theory,* first edition, S. Rao Vallabhaneni, SRV Professional Publications, 2000, p. 135.
17. *Computer Security Basics,* Deborah Russell and G. T. Gangemi, Sr., O'Reilly & Associates, Inc., 1991, p. 146.
18. EIA-649, National Consensus Standard for Configuration Management, Electronic Industries Alliance, August 1998, p. 14.
19. *Information Systems Security Engineering Handbook,* Release 1.0, National Security Agency, Central Security Service, February 28, 1994, p. 3-48-49.
20. *A Guide to Understanding Configuration Management in Trusted Systems*, National Computer Security Center, NCSC-TG-006, Version 1, 28 March 1988, p. 12, 13.
21. *Op. Cit.,* p. 12.
22. *Information Systems Security Engineering Handbook,* Release 1.0, National Security Agency, Central Security Service, February 28, 1994, p. 3-46.
23. AR25-3, Army Life Cycle Management of Information Systems, 9 June 1988, p. 36.
24. BS7799, British Standards 7799, Information Security Management, Part 1, Code of Practice for Information Security Management Systems, 1995, Section 6.2.4.
25. *Computer Security Basics,* Deborah Russell and G. T. Gangemi, Sr., O'Reilly & Associates, Inc., 1991, p. 145.
26. EIA-649, National Consensus Standard for Configuration Management, Electronic Industries Alliance, August 1998, p. 18-22.
27. Information Classification: A Corporate Implementation Guide, in *Handbook of Information Security Management*, 1999, p. 344.
28. Information Classification: A Corporate Implementation Guide, in Handbook of *Information Security Management,* 1999, p. 344
29. Systems Integrity Engineering, in *Handbook of Information Security Management,* 1999, p. 634.
30. British Standards (BS7799), Information Security Management, Part 1, Code of Practice for Information Security Management Systems, 1995, p. 19.
31. Ibid., p. 36.
32. British Standards (BS7799), Information Security Management, Part 2, Specification for Information Security Management Systems, 1998, p. 8.
33. EIA-649, National Consensus Standard for Configuration Management, Electronic Industries Alliance, August 1998, p. 24–34.
34. FIPS PUB 102, Performing Certification and Accreditation, Section 2.7.3, Change Control, p. 54
35. FIPS PUB 102, p. 9.
36. *Information Systems Security Engineering Handbook,* Release 1.0, National Security Agency, Central Security Service, February 28, 1994, p. 3-49.
37. Six Sigma — The Breakthrough Management Strategy Revolutionizing the World's Top Corporations, Mikel Harry and Richard Schroeder, Six Sigma Academy @2000.

38. What is Software CM?, *ICM Views,* ICM Web site, *Op.cit.*
39. EIA-649, National Consensus Standard for Configuration Management, Electronic Industries Alliance, August 1998, p. 34.
40. EIA-649, National Consensus Standard for Configuration Management, Electronic Industries Alliance, August 1998, p. 35-38.
41. Systems Integrity Engineering, in *Handbook of Information Security Management,* 1999, p. 628.
42. *Information Systems Security Engineering Handbook,* Release 1.0, National Security Agency, Central Security Service, February 28, 1994, p. 3-47.
43. EIA-649, National Consensus Standard for Configuration Management, Electronic Industries Alliance, August 1998, p. 23.
44. Systems and Operations Controls, *Handbook of Information Security Management,* 1993, p. 399.
45. Best Business Practice: Organising and Managing IT Security in a Distributed Environment, *European Security Forum,* September 1991, p. 38.
46. EIA-649, National Consensus Standard for Configuration Management, Electronic Industries Alliance, August 1998, p. 11.
47. *Security in Computing*, Charles P. Pfleeger, Englewood Cliffs, NJ: Prentice Hall, 1989.

60

Information Classification: A Corporate Implementation Guide

Jim Appleyard

Introduction

Classifying corporate information based on business risk, data value, or other criteria (as discussed later in this chapter), makes good business sense. Not all information has the same value or use, or is subject to the same risks. Therefore, protection mechanisms, recovery processes, etc. are — or should be — different, with differing costs associated with them. Data classification is intended to lower the cost of protecting data, and improve the overall quality of corporate decision making by helping ensure a higher quality of data upon which the decision makers depend.

The benefits of an enterprisewide data classification program are realized at the corporate level, not the individual application or even departmental level. Some of the benefits to the organization inclide:

- Data confidentiality, integrity, and availability are improved because appropriate controls are used for all data across the enterprise.
- The organization gets the most for its information protection dollar because protection mechanisms are designed and implemented where they are needed most, and less costly controls can be put in place for noncritical information.
- The quality of decisions is improved because the quality of the data upon which the decisions are made has been improved.
- The company is provided with a process to review all business functions and informational requirements on a periodic basis to determine priorities and values of critical business functions and data.
- The implementation of an information security architecture is supported, which better positions the company for future acquisitions and mergers.

This chapter will discuss the processes and techniques required to establish and maintain a corporate data classification program. There are costs associated with this process; however, most of these costs are front-end start-up costs. Once the program has been successfully implemented, the cost savings derived from the new security schemes, as well as the improved decision making, should more than offset the initial costs over the long haul, and certainly the benefits of the ongoing program outweigh the small, administrative costs associated with maintaining the data classification program.

Although not the only methodology that could be employed to develop and implement a data classification program, the one described here has been used and proved to work.

EXHIBIT 60.1 Threat/Risk Analysis

Application	Platform	Threat	Risk	Consequences of Loss
Application				

The following topics will be addressed:

- Getting started: questions to ask
- Policy
- Business Impact Analysis
- Establishing classifications
- Defining roles and responsibilities
- Identifying owners
- Classifying information and applications
- Ongoing monitoring

Getting Started: Questions to Ask

Before the actual implementation of the data classification program can begin, the Information Security Officer (ISO) — whom for the purposes of this discussion is the assumed project manager — must ask some very important questions, and get the answers.

Is there an executive sponsor for this project?

Although not absolutely essential, obtaining an executive sponsor and champion for the project could be a critical success factor. Executive backing by someone well respected in the organization who can articulate the ISO's position to other executives and department heads will help remove barriers, and obtain much needed funding and buy-in from others across the corporation. Without an executive sponsor, the ISO will have a difficult time gaining access to executives or other influencers who can help sell the concept of data ownership and classification.

What are you trying to protect, and from what?

The ISO should develop a threat and risk analysis matrix to determine what the threats are to corporate information, the relative risks associated with those threats, and what data or information are subject to those threats. This matrix provides input to the business impact analysis, and forms the beginning of the plans for determining the actual classifications of data, as will be discussed later in this chapter. (See Exhibit 60.1 for an example Threat/Risk Analysis table).

Are there any regulatory requirements to consider?

Regulatory requirements will have an impact on any data classification scheme, if not on the classifications themselves, at least on the controls used to protect or provide access to regulated information. The ISO should be familiar with these laws and regulations, and use them as input to the business case justification for data classification, as well as input to the business impact analysis and other planning processes.

Has the business accepted ownership responsibilities for the data?

The business, not IT, owns the data. Decisions regarding who has what access, what classification the data should be assigned, etc. are decisions that rest solely with the business data owner. IT provides the technology and processes to implement the decisions of the data owners, but should not be involved in the decision-making process. The executive sponsor can be a tremendous help in selling this concept to the organization. Too many organizations still rely on IT for these types of decisions. The business manager must realize that the data is his data, not IT's; IT is merely the custodian of the data. Decisions regarding access, classification, ownership, etc. resides in the business units. This concept must be sold first, if data classification is to be successful.

Are adequate resources available to do the initial project?

Establishing the data classification processes and procedures, performing the business impact analysis, conducting training, etc. requires an up-front commitment of a team of people from across the organization if the project is to be successful. The ISO cannot and should not do it alone. Again, the executive sponsor can

be of tremendous value in obtaining resources such as people and funding for this project that the ISO could not do. Establishing the processes, procedures, and tools to implement good, well-defined data classification processes takes time and dedicated people.

Policy

A useful tool in establishing a data classification scheme is to have a corporate policy implemented stating that the data are an asset of the corporation and must be protected. Within that same document, the policy should state that information will be classified based on data value, sensitivity, risk of loss or compromise, and legal and retention requirements. This provides the ISO the necessary authority to start the project, seek executive sponsorship, and obtain funding and other support for the effort.

If there is an Information Security Policy, these statements should be added if they are not already there. If no Information Security Policy exists, then the ISO should put the data classification project on hold, and develop an Information Security Policy for the organization. Without this policy, the ISO has no real authority or reason to pursue data classification. Information must first be recognized and treated as an asset of the company before efforts can be expended to protect it.

Assuming there is an Information Security Policy that mentions or states that data will be classified according to certain criteria, another policy — Data Management Policy — should be developed which establishes data classification as a process to protect information and provides:

- The definitions for each of the classifications
- The security criteria for each classification for both data and software
- The roles and responsibilities of each group of individuals charged with implementing the policy or using the data

Below is a sample Information Security Policy. Note that the policy is written at a very high level and is intended to describe the "what's" of information security. Processes, procedures, standards, and guidelines are the "hows" or implementation of the policy.

Sample Information Security Policy

> All information, regardless of the form or format, which is created or used in support of company business activities is corporate information. Corporate information is a company asset and must be protected from its creation, through its useful life, and authorized disposal. It should be maintained in a secure, accurate, and reliable manner and be readily available for authorized use. Information will be classified based on its sensitivity, legal, and retention requirements, and type of access required by employees and other authorized personnel.
>
> Information security is the protection of data against accidental or malicious disclosure, modification, or destruction. Information will be protected based on its value, confidentiality, and/or sensitivity to the company, and the risk of loss or compromise. At a minimum, information will be update-protected so that only authorized individuals can modify or erase the information.

The above policy is the minimum requirement to proceed with developing and implementing a data classification program. Additional policies may be required, such as an Information Management Policy, which supports the Information Security Policy. The ISO should consider developing this policy, and integrating it with the Information Security Policy. This policy would:

- Define information as an asset of the business unit
- Declare local business managers as the owners of information
- Establish Information Systems as the custodians of corporate information
- Clearly define roles and responsibilities of those involved in the ownership and classification of information
- Define the classifications and criteria that must be met for each
- Determine the minimum range of controls to be established for each classification

By defining these elements in a separate Information Management Policy, the groundwork is established for defining a corporate information architecture, the purpose of which is to build a framework for integrating

all the strategic information in the company. This architecture can be used later in the enablement of larger, more strategic corporate applications.

The supporting processes, procedures, and standards required to implement the Information Security and Information Management policies must be defined at an operational level and be as seamless as possible. These are the "mechanical" portions of the policies, and represent the day-to-day activities that must take place to implement the policies. These include but are not limited to:

- The process to conduct a Business Impact Analysis
- Procedures to classify the information, both initially after the BIA has been completed, and to change the classification later, based on business need
- The process to communicate the classification to IS in a timely manner so the controls can be applied to the data and software for that classification
- The process to periodically review:
  - — Current classification to determine if it is still valid
  - — Current access rights of individuals and/or groups who have access to a particular resource.
  - — Controls in effect for a classification to determine their effectiveness
  - — Training requirements for new data owners
- The procedures to notify custodians of any change in classification or access privileges of individuals or groups

The appropriate policies are required as a first step in the development of a Data Classification program. The policies provide the ISO with the necessary authority and mandate to develop and implement the program. Without it, the ISO will have an extremely difficult time obtaining the funding and necessary support to move forward. In addition to the policies, the ISO should solicit the assistance and support of both the Legal Department and Internal Audit. If a particular end-user department has some particularly sensitive data, their support would also provide some credibility to the effort.

Business Impact Analysis

The next step in this process is to conduct a high-level business impact analysis on the major business functions within the company. Eventually this process should be carried out on all business functions, but initially it must be done on the business functions deemed most important to the organization.

A critical success factor in this effort is to obtain corporate sponsorship. An executive who supports the project, and may be willing to be the first whose area is analyzed, could help persuade others to participate, especially if the initial effort is highly successful and there is perceived value in the process.

A Study Team comprised of individuals from Information Security, Information Systems (application development and support), Business Continuity Planning, and Business Unit representatives should be formed to conduct the initial impact analysis. Others that may want to participate could include Internal Audit and Legal.

The Business Impact Analysis process is used by the team to:

- Identify major functional areas of information (i.e., human resources, financial, engineering, research and development, marketing, etc.).
- Analyze the threats associated with each major functional area. This could be as simple as identifying the risks associated with loss of confidentiality, integrity, or availability, or get into more detail with specific threats of computer virus infections, denial of service attacks, etc.
- Determine the risk associated with the threat (i.e., the threat could be disclosure of sensitive information, but the risk could be low because of the number of people who have access, and the controls that are imposed on the data).
- Determine the effect of loss of the information asset on the business (this could be financial, regulatory impacts, safety, etc.) for specific periods of unavailability — one hour, one day, two days, one week, a month.
- Build a table detailing the impact of loss of the information (as shown in Exhibit 60.2 — Business Impact Analysis)

EXHIBIT 60.2 Business Impact Analysis

Function	Application	Type Loss (CIA)	Cost after 1 Hour	Cost after 2 Hours	Cost after 1 Day	Cost after 1 Week	Cost after 1 Month
Human Resources	Payroll	Confidentiality					
		Integrity					
		Availability					
	Medical	Confidentiality					
		Integrity					
		Availability					

- Prepare a list of applications that directly support the business function (i.e., Human Resources could have personnel, medical, payroll files, skills inventory, employee stock purchase programs, etc.) This should be part of Exhibit 60.2.

From the information gathered, the team can determine universal threats that cut across all business functional boundaries. This exercise can help place the applications in specific categories or classifications with a common set of controls to mitigate the common risks. In addition to the threats and their associated risks, sensitivity of the information, ease of recovery, and criticality must be considered when determining the classification of the information.

Establish Classifications

Once all the risk assessment and classification criteria have been gathered and analyzed, the team must determine how many classifications are necessary and create the classification definitions, determine the controls necessary for each classification for the information and software, and begin to develop the roles and responsibilities for those who will be involved in the process. Relevant factors, including regulatory requirements, must be considered when establishing the classifications.

Too many classifications will be impractical to implement; most certainly will be confusing to the data owners and meet with resistance. The team must resist the urge for special cases to have their own data classifications. The danger is that too much granularity will cause the process to collapse under its own weight. It will be difficult to administer and costly to maintain.

On the other hand, too few classes could be perceived as not worth the administrative trouble to develop, implement, and maintain. A perception may be created that there is no value in the process, and indeed the critics may be right.

Each classification must have easily identifiable characteristics. There should be little or no overlap between the classes. The classifications should address how information and software are handled from their creation, through authorized disposal. See Exhibit 60.3, Information/Software Classification Criteria.

Following is a sample of classification definitions that have been used in many organizations:

- **Public** — Information, that if disclosed outside the company, would not harm the organization, its employees, customers, or business partners.
- **Internal Use Only** — Information that is not sensitive to disclosure within the organization, but could harm the company if disclosed externally.
- **Company Confidential** — Sensitive information that requires "need to know" before access is given.

It is important to note that controls must be designed and implemented for both the information and software. It is not sufficient to classify and control the information alone. The software, and possibly the

EXHIBIT 60.3 Information/Software Classification Criteria

Classification	Storage Media	Minimum Data Controls	Minimum Software Controls	Transmission Considerations	Destruction Mechanisms
Application					

hardware on which the information and/or software resides, must also have proportionate controls for each classification the software manipulates. Below is a set of minimum controls for both information and software that should be considered.

Information — Minimum Controls

- **Encryption** — Data is encrypted with an encryption key so that the data is "scrambled." When the data is processed or viewed, it must be decrypted with the same key used to encrypt it. The encryption key must be kept secure and known only to those who are authorized to have access to the data. Public/private key algorithms could be considered for maximum security and ease of use.
- **Review and approve** — This is a procedural control, the intent of which is to ensure that any change to the data is reviewed by someone technically knowledgeable to perform the task. The review and approval should be done by an authorized individual other than the person who developed the change.
- **Backup and recovery** — Depending on the criticality of the data and ease of recovery, plans should be developed and periodically tested to ensure the data is backed up properly, and can be fully recovered.
- **Separation of duties** — The intent of this control is to help ensure that no single person has total control over the data entry and validation process, which would enable someone to enter or conceal an error that is intended to defraud the organization or commit other harmful acts. An example would be not allowing the same individual to establish vendors to an Authorized Vendor File, then also be capable of authorizing payments to a vendor.
- **Universal access: none** — No one has access to the data unless given specific authority to read, update, etc. This type of control is generally provided by security access control software.
- **Universal access: read** — Everyone with access to the system can read data with the control applied; however, update authority must be granted to specific individuals, programs, or transactions. This type of control is provided by access control software.
- **Universal access: update** — Anyone with access to the system can update the data, but specific authority must be granted to delete the data. This control is provided by access control software.
- **Universal access: alter** — Anyone with access to the system can view, update, or delete the data. This is virtually no security.
- **Security access control software** — This software allows the administrator to establish security rules as to who has access rights to protected resources. Resources can include data, programs, transactions, individual computer IDs, and terminal IDs. Access control software can be set up to allow access by classes of users to classes of resources, or at any level of granularity required to any particular resource or group of resources.

Software — Minimum Controls

- **Review and approve** — The intent of this control is that any change to the software be reviewed by someone technically knowledgeable to perform this task. The review and approval should be an authorized individual other than the person who developed the change.
- **Review and Approve Test Plan and Results**
  A test plan would be prepared, approved, documented, and followed.
- **Backup and recovery** — Procedures should be developed and periodically tested to ensure backups of the software are performed in such a manner that the most recent production version is recoverable within a reasonable amount of time.
- **Audit/history** — Information documenting the software change such as the work request detailing the work to be performed, test plans, test results, corrective actions, approvals, who performed the work, and other pertinent documentation required by the business.
- **Version and configuration control** — Refers to maintaining control over the versions of software checked out for update, being loaded to staging or production libraries, etc. This would include the monitoring of error reports associated with this activity and taking appropriate corrective action.
- **Periodic testing** — Involves taking a test case and periodically running the system with known data that has predictable results. The intent is to ensure the system still performs as expected, and does not

produce results that are inconsistent with the test case data. These tests could be conducted at random or on a regular schedule.

- **Random checking** — Production checking of defined data and results.
- **Separation of duties** — This procedural control is intended to meet certain regulatory and audit system requirements by helping ensure that one single individual does not have total control over a programming process without appropriate review points or requiring other individuals to perform certain tasks within the process prior to final user acceptance. For example, someone other than the original developer would be responsible for loading the program to the production environment from a staging library.
- **Access control of software** — In some applications, the coding techniques and other information contained within the program are sensitive to disclosure, or unauthorized access could have economic impact. Therefore, the source code must be protected from unauthorized access.
- **Virus checking** — All software destined for a PC platform, regardless of source, should be scanned by an authorized virus-scanning program for computer viruses before it is loaded into production on the PC or placed on a file server for distribution. Some applications would have periodic testing as part of a software quality assurance plan.

Defining Roles and Responsibilities

To have an effective Information Classification program, roles and responsibilities of all participants must be clearly defined. An appropriate training program, developed and implemented, is an essential part of the program. The Study Team identified to conduct the Business Impact Analysis is a good starting point to develop these roles and responsibilities and identify training requirements. However, it should be noted that some members of the original team, such as Legal, Internal Audit, or Business Continuity Planning, most likely will not be interested in this phase. They should be replaced with representatives from the corporate organizational effectiveness group, training, and possibly corporate communications.

Not all of the roles defined in the sections that follow are applicable for all information classification schemes and many of the roles can be performed by the same individual. The key to this exercise is to identify which of the *roles* defined is appropriate for your particular organization, again keeping in mind that an individual may perform more than one of these when the process is fully functional.

- **Information owner** — Business executive or business manager who is responsible for a company business information asset. Responsibilities include, but are not limited to:
  - — Assign initial information classification and periodically review the classification to ensure it still meets the business needs.
  - — Ensure security controls are in place commensurate with the classification.
  - — Review and ensure currency of the access rights associated with information assets they own.
  - — Determine security requirements, access criteria, and backup requirements for the information assets they own.
  - — Perform or delegate, if desired, the following:
    - Approval authority for access requests from other business units or assign a delegate in the same business unit as the executive or manager owner
    - Backup and recovery duties or assign to the custodian
    - Approval of the disclosure of information act on notifications received concerning security violations against their information assets
- **Information custodian** — The information custodian, usually an information systems person, is the delegate of the information owner with primary responsibilities for dealing with backup and recovery of the business information. Responsibilities include the following:
  - — Perform backups according to the backup requirements established by the information owner.
  - — When necessary, restore lost or corrupted information from backup media to return the application to production status.
  - — Perform related tape and DASD management functions as required to ensure availability of the information to the business.
  - — Ensure record retention requirements are met based on the information owner's analysis.

- **Application owner** — Manager of the business unit who is fully accountable for the performance of the business function served by the application. Responsibilities include the following:
  - — Establish user access criteria and availability requirements for their applications.
  - — Ensure the security controls associated with the application are commensurate with support for the highest level of information classification used by the application.
  - — Perform or delegate the following:
    - Day-to-day security administration
    - Approval of exception access requests
    - Appropriate actions on security violations when notified by security administration
    - The review and approval of all changes to the application prior to being placed into the production environment
    - Verification of the currency of user access rights to the application
- **User manager** — The immediate manager or supervisor of an employee. They have ultimate responsibility for all user IDs and information assets owned by company employees. In the case of nonemployee individuals such as contractors, consultants, etc., this manager is responsible for the activity and for the company assets used by these individuals. This is usually the manager responsible for hiring the outside party. Responsibilities include the following:
  - — Inform security administration of the termination of any employee so that the user ID owned by that individual can be revoked, suspended, or made inaccessible in a timely manner.
  - — Inform security administration of the transfer of any employee if the transfer involves the change of access rights or privileges.
  - — Report any security incident or suspected incident to Information Security.
  - — Ensure the currency of user ID information such as the employee identification number and account information of the user ID owner.
  - — Receive and distribute initial passwords for newly created user IDs based on the manager's discretionary approval of the user having the user ID.
  - — Educate employees with regard to security policies, procedures, and standards to which they are accountable.
- **Security administrator** — Any company employee who owns a user ID that has been assigned attributes or privileges associated with access control systems, such as ACF2, Top Secret, or RACF. This user ID allows them to set system-wide security controls or administer user IDs and information resource access rights. These security administrators may report to either a business division or Information Security within Information Systems. Responsibilities include the following:
  - — Understand the different data environments and the impact of granting access to them.
  - — Ensure access requests are consistent with the information directions and security guidelines.
  - — Administer access rights according to criteria established by the Information Owners.
  - — Creat and remove user IDs as directed by the user manager.
  - — Administer the system within the scope of their job description and functional responsibilities.
  - — Distribute and follow up on security violation reports.
  - — Send passwords of newly created user IDs to the manager of the user ID owner only.
- **Security analyst** — Person responsible for determining the data security directions (strategies, procedures, guidelines) to ensure information is controlled and secured based on its value, risk of loss or compromise, and ease of recoverability. Duties include the following:
  - — Provide data security guidelines to the information management process.
  - — Develop basic understanding of the information to ensure proper controls are implemented.
  - — Provide data security design input, consulting and review.
- **Change control analyst** — Person responsible for analyzing requested changes to the IT infrastructure and determining the impact on applications. This function also analyzes the impact to the databases, data-related tools, application code, etc.

- **Data analyst** — This person analyzes the business requirements to design the data structures and recommends data definition standards and physical platforms, and is responsible for applying certain data management standards. Responsibilities include the following:
  - — Design data structures to meet business needs.
  - — Design physical data base structure.
  - — Create and maintain logical data models based on business requirements.
  - — Provide technical assistance to data owner in developing data architectures.
  - — Record metadata in the data library.
  - — Creat, maintain, and use metadata to effectively manage database deployment.
- **Solution provider** — Person who participates in the solution (application) development and delivery processes in deploying business solutions; also referred to as an integrator, application provider/programmer, IT provider. Duties include the following:
  - — Work with the data analyst to ensure the application and data will work together to meet the business requirements.
  - — Give technical requirements to the data analyst to ensure performance and reporting requirements are met.
- **End user** — Any employees, contractors, or vendors of the company who use information systems resources as part of their job. Responsibilities include:
  - — Maintain confidentiality of log-on password(s).
  - — Ensure security of information entrusted to their care.
  - — Use company business assets and information resources for management approved purposes only.
  - — Adhere to all information security policies, procedures, standards, and guidelines.
  - — Promptly report security incidents to management.
- **Process owner** — This person is responsible for the management, implementation, and continuous improvement of a process that has been defined to meet a business need. This person:
  - — Ensures data requirements are defined to support the business process.
  - — Understands how the quality and availability affect the overall effectiveness of the process.
  - — Works with the data owners to define and champion the data quality program for data within the process.
  - — Resolves data-related issues that span applications within the business processes.
- **Product line manager** — Person responsible for understanding business requirements and translating them into product requirements, working with the vendor/user area to ensure the product meets requirements, monitoring new releases, and working with the stakeholders when movement to a new release is required. This person:
  - — Ensures new releases of software are evaluated and upgrades are planned for and properly implemented.
  - — Ensures compliance with software license agreements.
  - — Monitors performance of production against business expectations.
  - — Analyzes product usage, trends, options, competitive sourcing, etc. to identify actions needed to meet project demands of the product.

Identifying Owners

The steps previously defined are required to establish the information classification infrastructure. With the classifications and their definitions defined, and roles and responsibilities of the participants articulated, it is time to execute the plan and begin the process of identifying the information owners. As stated previously, the information owners *must* be from the business units. It is the business unit that will be most greatly affected if the information becomes lost or corrupted; the data exists solely to satisfy a business requirement. The following criteria must be considered when identifying the proper owner for business data:

- Must be from the business; data ownership is *not* an IT responsibility.
- Senior management support is a key success factor.
- Data owners must be given (through policy, perhaps) the necessary authority commensurate with their responsibilities and accountabilities.
- For some business functions, a multi-level approach may be necessary.

A phased approach will most likely meet with less resistance than trying to identify all owners and classify all information at the same time. The Study Team formed to develop the roles and responsibilities should also develop the initial implementation plan. This plan should consider using a phased approach — first identifying from the risk assessment data those applications that are critical or most important by orders of magnitude to the corporation (such as time-critical business functions first, etc.). Owners for these applications are more easily identified and probably are sensitized to the mission criticality of their information. Other owners and information can be identified later by business functions throughout the organization.

A training program must also be developed and be ready to implement as the information owners and their delegates are named. Any tools such as spreadsheets for recording application and information ownership and classification and reporting mechanisms should be developed ahead of time for use by the information owners. Once the owners have been identified, training should commence immediately so that it is delivered at the time it is needed.

Classify Information and Applications

The information owners, after completing their training, should begin collecting the meta data about their business functions and applications. A formal data collection process should be used to ensure a consistency in the methods and types of information gathered. This information should be stored in a central repository for future reference and analysis. Once the information has been collected, the information owners should review the definitions for the information classifications, and classify their data according to that criteria. The owners can use the following information in determining the appropriate controls for the classification:

- Audit information maintained: how much and where it is, and what controls are imposed on the audit data
- Separation of duties required: yes or no; if yes, how is it performed
- Encryption requirements
- Data protection mechanisms; access controls defined based on classification, sensitivity, etc.
- Universal access control assigned
- Backup and recovery processes documented
- Change control and review processes documented
- Confidence level in data accuracy
- Data retention requirements defined
- Location of documentation

The following application controls are required to complement the data controls, but care should be taken to ensure all controls (both data and software) are commensurate with the information classification and value of the information:

- Audit controls in place
- Develop and approve test plans
- Separation of duties practiced
- Change management processes in place
- Code tested, verified for accuracy
- Access control for code in place
- Version controls for code implemented
- Backup and recovery processes in place

Ongoing Monitoring

Once the information processes have been implemented and data classified, the ongoing monitoring processes should be implemented. The internal audit department should lead this effort to ensure compliance with policy and established procedures. Information Security, working with selected information owners, Legal, and other interested parties, should periodically review the information classifications themselves to ensure they still meet business requirements.

The information owners should periodically review the data to ensure that it is still appropriately classified. Also, access rights of individuals should be periodically reviewed to ensure these rights are still appropriate for the job requirements. The controls associated with each classification should also be reviewed to ensure they are still appropriate for the classification they define.

Summary

Information and software classification is necessary to better manage information. If implemented correctly, classification can reduce the cost of protecting information because in today's environment, "one size fits all" will no longer work within the complexity of most corporation's heterogeneous platforms that make up the IT infrastructure. Information classification enhances the probability that controls will be placed on the data where they are needed the most, and not applied where they are not needed.

Classification security schemes enhance the usability of data by ensuring the confidentiality, integrity, and availability of information. By implementing a corporate-wide information classification program, good business practices are enhanced by providing a secure, cost-effective data platform that supports the company's business objectives. The key to the successful implementation of the information classification process is senior management support. The corporate information security policy should lay the groundwork for the classification process, and be the first step in obtaining management support and buy-in.

61

A Matter of Trust

Ray Kaplan, CISSP, CISA, CISM

There is a continuous stream of security-related bug reports permeating the news these days. With all the noise, it is difficult to spot the core issue, let alone to keep one's eye on it. The simple questions of what one trusts and why one trusts it are often ignored. Moreover, the need to define both inter- and intra-infrastructure trust relationships is often overlooked. The core question of what trust is and its importance is usually forgotten altogether. Security is a matter of trust. This chapter explores the nature of trust and trust relationships, and discusses how one can use trust to build a secure infrastructure.

A Matter of Trust?

Trust is the core issue in security. Unfortunately, simply understanding that is not going to get one very far when one has an infrastructure to secure. The people in an organization, its customers, and their customers are depending on the security of one's infrastructure. Strangely enough (and do not take this personally), they probably should not. The reality is that people make poor decisions about trust all the time, and often engage in relationships based on flawed trust decisions.

Before exploring this further, it is important to understand what trust is and how it is used to build and maintain a trustworthy infrastructure. One can start with the definition of trust — what it is and what it is not. Then this chapter explores how to build and maintain a trustworthy infrastructure.

Trust Defined

The dictionary variously defines trust as a *firm belief or confidence in the honesty, integrity, reliability, justice, etc. of another person or thing*. It goes on to talk about confident expectation, anticipation or hope, imparting responsibility, and engendering confidence. This allows for the development of relationships. Consider committing something or someone to someone else's care, putting someone in charge of something, allowing something to take place without fear, and granting someone credit. All these things are examples of how most people operate — as individuals, as citizens, as organizations, and as a society (locally, nationally, and internationally).

In matters of real-world models of trust for the Internet, law, E-commerce, linguistics, etc., one base definition applies:

> Trust is that which is essential to a communication channel but cannot be transferred from source to destination using that channel.[1]

One can look to information theory as an anchor:

> In Information Theory, information has nothing to do with knowledge or meaning. In the context of Information Theory, information is simply that which is transferred from a source to a destination, using a communication channel.[2]

Think of trust as a value attached to information.

Examples of where people rely on trust in matters of security are everywhere in computing and networking. For example, the scheduler of an operating system trusts the mechanism that is giving it entities to schedule

for execution. A TCP/IP network stack trusts that the source address of a packet can be trusted to be its originator (unless a security mechanism demands proof of the source's identity). Most users trust their browsers and the Web sites they access to automatically "do the right thing" security-wise. In doing so, they trust the operating system schedulers and network stacks on which they rely. The NSA sums it up best when it says that a trusted system or component is one with the power to break one's security policy. However, most organizations do not consider trust in this context.

It is extraordinarily important to understand how this puzzle fits together because everything concerning the security of the distributed systems being developed, deployed, and used depends on it. Consider PKIs and operating systems with distributed trust models such as Windows NT and Windows 2000 as examples.

What Trust Is Not

It is also important to talk about what trust is not. In his works on trust, Dr. E. Gerck explains that trust is not transitive, distributive, associative, or symmetric except in certain instances that are very narrowly defined.[3] Gerck uses simple examples, mathematical proofs, and real-world experience to illustrate trust. Because practical experience in security agrees with him, it is comforting to know that Gerck begins his work with a quote from the Polish mathematician Stanislaw Leshniewski:

> A theory, ultimately, must be judged for its accord with reality.[4]

Because rules regarding trust are regularly ignored, people are going to continue to have heartburn as they deal with trust between UNIX systems, build out Public Key Infrastructures (PKIs) and distributed infrastructures, and deal with the practical aspects of Microsoft Windows 2000's new security model. Note that these are just a few of the problem areas.

Before beginning, a note is in order; this is NOT an exercise in demeaning Windows 2000. However, Windows 2000 provides excellent examples of:

- How trust rules are broken with alacrity
- How detailed things can get when one sets about the task of evaluating a trust model
- The problems presented by trust models that break the rules

Take one of Gerck's assertions at a time, using simple examples based on research, mathematical proofs, and real-world experience — starting with an introduction to the problem with the basic Windows 2000 trust model: transitivity.

Trust Is Not Transitive

If X trusts Y and Y trusts Z, X cannot automatically trust Z. That is, the simple fact that I trust you is not reason for me to trust everyone who you trust. This is a major limiting factor in "web-of-trust" models such as that of PGP. This is quite understandable because PGP was developed as e-mail security software for a close group of friends or associates who would handle trust management issues.[5] Within a "closed group," trust is transitive only to the extent that each group member allows it to be. Outside a "closed group," there is no trust. A problem arises when the group is large and its members do not restrict the trust they place in the credentials presented to them by a "trusted" group member. Consequently, a problem results when systems rely on "relative" references. Windows 2000 is such a system because it has a model based on transitive trust. Simply the way that transitive trust is expected to be used in a Windows 2000 system is problematic, as illustrated by the following descriptions of how it works.

First, excerpts from the Windows NT Server Standards documentation that discuss Primary and Trusted Domains point out the differences between Windows NT 4.0 and Windows 2000:

> ...A trusted domain is one that the local system trusts to authenticate users. In other words, if a user or application is authenticated by a trusted domain, its authentication is accepted by all domains that trust the authenticating domain.

On a Windows NT 4.0 system, trust relationships are one-way and must be created explicitly. Two-way trust is established explicitly by creating two one-way trusts. This type of trust is nontransitive, meaning that if one trusts a domain, one does not automatically trust the domains that domain trusts.

> On a Windows NT 4.0 workstation, a Trusted Domain object is used to identify information for a primary domain rather than for trusted domains...

> ...on a Windows 2000 system, each child domain automatically has a two-way trust relationship with the parent. By default, this trust is transitive, meaning that if you trust a domain, you also trust all domains that domain trusts.[6]

Second, an excerpt from a Microsoft NT Server Standards document:

> Windows 2000 Domains can be linked together into an ADS "tree" with automatic two-way transitive trust relationships. ADS (Active Directory Server) "trees" can also be linked together at their roots into an "enterprise" or "forest," with a common directory schema and global catalog server by setting up static "site link bridges," which are like manual trust relationships.[7]

Finally, an excerpt from the Microsoft 2000 Advanced Server Documentation:

> Because all Windows 2000 domains in a forest are linked by transitive trust, it is not possible to create one-way trusts between Windows 2000 domains in the same forest.
>
> …All domain trusts in a Windows 2000 forest are two-way transitive trusts.[8]

The Microsoft 2000 Advanced Server Documentation explains that all the domains in the forest trust the forest's root domain, and all the interdomain trust paths are transitive by definition. Note that all of this stands in stark contrast to what we know: trust is not transitive except in certain, narrowly defined cases. Suffice it to say, the implications of this dependence on transitive trust and the accompanying default behavior present significant challenges. Consider a classic example: the Human Resources (HR) department's domain. Due to the sensitive nature of HR information, it is not clear that an automatic, blanket, transitive interdomain trust relationship with every other domain in the infrastructure is appropriate. For example, HR may be segregated into its own domain to prevent non-HR network administrators from other domains from accessing its resources and protected objects (such as files.)

Other examples of inappropriate transitive trust abound. Examples of why it is a problem, how it must be handled, and the problems associated with it in the UNIX environment can be found in Marcus Ranum's explanation of transitive trust in the UNIX NFS (Network File System) and rlogin (remote login) facilities.[9]

Trust Is Not Distributive

If W and Y both trust Z, W cannot automatically trust Y and Z as a group.

Suppose your organization and your biggest competitor both get certificates from a Certification Authority (CA). Sometime later, that competitor buys that CA, thereby gaining access rights to all of your information. One cannot automatically trust that your biggest competitor would not revoke your certificate and access all of your information.[10] Practically speaking, such a situation might be met with a lawsuit (if your contract with the CA has been breached, for example). However, this is likely to be difficult because agreements with CAs may not provide for this eventuality.

One could also merely change one's behavior by:

- No longer trusting the offending CA or the credentials that it issued to you
- Ensuring that these, now untrustworthy credentials are revoked
- Getting new credentials from a CA with which one has a viable trust relationship

Trust Is Not Associative

If X trusts the partnership formed with Y and Z for some specific purpose, X cannot automatically trust Y and Z individually.

Just because one trusts a group (that presumably was formed for some specific purpose) does not mean that one can trust each member of that group. Suppose one trusts the partnership formed between two competitors for some specific purpose. That, in and of itself, does not mean that one can trust them individually, even in a matter that has do with the business of the partnership.

Trust Is Not Symmetric

Just because X trusts Y, Y cannot automatically trust X. That is, trust relationships are not automatically bidirectional or two-way. Trust is unidirectional or asymmetric. Just because I trust you, you cannot automatically trust me.

As illustrated several times in the preceding discussion, the trusting party decides the acceptable limits of the trust. The only time trust is transitive, distributive, associative, or symmetric is when some type of "soft

trust" exists — specifically where the trusting party permits it.[11] Practically speaking, many trusting parties do not limit the scope of the trust they place in others. Accordingly, those trust relationships are ill-founded. This is a problem — not a benefit.

Trustworthiness

Whereas trust means placing one's confidence in something, trustworthiness means that one's confidence is well-founded. Trusting something does not make it trustworthy. This is the pay dirt of the trust business.

While many systems and networks can be trusted, few are trustworthy. A simple example will help tease this out. Suppose you live far from your job in a metropolitan area that has little or no mass transit. Chances are that you will commute to work by automobile. You may trust your automobile to get you to and from work just fine without ever experiencing a hitch. However, you may not trust it to get you across Death Valley in the middle of a hot summer day. The reason might be that help is only a cell phone call away in the metropolitan area and you know that a breakdown will not be life-threatening. On the other hand, help might be very difficult to find on the journey across Death Valley, and if you break down, dehydration is a threat. You have decided that your car is trustworthy for commuting to work, whereas it is not trustworthy for long journeys through hostile environments. That is, for the purposes of commuting within your own metropolitan area, you trust your automobile for transportation.

Simply put, trust is situational. That is, one decides to trust something in certain, specific circumstances. Trust is about confidence.

One can consider systems and networks to be trustworthy when they have been shown to perform their jobs correctly in a security sense. That is, one has confidence in them under specific circumstances. Accordingly, this encompasses a wide spectrum of trustworthiness. On one end of this spectrum are the so-called trusted systems that require formal assurance of this assertion based on mathematical proofs. On the other end of this spectrum lie bodies of anecdotal evidence gathered over a long period of time that seems to say "the system is doing its job."

As a practical example of how all this fits together, consider one of Dr. Ed Gerek's notes on the definition of trust, which refers to the American Bar Association's Digital Signature Guidelines:[12]

> Trust is not defined per se, but indirectly, by defining "trustworthy systems" (or, systems that deserve trust) as "Computer hardware, software, and procedures that: (1) are reasonably secure from intrusion and misuse; (2) provide a reasonably reliable level of availability, reliability and correct operation; (3) are reasonably suited to performing their intended functions; and (4) adhere to generally accepted security principles." This definition is unfortunate in that it confuses trust with fault-tolerance, especially so because fault-tolerance is objective and can be quantitatively measured by friends and foes alike — whereas trust is the opposite.[13]

As can be seen, one tries to define trust (trustworthiness) as a measurable quantity in many ways. On the technical side of security, there are several ways to accomplish this, including:

- Formal criteria such as the *Trusted Computer Security Evaluation Criteria* (TCSEC, also known as the *Orange Book*) and its successor the Common Criteria and accompanying formal methods of test
- Less formal testing that is performed by the commercial product testing labs such as those that certify firewalls
- So-called "challenge sites" that seek to prove themselves trustworthy by demonstrating that they can withstand attacks
- Penetration testing that seeks to exhaustively test for all known vulnerabilities
- Assessments that seek to show where vulnerabilities exist past those that can be found using purely technical means
- Alpha, beta, and pre-releases of software and hardware that attempt to identify problems before a final version of a product is shipped

All of these are designed to demonstrate that we can trust systems or networks *under certain circumstances*. The object of all of them is to build trust and confidence and thereby to arrive at a level of trust — circumstances under which the systems or networks are trustworthy. For example, among the many things one finds in the so-called *Rainbow Series* of books that contains the *Orange Book* of the TCSEC are *Guidelines for Writing Trusted Facility Manuals*[14] and *A Guide to Understanding Trusted Facility Management*[15] that discuss how a trusted system must be deployed. Quoting the manual:

Guidelines for Writing Trusted Facility Manuals provides a set of good practices related to the documentation of trusted facility management functions of systems.

"Trusted facility management" is defined as the administrative procedures, roles, functions (e.g., commands, programs, interfaces), privileges, and databases used for secure system configuration, administration, and operation.

Before one can trust a system to be secure, the facility in which the system is deployed must be managed in such a way that it can be trusted. Before giving up on this military-style thinking, consider that commercial systems and network components such as routers must be treated in the same way before one can trust them.

Note that these theories and various testing methods are limited; they do not always work in practice. However, one generally uses adherence to criteria and high scores on tests as measures of trustworthiness.

Another way to look at it is that one mitigates as many risks as one can and accepts the remaining risks as residual risk. Nothing is risk-free, including systems and networks. Hence, our job in security is risk management. Eliminate the risks that one can and accept the rest. The reason for this is that it would be much too expensive to eliminate all risks; even if this were possible, one usually cannot identify absolutely all of them.

Why Is Trust Important?

It is easy to see why trust and trustworthiness are important. Start with a global view. The best articulation of this global view that this author has found is in Francis Fukuyama's *Trust, The Social Virtues & The Creation of Prosperity.* One quote seems to apply to everything we do in life and everything we do in computing and networking, including security:

> A nation's well-being, as well as its ability to compete, is conditioned by a single pervasive cultural characteristic: the level of trust inherent in the society.[16]

Consider that the well-being of our enterprises, including their ability to compete, is conditioned on a single pervasive characteristic of their infrastructures and those on which they depend: the level of inherent trust. Simply put, if one cannot trust one's infrastructure, all bets are off. Consider your own desktop system. How comfortable will you be in using it if you cannot trust it?

As a Ph.D. candidate in 1990, Dr. David Cheriton commented:

> The limit to distributed systems is not performance, it is trust.[17]

Cheriton's statement is especially applicable in an age where everything, including toasters, has computing power and the network accoutrements necessary to connect it to everything else in our lives. The interconnectivity aspect of this developing complexity is best illustrated by the following quote from Robert Morris, Sr.:

> To a first approximation, everything is connected to everything else.[18]

This can be a very scary thought. Increasingly, people are trusting more and more of what they are, have, and know, to parties they may not even know, much less have a basis upon which to establish trust. Trust is becoming increasingly important, but most individuals and organizations do not realize or appreciate this until assets are lost or compromised.

Why Do People Trust?

As previously discussed, there are many reasons why people trust. It is important to note that most people never get to the point where they consider any of the reasons in the trustworthiness spectrum. There is only one reason that most of us trust: blind faith. The reasons seem to include:

- Evidence that "things seem to be doing their jobs"
- Lack of evidence to the contrary
- Anecdotal evidence from others in the community

Moreover, the nature of people in many cultures of the world is to trust first and ask questions later — if ever. This is a little confusing because there is often much evidence to the contrary. Nevertheless, it seems to remain a key part of many cultures.

Why Should People Not Trust?

Perhaps the best way to show the importance of trust is to talk about distrust: the lack of trust, faith, or confidence, doubt or suspicion.

The scary part is that most of what people trust is beyond their control. By way of illustration, consider only one dimension of the problem: complexity. In his musings about complexity, Marcus Ranum observes that Web browsers themselves have become tools for managing complexity. Consider that most every browser is in the business of hiding the complexity of having to deal with the myriad of protocols that most of them support (such as HTTP, FTP, Telnet, etc.). Ranam asks how many of us know all of the features and hooks of the cool, new Web apps that continue to pop up. He posits that probably the only people who know are the ones who coded them. Moreover, the details of the security of such protocols are not published and change from version to version.[19]

As an example that gives life to this, consider this discussion of the "Smart Browsing" feature that showed up in version 4.06 of the Netscape browser:[20]

> Netscape Communications Corporation's release of Communicator 4.06 contains a new feature, 'Smart Browsing,' controlled by a new icon labeled What's Related, a front end to a service that will recommend sites that are related to the document the user is currently viewing. The implementation of this feature raises a number of potentially serious privacy concerns, which we have examined here.
>
> Specifically, URLs that are visited while a user browses the Web are reported back to a server at Netscape. The logs of this data, when used in conjunction with cookies, could be used to build extensive dossiers of individual Web users, even including their names, addresses, and telephone numbers in some cases.

If one is having trouble with this, one can easily make a headache worse by trying to get one's arms around all of the trust questions that surround PKIs and Windows 2000 if not already doing so. Consider that the problem of figuring out how to build and maintain a long-lived trust model with Windows 2000 pales when compared to the problem of figuring out how to trust Windows 2000 itself. This, since it reportedly has 27 to 40 million lines of code,[21] some half or more of which are reportedly new to the initial release. The number of security-related bug fixes is likely to be just as astounding.

Complexity and protocol issues notwithstanding, there is no reason for most people and organizations to trust their infrastructures. The reasons to distrust an infrastructure are legion. Very few infrastructures are trustworthy. Given Marcus Ranum's observation about complexity and what the author of this chapter knows about how things work (or do not work), this author has trouble trusting his own infrastructures much of the time.

Finally, there are the continual reminders of purposeful deceit that confront us every day. These begin with virus, worm, and Trojan horse writers foisting their wares on us, continue through the daily litany of security problems that flood past us on lists such as Bugtraq,[22] and end with the malice of forethought of attackers. Clever competitors, in either business or war, will deceive people at every available opportunity. For an instruction manual, I recommend Sun-Tzu's *On the Art of War* for your study.[23] Your adversaries, be they your competitors in business or those who would attack your infrastructure, are likely out to deceive you at every opportunity.

What justifies the trust placed in an infrastructure? Most of the time, there is only one answer: We have never considered that question. However, the absence of justification for trusting infrastructure does not stop there. Some people — and entire organizations — deceive themselves. *The Skeptic's Dictionary*[24] aptly describes this situation: "The only thing infinite is our capacity for self-deception." Better yet: "There is no accounting for self-delusion."[25]

Securing One's Infrastructure

Now one is down to the nub of the matter. There is only one question left to ask: "Where to from here?" Thankfully, the answer is relatively straightforward. Not to say that it will not take some work. However, it is easy and intuitive to see how to attain trust in one's infrastructure:

1. Decide to approach the problem of gaining trust as an exercise in risk management.
2. Develop a plan.
3. Implement the plan.

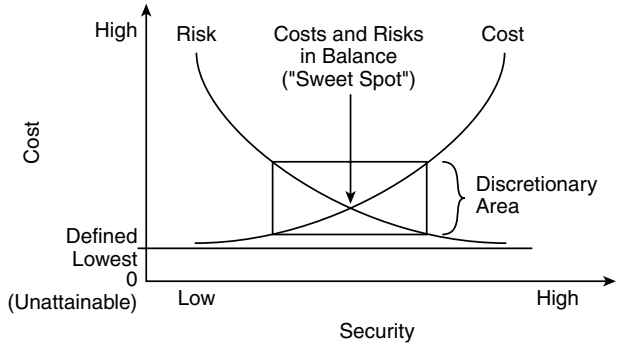


EXHIBIT 61.1 Balancing cost and security.

4. Assess the plan's effectiveness.
5. Modify the plan if necessary.
6. Go to step 1.

This is a sure-fire recipe for success — guaranteed. Barring total disaster outside the control of whomever is following the instructions (e.g., the company going out of business), it has never failed in this author's experience. That is because it is simply a basic problem-solving model. One can fill in the details, starting with risk management.

Risk Management 101

Risk management is an exercise in which one balances a simple equation. Exhibit 61-1 presents a simple picture of how the process of risk management works. A few definitions will help make it readable, starting with security. The *American Heritage Dictionary* offers several definitions, including:

> Freedom from risk or danger; safety.

Accepting this as a starting point presents the first challenge. How does one define risk, danger, and safety for a computing infrastructure? Start with some terminology.

Vulnerabilities, Threats, Risks, and Countermeasures

Sticking with commonly accepted security terminology, one can build a list that is oddly self-referential:

- *Vulnerability*: a weakness that can be exploited. Alternatively, a weakness in system security procedures, design, implementation, internal controls, etc. that could be exploited to violate a security policy.
- *Threat*: anything or anyone that can exploit a vulnerability. Alternatively, any circumstance or event with the potential to cause harm to a system in the form of destruction, disclosure, modification of data, or denial-of-service.
- *Risk*: the likelihood and cost of a particular event occurring. Alternatively, the probability that a particular threat will exploit a particular vulnerability of a system.
- *Countermeasure*: a procedure or mechanism that reduces the probability that a specific vulnerability will be exploited or reduces the damage that can result from a specific vulnerability's exploit. Alternatively, a technique, an action, device, or other measure that reduces an infrastructure's vulnerability. (All of these are risks.)

The game is to balance the expense of incurring risk with the expense of mitigating (not merely mediating) risk by applying just the right amount of countermeasures to offset the vulnerabilities that exist.

Another way to look at this is from the point of view of cost. Exhibit 61.1 illustrates how costs and risk relate to each other. In addition, it shows how one can determine the optimum amount to spend on security. It plots the cost of security against the amount of security one is able to attain for that expenditure.

Perhaps one can now begin to see some of the challenges ahead. All by itself, building a common vocabulary is problematic. One can tackle each of these terms to see how to mold infrastructure security out of them.

These three concepts are logically related and can be grouped together for the sake of discussion. Using them, securing an infrastructure can be as simple as following three simple steps:

1. Identify a vulnerability
2. Identify the threats that can exploit it
3. Design and implement a countermeasure that will reduce the likelihood (risk) that a specific threat can exploit this vulnerability.

Seems simple enough. However, most infrastructures have enough different components in them to make this approach a truly daunting task. Nevertheless, it is an iterative process that uses these steps over and over again. In an ideal situation, every threat, vulnerability, and countermeasure is considered for every infrastructure component in an iterative process. Experience shows that unless one examines every component of an infrastructure in this manner, one simply cannot secure the infrastructure at large.

Practically speaking, however, this is impossible for all but the smallest organizations. Imagine stopping the business of an organization while one goes through this process. After all, the infrastructure is probably in place to support an organization's business needs — not the other way around.

The only exceptions to this rule are where good security is a design goal and there is a resource commitment to go with it. For example, an opportunity certainly exists when brand-new components or entirely new infrastructurea are being installed.

Most people seem to believe that this thinking is restricted to so-called military-grade security. However, most segments of the economy are concerned enough about protecting their information assets to get serious about security. This includes most commercial, industrial, and educational organizations.

One problem is outdated thinking about threats and vulnerabilities. This is being overcome by new thinking that takes a fresh look at them, such as Donn Parker's new framework. It lists several dimensions of threats and vulnerabilities alongside asset categories.[26]

Take a look at how to solve the practical problem of how to complete this risk management exercise without stopping the business of the organization.

Analysis and Quantification

It seems almost obvious to say that the keys to finding and fixing vulnerabilities are analysis and quantification. Only almost, because most organizations do not approach security from a business or technical engineering perspective. Moreover, many organizations run off and buy security technology before they have even identified the problem. Suffice it to say, this sort of thinking is a trap.

To the experienced business or technical problem-solver, there is no other way to proceed. Given a well-stated problem, one simply has to analyze it and quantify the results as the first step.

So, how does one analyze and quantify vulnerabilities, threats, and countermeasures? Were it not for the maturity of the discipline of securing information systems, one would have an impossible task. As it is, this problem is well understood, and there are tools and techniques available. However, it is important to note that no tool is complete. In any case, most successful approaches use another one of the concepts in our basic vocabulary: risk.

Before taking even one step forward from here, a word of caution is in order:

> Quantification of risk is a hard problem.[27] In fact, all attempts to develop reasonable methods in this area have utterly failed over the last several decades. Donn Parker explains exactly how this happened in the 1998 edition of his book *Fighting Computer Crime*.[28] One might succeed in quantifying specific ratings in a narrowly defined area using an arbitrary ranking scale. However, experience shows that reconciling these ratings with others that use equally arbitrary ranking is impossible, especially on the scale of a contemporary, large, highly distributed organization.

One can use quantitative risk assessment methods. However, experience shows that one will end up using some form of qualitative measure in many areas. Consider the evaluation of a security program at large. One will likely want to score it based on an opinion of how well it is able to do its job for its own organization. Clearly, this requires a qualitative rating scale such as one that ranges from "poorly" to "superbly."

Dealing with Risks

Anytime probability or likelihood is mentioned, most people get nervous. After all, there are systems and networks to secure. One does not need to get lost in "the possibilities." However, there is an intuitive appeal to quantifying things — especially when one has to ask for a budget to support one's security-related efforts.

Management and bean counters have little tolerance for pure speculation, and techies and engineers want details. Everyone wants something concrete to work with. Therein lies the rub.

Given ample time and resources, all system and network managers worth their salt can identify security problems. The missing piece is the ability to quantify the effect of identified security problems in terms of their likelihood. This problem is discussed shortly. In the meantime, take a brief look at how risks are analyzed and quantified.

First, one must seek precise problem definition, analysis, and quantification. In addition, experienced problem-solvers will always ask about their goals. A good way to characterize problem solution goals is to rank them according to completeness:

- *Necessary*. These are the essential elements required to solve a problem. Necessary elements can be viewed as fundamental, cardinal, mandatory, or prerequisite.
- *Sufficient*. These are the additional elements that move a problem solution to completeness by making it adequate, ample, satisfactory, and complete.

Experience with security in commercial, industrial, and educational organizations shows that the essence of picking a reasonable security solution is found in the business and technical *artistry* of combining necessity and sufficiency. In this arena, it is not a science. However, high levels of security are only achieved through the rigor of mathematical proof and the application of rigid rules that strictly control their variables. Here, necessity is assumed and sufficiency is a baseline requirement.

The idea of introducing these concepts at this point is to focus on the cost of security. Properly chosen countermeasures mediate risks. However, in the same way that it takes money to make money, it takes money to provide security. Identifying needed countermeasures does no good if those countermeasures cannot be implemented and maintained because they are dismissed as too expensive.

Where the Rubber Meets the Road

There are few — if any — hard numbers surrounding security. This is especially bothersome when a system or network manager tries to explain to management exactly why those extra person-weeks are needed to properly configure or test the security-related aspects of infrastructure components. While a management team can usually quantify what a given information resource is worth to the business, it is nearly impossible for a system or network manager to translate this valuation directly into hard numbers for a security budget. Some relief is found in longtime security pundit Bob Courtney's summary:

You never spend more than something is worth to protect it.

The problem is determining how much something is worth. The answer is achieving an appropriate balance between cost and risk. Exhibit 61.1 presents a time-honored graphic view of how this works in practice.

As one can see, the balance point between the amount of security one has and its cost (the risks or lack of security) is identified as the *Sweet Spot*. Also note that there is a box labeled *Discretionary Area* that includes an area around the *Sweet Spot*. This is the area in which the amount of security and its cost can be in balance. This is based on the fact that perfectly balancing the risk that one incurs with what it costs to maintain that level of security is very difficult, if not impossible. In other words, there is some discretion in the amount of risk one incurs before either the risks or the high costs being incurred would be considered out of hand by some commonly accepted standard.

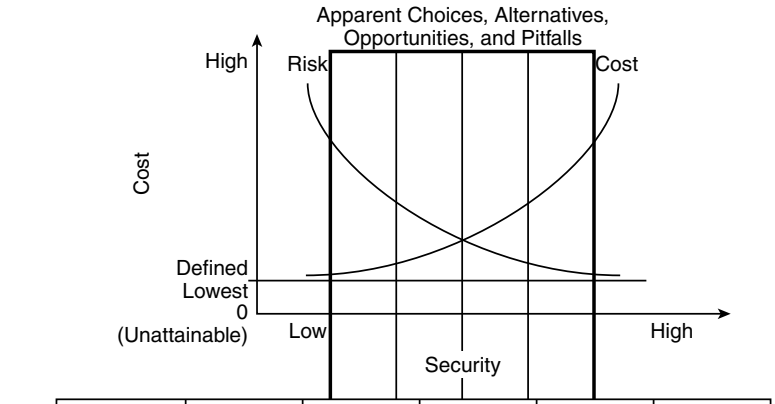
For example, one will never be able to buy a virus scanner that protects against all viruses. Thus, one incurs more risk. On the other hand, one might operate under a conservative policy that severely restricts what can be executed. Thus, one incurs less risk because there are presumably fewer ways for a virus to propagate in one's environment.

Another way to think about this is that the *Discretionary Area* is an area in which both risks and expenditures are "reasonable." Generally speaking, points on the risk curve to the right of the *Sweet Spot* represent overspending (more security than is needed). Points of the risk curve to the left of the *Sweet Spot* represent underspending (less security than is needed).

Limits

A careful inspection of Exhibit 61.1 reveals that neither of the curves ever reach zero and that there are two zeros identified. Two important points about limits explain this:

1. One must define zero points. Infrastructures with absolute zero risk and security with absolute zero cost do not exist and cannot be created.



Sample Risk Evaluation Criteria Categories						
	Expenditure-Based	No Spending	Minimize Costs	Balance Costs and Risks	Invest in the Future	Waste Resources
	Degree of Risk-Based	Maximum Risk	Intolerable Risk	Defined Highest Acceptable Risk	Defined Lowest Acceptable Risk	Zero Risk (Not Attainable)
	Complete-ness-Based	Not Adequate	Not Complete	Necessary	Sufficient	Overkill
	Implementa-tion Style-Based	Haphazard	Expedient	Due Care	Painstaking	Paralized

EXHIBIT 61.2 Risk evaluation criteria.

2. One must define maximums. One can spend as much as one has and still end up with an insecure infrastructure.

Keep it simple. In general, the less one spends, the more risk one incurs. The trick is to identify the level of risk that is acceptable. Again, this is the area Exhibit 61.1 identifies as *Discretionary.*

All of this begs the questions: how does one know what is reasonable?, and how does one determine the risks that exist? Take a look at one time-honored approach.

A Do-It-Yourself Kit

Exhibit 61.2 adds several ways to evaluate risk to the x-axis (security level) of Exhibit 61.1. These risk evaluation criteria are alternative scales on which to measure risk. Each scale includes labels that suggest how the cost of mitigating risk at this level is thought of by commonly accepted standards of due care.

Note that the additional information under the x-axis (security level) is in a box labeled *Apparent Choices, Alternatives, Opportunities, and Pitfalls*. This box encloses the acceptable ranges of risk (listed horizontally on the bottom of the diagram), just as the box labeled *Discretionary Area* did in Exhibit 61.1. These ranges are determined by various risk evaluation criteria (listed next to the right of their respective risk ranges on the bottom of the diagram).

For example, look at *Expenditure-Based* risk evaluation criteria. To judge how much risk is acceptable, one can see that *No Spending* and *Waste Resources* are both outside of the box. N*o Spending* is underkill, and *Waste Resources* is considered overkill — just as one would expect them to be. Using *Implementation Style Based* risk evaluation criteria, one can see that the *Apparent Choices, Alternatives, Opportunities, and Pitfalls* box encloses the range from *Expedient* to *Due Care* to *Painstaking*. Again, this is just as one would expect it to be.

One can add most any criteria one chooses. These are only examples to get started.

A note of caution is in order:

> Attempts to come up with a method to quantify risks have been largely unsuccessful and those that exist are problematic to use, at best. This is not an attempt to provide a quantitative approach to risk analysis past what is necessary for you to see how all of the factors affecting risk interact. In fact, one

can see that the risk evaluation criteria that are listed below the x-axis (level of security) are actually a mix of quantitative and qualitative measures.

Asking what is important and how it will be measured is the best place to start. Once that is done, one can consider the various risk evaluation criteria that are available. While there are many considerations to choose from, those that matter to a particular organization are the most important.

Surrounding an organization's unique concerns, there are standards of due care, common practice, and compliance that can be used as risk evaluation criteria — both from the point of view of industry-specific measures and measures that are common to all organizations. Auditors and other security professionals practiced in doing audits and assessments for a particular industry can provide the industry-specific standards that are important to an organization, as well as what is considered to be due care and common practice in general. For example, some industries such as defense contracting are required to do certain things and there is wide agreement in the security industry about what it takes to protect specific systems such as NT, UNIX, routers, etc. in general.

One will also have to find risk evaluation criteria that match the management style and culture of one's organization. Certainly, one would like to have risk evaluation formulae, models that implement them, and tools that automatically do all this according to Exhibit 61.2 suggestions. However, the state-of-the-art for analysis and quantification in security is quite far from point-and-click tools that do everything. No point-and-click tools do it all. Most of the tools that exist are basically spreadsheets that are elaborately scripted. Unfortunately, these tools help maintain a facade of value for quantitative risk assessment methods.

For now, risk assessment is still very much a job that falls to creative and experienced security professionals to sort out — one little, ugly detail at a time.

The Bottom Lines

Despite the apparent complexity, the process of securing one's infrastructure is quite well-understood and widely practiced. There is hope. The trick is to look at the information system and network infrastructures from a business point of view with a plan. Here are the steps to take and a framework in which one can operate — a time-tested way to approach the problem:

1. Identify the vulnerabilities and threats that the infrastructure faces.
2. Translate these vulnerabilities and threats into statements about the risks that they represent.
3. Organize the risks in a hierarchy that reflects the business needs of the organization.
4. Identify the countermeasures that are required to balance them.
5. Start working on a plan of attack for this list of risks.
6. Start working on reducing the biggest risks — today.

Now that one has a handle on risk, one can proceed to discuss how to gain trust in an infrastructure.

Gaining Trust

The reader will be happy to know that gaining trust in an infrastructure is a well-understood process. It is well-documented and widely practiced. In fact, the literature is ripe with examples. A good reference to the process is found in *Learning from Leading Organizations*.[29] Do not let the fact that it is a government document turn you away. It is an excellent reference, based on processes successfully applied by leading private-sector and government organizations. This author has used a modified version of this model to do security assessments (that is how I know it works so well). This GAO model has been extended to include some of the steps that precede one of the steps in the process. This is represented in Exhibit 61.3 as part of a methodology that works in practice.

In examining Exhibit 61.3, one sees that it includes an iterative loop. The assessment phase of the model has been expanded into a risk assessment and the parts that feed it:

- *Legal, Regulatory, and Business Requirements:* the process of sorting through an organization to identify its constraints (e.g., laws and oversight that determine how it must behave)
- *Identify Assets and Threats:* the process of sorting through an organization's priorities to find what is important and then isolating the threats to those assets
- *Security Advisories and Results of Audits and Monitoring:* the process of identifying the infrastructure's vulnerabilities

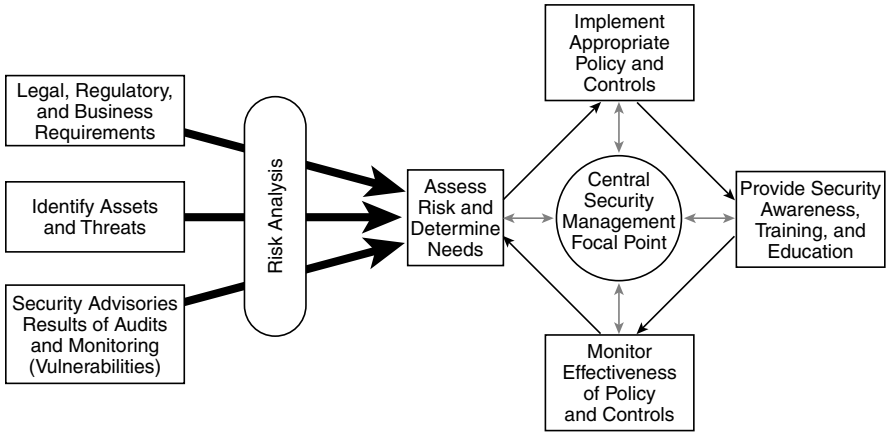


EXHIBIT 61.3 A plan for gaining trust.

The gory details of gaining trust in specific components of the infrastructure are a story for another dissertation. Suffice it to say, following a model such as that presented in the GAO report (as presented in Exhibit 61.3) to identify and mediate risks is a tried and true way to gain trust in one's infrastructure.

Acknowledgments

This chapter was originally an Invited Paper for the *Spring 2000 Internet Security Conference,* http://tisc.core-com.com/. Charlie Payne, Tom Haigh, Steve Kohler, Tom Markham, Rick Smith, Dan Thompson, and George Jelatis of Secure Computing; Randy Keader, Lori Blair, Bryan Koch, and Andrea Nelson of Guardent, Inc.; and Dave Piscitello of Core Competence, Inc., contributed to this paper.

Notes

URLs have been included for as many references as possible. Due to the volatile nature of the Web, these may change from time to time. In the event that a URL does not work, using the title of the reference as the search string for a capable search engine such as http://www.google.com should produce the page or a suitable substitute.

1. Gerck, E., *Towards a Real-World Model of Trust*, http://www.mcg.org.br/trustdef.htm.
2. Gerck, E., *Certification: Intrinsic, Extrinsic and Combined*, MCG, http://www.mcg.org.br/cie.htm.
3. Gerck, E., *Overview of Certification Systems: X.509, CA, PGP and SKIP* http://www.mcg.org.br/cert.htm#CPS; Gerck, E., e-mail message titled: *Towards a Real-World Model of Trust,* http://www.mcg.org.br/trustdef.txt; Gerck, E., e-mail message titled: *Re: Trust Properties*, http://www.mcg.org.br/trustprop.txt.
4. Leshniewski, Stanislaw, (1886–1939) http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Leshniewski.html.
5. Gerck, E., taken together: E-mail message titled: *Towards a Real-World Model of Trust*, http://www.mcg.org.br/trustdef.txt and e-mail message titled: Re: Trust Properties, E. Gerck, http://www.mcg.org.br/trustprop.txt; Gerck, E., *Summary of Current Technical Developments Near-Term Perspectives for Binarily-Secure Communications*, http://www.mcg.org.br/report98.htm.
6. *Primary and Trusted Domains, Local Security Authority Policy Management*, Microsoft MSDN Online Library, http://msdn.microsoft.com/library/default.asp?URL=/library/psdk/lsapol/lsapol_2837.htm.
7. *Microsoft Windows NT Server Standards*, http://www.unc.edu/~jasafir/nt-main.htm.

8. Taken together: Microsoft Windows 2000 Advanced Server Documentation, Understanding Domain Trusts, http://www.windows.com/windows2000/en/advanced/help/sag_AD_UnTrusts.htm — for the table of contents which contains this article see: http://www.windows.com/windows2000/en/advanced/help then choose *Security Overview* then choose *Trust*. Other references one can use to gain an understanding of how the new Windows 2000 trust model works include the following Microsoft online help document heading: *Understanding domain trees and forests*, http://www.windows.com/windows2000/en/server/help/default.asp?url=/windows2000/en/server/help/sag_ADintro_16.htm. In addition, see *Planning Migration from Windows NT to Windows 2000*, http://www.microsoft.com/technet/win2000/win2ksrv/technote/migntw2k.asp.
9. Ranum, Marcus, *Internet Attacks*, http://pubweb.nfr.net/%7Emjr/pubs/attck/index.htm; specifically the section on transitive trust: http://pubweb.nfr.net/%7Emjr/pubs/attck/sld015.htm.
10. Gerck, E., e-mail message titled: *Towards a Real-World Model of Trust*, http://www.mcg.org.br/trustdef.txt.
11. Gerck, E., Summary of Current Technical Developments Near-Term Perspectives for Binarily-Secure Communications, http://www.mcg.org.br/report98.htm.
12. American Bar Association, Legal Infrastructure for Certification Authorities and Secure Electronic Commerce, 1996, http://www.abanet.org/scitech/ec/isc/dsgfree.html.
13. Gerck, E., Towards a Real-World Model of Trust, E. Gerck, http://www.mcg.org.br/trustdef.htm, also Gerck, E., in a 1998 e-mail message defining trust, http://www.sandelman.ottawa.on.ca/spki/html/1998/winter/msg00077.html which references the *American Bar Association Digital Signature Guidelines*, http://www.abanet.org/scitech/ec/isc/dsgfree.html.
14. National Computer Security Center, *Guidelines for Writing Trusted Facility Manuals*, NCSC-TG-016.
15. National Computer Security Center, *A Guide to Understanding Trusted Facility Management*, NCSC-TG-015.
16. Fukuyama, Francis, *Trust, The Social Virtues & the Creation of Prosperity*, ISBN 0-02-910976-0, The Free Press, New York, 1995.
17. From a presentation on security in distributed systems by David Cheriton in a Computer Science Department colloquium at the University of Arizona in the early 1990s.
18. A comment made by NSA computer security researcher Robert Morris, Sr., at a National Computer Security Conference in the early 1990s. He was explaining why he has to work so hard on such things as eliminating covert channels in order to protect the 1000 bit keys that could unleash a nuclear war. (He is the father of Robert T. Morris, who was responsible for the 1988 Internet Worm.)
19. Ranum, Marcus, *The Network Police Blotter*, *Login:* (the newsletter of USENIX and SAGE), February 2000, Volume 25, Number 1, http://pubweb.nfr.net/%7Emjr/usenix/ranum_1.pdf.
20. *What's Related? Everything but Your Privacy*, Matt Curtin, http://www.interhack.net/pubs/whatsrelated/ ; and Curtin, Matt, *What's Related? Fallout*, http://www.interhack.net/pubs/whatsrelated/fallout/.
21. Variously reported to be in that range: The Long and Winding Windows NT Road, Business Week, http://www.businessweek.com/1999/99_08/b3617026.htm, Schwartz, Jeffrey, *Waiting for Windows 2000*, http://www.Internetwk.com/trends/trends041299.htm; Surveyer, Jacques and Serveyer, Nathan, Windows 2000: Same body, two faces, http://www.canadacomputes.com/v3/story/1,1017,1961,00.html; Michetti, Greg B., *Windows 2000 — Another Late System*, http://www.canoe.ca/TechNews9909/13_michetti.html.
22. The bugtraq forum on http://www.securityfocus.com.
23. Tsu, Sun, *On the Art of War, The Oldest Military Treatise in the World*, an easily accessible version, can be found at http://www.chinapage.com/sunzi-e.html.
24. *The Skeptic's Dictionary*, http://skepdic.com/.
25. Connie Brock.
26. Parker, Donn, *Fighting Computer Crime*, John Wiley & Sons, Inc., 1998, Chapter 11, Information Security Assessments, in particular. A summary of risk assessment failure begins on p. 277 of this chapter.
27. There are two styles of risk analysis: quantitative and qualitative. Dictionary definitions imply how they work: quantification — "to determine, express, or measure the quantity of," qualitative — "of, relating to, or involving quality or kind," *WWWebster WWW Dictionary*, http://www.m-w.com/. In his book *Fighting Computer Crime,* Donn Parker presents a complete tutorial on them and why quantitative methods have failed.

28. Parker, Donn, *Fighting Computer Crime*, John Wiley & Sons, Inc., 1998, Chapter 11, Information Security Assessments, in particular. A summary of risk assessment failure begins on p. 277 of this chapter.
29. U.S. General Accounting Office, *Executive Guide, Information Security Management, Learning from Leading Organizations*, GAO/AIMD-98-68, Information Security Management, http://www.gao.gov/special.pubs/pdf_sing.pdf.

62

Trust Governance in a Web Services World

Daniel D. Houser, CISSP, MBA, e-Biz+

The problem space of trust governance is discussed, and five business drivers for trust governance are detailed, including the rise of Web Services, SAML, and Cross-Company Authentication. Xota[SM], a protocol for providing lightweight standards-based trust assertions, is introduced, as well as a framework for utilizing trusted third parties for generating trust assertions. With these in place, enterprise and division security postures can be dynamically evaluated for trustworthiness at the time each transaction or connection is made.

Introduction

Web Services are rapidly changing the face of E-business, while the trust models we use for conducting commerce remain largely unchanged. Cross-company authentication and portals are also increasing interdependence on business partner trustworthiness, while many trust models are built on houses of cards. Many organizations have little means of determining and evaluating the trustworthiness of business partners and their divisions aside from error-prone, expensive, and time-consuming processes. This chapter further outlines the business drivers in this space, and includes analysis of how hacker insurance and changing security attack patterns will likely lead to a regulated industry. With business drivers firmly in place and the problem space defined, a new open protocol for establishing trustworthiness at the transaction and message level is provided (Xota), which permits a dynamic assessment of a business partner's trust status at business application execution time, instead of months earlier. To deliver this protocol, a framework for utilizing trusted third-party assertions is also detailed, along with likely implementation plans for dynamic trust analysis in the B2B and B2C environment.

Prologue: A Story of E-Business and Trust Governance

The thought came to me one day as I reflected on meetings I had attended that morning. In two consecutive meetings, I was asked to provide a time estimate for my team's involvement to secure a data feed to a business partner. This straightforward, everyday IT project meeting had two very different outcomes because of a simple but significant difference. In the first meeting, I asked the business partner what kind of cryptography he provided to protect the data stream. His clear-cut answer described secure, well-proven means for protecting data. We heard names we trusted: PGP, SSL, RSA, VeriSign. I asked a few more questions and determined the protocols and modes he supported, and got solid answers. Finally, he was forthcoming with a vulnerability assessment from a reputable information security consulting firm, which included an assessment of the application. In five questions, I had determined enough to permit a comfortable level of trust in the strength of his data protection, because trusted third parties had done most of the work for us.

In the next meeting, it was "déjà vu all over again," to quote Yogi Bera. We had a similar discussion about a virtually identical product and process, and I asked the same questions but received starkly different answers. The second business partner had developed proprietary cryptography using its own homegrown algorithms.

It had also spurned using a cryptographic toolkit for development, and used native C++ code, with its own random number generator. From a security perspective, this was a disaster! Proprietary cryptography is usually a very poor substitute for the real thing, and often easy to break. The costs to hire a cryptographer, plus our team's involvement of 180 to 250 hours to certify the proprietary cryptography, would cost more than the expected profits. We could not trust this cryptography.

As I reflected back on these events at the end of the day, the stark contrast between the two meetings struck me. The first partner had enabled me to determine the risk of the transaction with just a few questions. By using protocols, toolkits, and a certificate authority that met our standards, and then wrapping it in a completed assessment delivered by a reliable third party, we had enough information to make a decision about the trustworthiness of their application. Then it hit me — what if trust assertions, during E-business transactions, could be driven down to this level of simplicity? How might that change the process for determining vendor trustworthiness?

Business Driver 1: Acceleration of E-Business through Web Services

In case you have been asleep for the past ten years, business acceleration through technology is moving at an incredible rate. Where formerly a typesetter could produce a book in three to six months, the same book can now be printed on-demand and bound in five minutes. Business formerly done through a handshake over dinner at the club is now brokered autonomously through EDI. Massive diversification, outsourcing, insourcing, and merger and acquisition fragmentation constantly shift the way industries and partners connect their networks together. In this arena, the latest force that promises to revolutionize the speed and ease of doing business is Web Services.

Consider the business and trust decisions that are made through conducting E-business in a Web Services transaction. Web Services offerings provide a near "instant-on" delivery of services and information through interoperable lightweight protocols: largely HTML, XML, and SOAP. Web Services transactions are ideally conducted through flexible interfaces that permit corporations to provide adaptable and extensible access portals for their legacy business logic and processes. Read the previous sentence again. It is not rhetoric; it is sincere. Consider the power of being able to broker a Web Service to a partner by providing an interface to the same object model that rests at the core of your business application. When you roll out new business functions for your business application, providing that same interface through your Web Service portal is a relatively simple operation. Imagine how quickly your business could pivot and respond to market changes if you remove the need to create and update yet another presentation and application layer each time you make a change. At long last, object technology is making a difference on the bottom line as competitive advantage is achieved by slashing time-to-market. This is a powerful force that is driving business to the speed of thought.

However, it is not just B2B (business-to-business) reaping the benefits of Web Services. Since mobile computing is driving the computing power of the individual to this ubiquitous edge, Web Services promise to deliver instant menus on-demand to handhelds when standing down the street, or perhaps instant call-ahead reservations for your table two minutes from now. These, however, are low-trust transactions, as there is little harm in a menu item being left off, or a reservation for four being dropped to two. In contrast, consider the trust necessary to conduct stock trades through the same device. This type of transaction requires a starkly different trust model for conducting instant business, particularly in B2C (business-to-commerce).

The need to provide stated trust levels for instant services executed on behalf of others is one of the driving forces behind the Security Assertions Markup Language (SAML), although it may not be immediately apparent. Most consumers do not have digital certificates, so consumers cannot easily assert their identity through a third party. Consumers' authentication is necessarily going to be brokered for B2B2C transactions for the foreseeable future. If you need that translated, brokered authentication equates to cross-realm authentication, or cross-company authentication. This is also referred to in some industry groups as "federated identity." With apologies to entities and organizations that do not call themselves a "company" (e.g., the FBI, the Republican Party, and UCLA), I will refer to this authentication as cross-company authentication, or CCA. Although CCA can carry substantial risks, because you are permitting another organization to manage the authentication credentials to your site, CCA is one of the drivers promising single sign-on between business partners and portals. There are already a substantial number of CCA projects that have been implemented in finance, government, and other industries, largely through proprietary (expensive) protocols. As an example, when consumers connect to Travelocity to book airfare on America West Airlines, they do not have to log in to Sabre

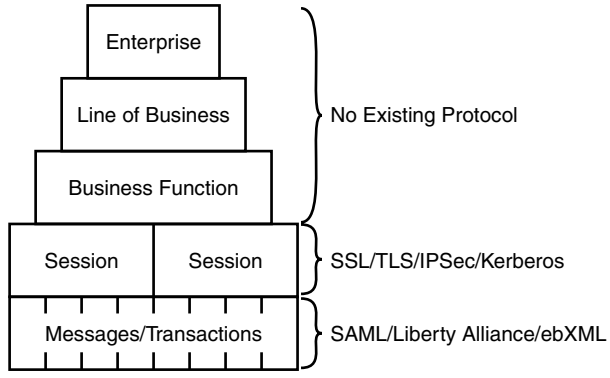


EXHIBIT 62.1 Protocols providing trust.

or America West. Rather, Travelocity provides that authentication for them, as a brokered authentication session for that transaction conducted on their behalf. Through proprietary protocols, Travelocity asserts the identity of the consumer to Sabre, and Sabre asserts the identity of Travelocity to America West. This is a familiar model, but expensive to replicate, due to the proprietary protocols. Resolving this expensive implementation is the promise of SAML, as it provides the ability to ensure that the trust inherent in transactions is asserted through an interoperable authentication assertion protocol. Because it is interoperable, it is repeatable.

Web Services and cross-company authentication drive business to the cusp of instantaneous trust decisions because E-business and E-commerce trust decisions must be delivered within a very short click-stream. Remember: your customers may only tolerate ten seconds of Internet lag before they jump to a competitor's site, and connecting with the Web Service interface has already spent 15 percent of that time, so your model must be able to make a trust decision in two seconds or less.

Before you make that trust decision, there are some things to consider, not least of which is determining what you are trusting. Smart and well-meaning IT and business professionals are often far too loose with how they use the term "trust." If you listen to the Root Certificate Authorities discuss trust, many of them call their certificates "trust," which is over-simplified. Trust, at its core, is derived from "trustworthiness." The trust extended by a digital certificate through SSL is merely trust in the identification of the party on the other end of the SSL tunnel, which is *session trust.*

There are multiple levels of trust that must be embraced to conduct E-business, and digital certificates cannot provide trust in individual transactions/messages, business functions, or corporations (see Exhibit 62.1). Unfortunately, there are many companies that have trustworthy digital certificates about untrustworthy transactions or their untrustworthy corporations. I am sure that WorldCom stockholders can help sharply differentiate the "trust" embodied in the SSL session that the lock icon in their browser assured them of as they connected to AOL and the trust they had in WorldCom after news of their financial scandal was made public.

It is this core differentiation, between the trustworthiness of a session and the trustworthiness of an organization, that is so problematic in the paradigm of Web Services, where instant trust decisions are necessary for trusted transactions. If this trust decision is further permitting outsourced authentication through CCA, the criticality of this trust is even more acute. How is it that you can determine, instantly, at the moment a Web Service transaction is occurring, if you can trust the security and privacy posture of a business partner or vendor?

If you do not think this is a problem, there are even stronger business drivers for trust governance.

Business Driver 2: Death by 1000 Cuts

When establishing new business relationships, companies must determine a number of things about their new partner, including their security posture, financial strength, and a number of other factors that help measure the trustworthiness of the organization. Because of the need to attest to the security of a partner organization, most organizations have reverted to making determinations of security posture through large proprietary checklists of questions that are exchanged and completed. Typically, after asking hundreds (or thousands) of probing questions, security and privacy analysts review the answers, score the organization's security posture,

and make some report of findings. Based on the report, management is able to make some basic decision regarding the trustworthiness of the organization. As organizations become more interconnected with business partners and vendors, these stacks of checklists arrive more frequently. Many organizations have had to hire multiple employees to do nothing more than manage incoming and outbound ad-hoc security assessments. Furthermore, the margin of error after completing hundreds of manual and subjective assessments makes it likely that a few untrustworthy organizations were given a clean bill of health. However, that is only a small part of the problem. If the security policy of your company changes, many of your assessments are no longer valid. If your policy is strengthened, all the assessments you performed were measured against an old standard that was weaker, so your trust models are no longer valid. Your business practices at that point could be construed as no longer providing due diligence in protecting your information assets, because your effective security policy is less than your stated policy. If you have instead relaxed your security policy, all the assessments you provided to other organizations have been invalidated, which could represent liability.

Business Driver 3: Trust Erosion

Imagine that you moved to a new city six months ago, and your car's air conditioning goes out. How do you find a mechanic you can trust? Of course, you ask your neighbor, a friend at work, or a buddy at the gym where you can find a trustworthy mechanic. Five years later, their recommendation is now stale. If you did not return to the garage for five years, how would you know that that garage could still be trusted? This is one of the problems of trust, which degrades over time.

Security assessments, like any audit function, are out-of-date the minute they are performed. Simply put, an assessment from yesterday or last year cannot provide any trustworthiness in a vendor's current or future security posture. Assessments are also largely undertaken during the intense scrutiny brought about through purchasing, merger and acquisition, and contract negotiation activities, and may never be conducted again at that level of diligence. Very few organizations will undertake the expense of performing monthly, quarterly, or even annual audits of all business partners and vendors, so their partners' security postures are unknown at any given point. You certainly have a contract with various vendors who manage some of your customer, employee, or financial data. If one of those vendors eliminates employee bonding and drug testing, removes most firewalls, moves its hosting servers from a data center to a warehouse, and converts long-term debt into junk bonds, would your organization ever know?

At this point, lawyers will interject that contracts should protect the organization, if they have required that the business partner provide a stated level of security. This is true, but contracts can rarely provide adequate compensation for reputation and consequential damages from a serious security breach. There is no monetary salve that can erase your company's name from bad press.

Business Driver 4: Lack of Common Standards

Corporations today are inundated with a variety of organizations asking them to attest to their level of security in dozens of unique and proprietary ways, and there is no single security standard used in all organizations to provide a measurement of trust. Although many fine standards exist, such as ISO 17799, COBIT, GASSP, and the Common Criteria, they are so massive and stringent that few organizations can be 100-percent compliant with any one of them, and could never hope to achieve compliance with more than one. Some of these standards, such as the Common Criteria, make generalities about a corporation's recommended security stance, regardless of industry, which is patently false. To expect that all industries could (or should) have similar security standards and policies is ludicrous. While most U.S. financial organizations conduct drug tests and background checks when hiring, most higher education institutions would likely consider such measures heavy-handed, and might instead focus on evaluating employee curriculum vitae and transcripts. At the other end of the spectrum, the CIA, NSA, and defense contractors commonly use lifestyle polygraph tests on employees holding sensitive positions, security measures that would certainly not be viable for most organizations.

By contrast, other standards, such as GASSP, are such a generalized framework that compliance with GASSP is largely subjective. An assessment that asserted compliance with GASSP would not be useful to another entity because the underlying principles of best practice necessary to ensure compliance with GASSP are largely open to interpretation.

Clearly, a common standard or framework is needed so that assessments, conducted by one company's auditors, can be easily evaluated by partner companies to save expense and provide a better determination of trust.

Business Driver 5: Security Standard Regulation

At the RSA 2002 Conference, Bruce Schneier proposed a vision of the future that I found startling. It took several months for me to realize his vision was sound, although I disagreed with his forecasted end-result. Schneier related the following timeline, which I present with my revisions:

- The "hacker insurance" market will continue to grow as CEOs look for ways to transfer risk.
- Small security events, such as the SQL Slammer attack of February 2003, will dramatically increase interest in acquiring hacker insurance.
- Many insurers will aggressively enter the hacker insurance market, seeking lucrative policies.
- Massive security events like Melissa and Code Red are evolving to resemble worldwide weather patterns and acts of God, rather than focused or targeted attacks. Just like weather-induced floods and earthquakes, these massive security events will attack quickly, universally, and cause widespread damage.
- A massive security event of Code Red proportion will overwhelm networks worldwide, and cause widespread, actual damage to corporate information through a destructive payload. Billions in damages will be registered, and several insurance companies will face crippling losses or go out of business.
- Just as insurers formed Underwriters Laboratories (UL) in the 1890s because they were tired of paying for electrical fires, insurers will again turn to UL and ask them to establish security standards that will be required for hacker insurance coverage.
- UL standards for security best practice will be published, and will start to appear in contracts as a base requirement to protect participants, just as ISO 9000 showed substantial growth through pressure by Fortune 500 companies on their supply chains.
- Eventually, UL standards will become codified by various government bodies and gain the rule of law, similar to the adoption of UL wiring and fire suppression standards that have been codified through municipal building codes.

Although some industries might welcome this regulation of security standards, many others would rather develop and police their own standards to ensure they were meeting their particular needs, rather than the needs of the insurers. We must either develop our own standards (soon!) and police ourselves, or the choices on how we secure our company data may be forced upon us, just as privacy law has done.

The Trust Governance Answer: Certified Trust Assertions

We return now to the original story, about the two business partners with starkly different cryptography solutions, and the idea it sparked in my brain.

I realized that if industries developed a simplified set of standards, perhaps they could simply state, with 100 to 300 statements, enough indicators about their security posture to enable others in the same industry to determine their trust status. They might be able to say six things about how they secure firewalls, five things about their host server hardening, five statements about their CIRT, three statements about hiring practices, etc. Once you have these answers, you can readily determine the high-level trustworthiness of a partner in the same industry. Perhaps 50 common questions could be established across all industries, and several more questions would provide the industry-specific standards.

This industry-specific security standard solves the problem of a common standard that can be evaluated, but does not address the timeliness of trust decisions. However, that model is even easier to provide. The same consortium that establishes the standard 100 to 300 questions also establishes the standards that certified auditors must meet in testing and reporting compliance with those standards. When an assessment is completed, the auditor, likely a major accounting firm, would generate a score and statement of scope. With this information, an organization would only need to know the answer to five questions to determine trustworthiness:

1. Who provided the audit?
2. What standard was used?
3. What was the score?
4. What was the scope of the audit?
5. What date was the audit conducted?

EXHIBIT 62.2 Example of What a Xota Assertion Would Look Like

Standard:	ISO17799-ABCDE
Score:	6.7.19.22.8.5.9.4.2.5.6.x.x.x.x.x.x.x.x.x
Score (Raw):	CACEADD9F7BFF7FDFF7B6D90E7D8CA04106C8B70
ORG:	O = EXAMPLE ORG; C = US; OU = BANKING;CN = CCU_APP
Included:	OU = BANKING
Excluded:	NONE
Date:	20020103101411.2Z

EXHIBIT 62.3 Answers to the Five Questions

Q1: Who provided the audit?
A1: "PDQ Audit Solutions" provided the audit.
Our business accepts them. Passed.

Q2: What standard was used?
A2: The standard was ISO 17799-ABCDE.
That's a standard we support. Passed.

Q3: What was the score?
A3: The score was 6.7.19.22.8.5.9.4.2.5.6.
Minimum for ISO 17799-ABCDE is 5.5.17.22.8.2.9.3.2.3.3. Passed.

Q4: What was the scope of the audit?
A4: The scope was the OU = Banking.
Business app is in Banking Division. Passed.

Q5: What date was the audit conducted?
A5: The date was 1/3/2002.
Maximum age is 18 months. Failed. Untrusted state.

These questions cover everything a relying party needs to know, provided they trust the standard and the auditor, and have established scoring criteria for their organization. However, the real power is yet to come. As a required deliverable of the assessment, the auditing party would issue a digital signature that contains these five fields of information, and is issued by a trusted Root Certificate Authority. Because an X.509 certificate can be readily verified, such credentials would be nearly impossible to forge.

When connecting to a business application, part of the connection negotiation would require exchange of digital certificates to establish secure communications, and the trust assertion certificate could be included in this handshake (see Exhibit 62.2). The relying party can then extract the information, verify the public key, ensure that the integrity of the information is intact, and that the digital certificate has not been revoked. That process takes sub-seconds to perform, and is a time-honored process currently used in SSL.

For each business application, all organizations taking part in the transaction would establish minimal scoring standards for that application, aligned with the stipulations in the contractual agreement. The trust assertion analysis process then checks those baseline standards against the assertions, represented as the answers to those five questions detailed above (see Exhibit 62.3).

Again, because these standards are an extension of contractual obligations stipulated in the agreement between the two companies, the terms should be clear to all involved, and the trust analysis merely reflects the trust embodied in the contract. Because the standard and scope are flexible, each business application can determine what level and scope are required for the trust posture necessary for the application. Scope can readily be defined through embedded XML in the certificate, providing a pointer to LDAP-compliant Fully Distinguished Names (FDNs) and Relative Distinguished Names (RDNs). This would permit the application to determine how much of the infrastructure was covered by the auditor's assessment.

In addition to providing scoring ranges for quick compliance scoring, the answer to each compliance question would also be communicated in hexadecimal notation (00-FF), providing a compact means of conveying assessment information, to facilitate very granular compliance analysis. For example, "FC" represents scoring answers of 11111100, which indicates the company is compliant with requirements 1 through 6, but is not compliant with standards 7 and 8.

Basing Certified Trust Assertions on X.509 is not an accident, because Certificate Revocation List (CRL) checking becomes an integral control of the methodology. If a security score downgrade is determined through an audit, event, or discovery, the auditor would be required to revoke the prior certificate and reissue the new one (which is a simple process). For a security score upgrade, a new certificate would be issued without revoking the "weaker" assertion, to avoid a denial-of-service in ongoing transactions using trust assertions. Checking certificate revocation against Certificate Revocation Lists ensures that the trust rating is still viable, and provides protection against fraud. Further, it permits all other parties relying on that trust assertion to know, near instantaneously, that the trust model has changed for that transaction.

This methodology, of determining trustworthiness through exchange of standards-based assertions of trust from third parties, is the core of the recently developed Xota protocol. Xota — eXtensible Organizational Trust Assertions — is the combination of the methodology and practices, which use trusted third parties, with the lightweight protocol to report the scope and standard used during the assessment, and the "score" of that assessment.

Because the trust assertion is provided via lightweight and ubiquitous X.509 digital certificates, nearly any system designed to provide authentication could readily request and analyze trust assertions. Both traditional client/server E-commerce and Web Services business applications can dynamically determine session trust, application trust, and entity trust, all at execution time. The same technology could be embedded in SSH to determine trustworthiness for logins, to protect file transfers and terminal emulation sessions. By placing trust assertion processing in e-mail gateways, spam can be deflected or routed based on the trust assertions embedded with the message, either by mail router trust assertions or those from the author's systems.

Trustworthiness of executables could also be governed if a secure kernel would not only verify the integrity against known, signed hashes, but would also perform trust assertion validation. By performing an assessment of the executable's Xota trust assertion, most importantly by assessing the viability of the certificate against a CRL, the kernel would be able to determine if the executable had lost its certification, perhaps because vulnerabilities had been published against that version of the application. Implementing such a methodology would require some serious shoring up of the certification and vetting process to ensure that a bogus CRL did not cause a massive denial-of-service attack, but still presents useful extensions for a trustworthy computing environment, particularly in a government or military application requiring certified executables.

Xota trust modeling is also viable for the B2C environment, and could be built into a browser quite easily to provide a security assessment automatically, just as PGP does for privacy. Java applets, ActiveX controls, JavaScript, VBScript, and embedded components could be required not only to be signed, but also to include a trust assertion for the application. Consumers would be able to determine the trustworthiness of the application, company, and privacy controls automatically at download, which could be a very powerful tool for consumers to identify malicious payloads, spyware, and untrustworthy companies.

Conclusion

The technical challenges to building a Xota-compliant trust assertion model are minimal, as all the necessary components exist today. Common standards would be helpful, but merely provide implementation lubrication to remove barriers and expense from implementation. Most of the assessments are already being conducted as part of vulnerability assessments, SAS 70 audits, and regulatory compliance assessments. The process could technically be implemented tomorrow by using an existing standard (e.g., ISO 17799), although it will almost certainly take the establishment of a consortium to develop standards that will evoke trust by participants. Additionally, the consortium should develop auditing standards and auditing certification processes to ensure that issuers of trust assertions follow the standards.

The benefits realized from developing this system speak directly to the five business drivers introduced earlier. Presuming adoption of the Xota protocol by business partners within one or more industries, what might trust governance look like within these paradigms?

Acceleration of E-Business through Web Services

By utilizing Xota trust assertions as an integral component of their Web Services offerings, business partners can now interconnect their Web Services very quickly. UDDI and WSDL are two protocols that permit Web Services and their interfaces to be published in a meta-directory, and hold the promise of "drag-and-drop"

Web Services interface deployment. However, they are currently only used for referencing low-value transactions, due to the lack of trust and contractual assurances. By utilizing Xota trust assertions, UDDI and WSDL could also be used for high-value Web Services transactions. This would mean that the only remaining barrier for instant-on Web Services is contract negotiation. Business partners can now react very quickly to market changes by rolling out Web Services interfaces to existing and new partners in days instead of months, because the security and interface barriers can be identified within minutes instead of weeks.

Several consortiums and business groups are currently working to create "circles of trust" within industries, to permit Single Sign-on through federated identity management. However, these circles of trust are constrained when they cross industry, country, and other trust barriers. If these business trust models use Xota trust assertions to provide a common language and framework for trust modeling, these circles of trust may no longer be constrained to single industries or circles, but can now enable the rapid deployment of cross-industry E-business.

Death by 1000 Cuts

The most striking effect from implementation of trust governance lies in the compliance and assessment functions within organizations. By implementing rapid assessments with the Xota protocol, the tasks of establishing, assessing, and governing business trust models becomes an automated process. Further, by moving the trust assessments to the transaction point, compliance with the business trust model is provided automatically and proactively. Trust assertions can easily be forwarded to a central repository for further compliance analysis.

Once Xota trust modeling is implemented, compliance organizations can shift compliance workers from a cost to a revenue basis. Instead of the drudgery of assessing and reporting on security models, security knowledge workers can focus on building and extending trust models. Security assessments become a key business enabler, rather than a cost sink. Further, the hidden costs of continuous assessments and governance are converted into hard-dollar infrastructure and application costs that can be included in budgets for projects that implement those risks, rather than being borne by the security and compliance organizations as overhead. The risk posture of partnerships can also be determined and evaluated at the point of project initiation, rather than weeks later. By attaching costs and risks to the projects that generate them, senior management can make more-informed decisions on project return on investment (ROI) and return on risk.

Trust Erosion

Although most contracts today include verbiage that permits a periodic or unscheduled on-site visit by one or both parties of the contract, this assessment is rarely executed, due to the high cost of such assessments. However, with the availability of continuous determinations of trust compliance, these components of contract compliance are now verified automatically. If the contracts are structured to require periodic third-party assessments and Xota assertions, the trust models can be self-regulated through ongoing analysis of the trust assertions.

Common Standards

Through the creation of a common framework and language for discussing standards compliance, Xota permits translations of assessments across international and industry boundaries. If an assessment was provided against the Common Criteria standard, but the organization has based its policies and trust models on BS 7799, the assessment can still be used by the business partners. The relying organization would have to assess the individual answers to all the questions of the "new" standard, and then determine what its requirements would be within that business context. Once completed, the organization would have a template that can be used to translate Common Criteria to BS 7799, and this could be extended to other trust models in the organization. Although Xota does not provide a common standard, it does provide a common language for interpreting standards, and permits wide reuse of assessments across many isolated contexts.

Security Regulation

Security regulatory proponents primarily cite the need for regulation to establish and enforce common standards. With industry-wide adoption of Xota and the underlying standards, regulators can assess the compliance of organizations within their jurisdictional purview without the need to create yet another security standard. Insurers could likewise determine the risk posture of policyholders, and reward security diligence (or punish poor security) through a tiered pricing structure. By moving industries to a common language for communicating compliance with existing standards, the need to regulate security evaporates. Governing and regulatory bodies are able to provide compliance metrics and oversight without the need to enforce monolithic standards across the industry, and organizations are able to report their security posture without necessarily migrating to yet another security standard.

The power of Xota as the language of trust governance extends from the ability to make a clear determination of trustworthiness with five simple questions that can be dynamically assessed. The instant payoff from implementation is the ability to determine the trustworthiness of business partners without long checklists and expensive manual processes; and by ensuring that businesses, divisions, and applications are trustworthy at the point that messages and transactions are processed.

63

Risk Management and Analysis

Kevin Henry, CISA, CISSP

Why risk management? What purpose does it serve and what real benefits does it provide? In today's overextended work environments, it can easily be perceived that "risk management and analysis" is just another hot buzzword or fashionable trend that occupies an enormous amount of time, keeps the "administrative types" busy and feeling important, and just hinders the "technical types" from getting their work done.

However, risk management can provide key benefits and savings to a corporation when used as a foundation for a focused and solid countermeasure and planning strategy.

Risk management is a keystone to effective performance and for targeted, proactive solutions to potential incidents. Many corporations have begun to recognize the importance of risk management through the appointment of a Chief Risk Officer. This also recognizes that risk management is a key function of many departments within the corporation. By coordinating the efforts and results of these many groups, a clearer picture of the entire scenario becomes apparent. Some of the groups that perform risk management as a part of their function include security (both physical and information systems security groups), audit, and emergency measures planning groups.

Because all of these areas are performing risk analysis, it is important for these groups to coordinate and interleave their efforts. This includes the sharing of information, as well as the communication of direction and reaction to incidents.

Risk analysis is the science of observation, knowledge, and evaluation — that is, keen eyesight, a bit of smarts, and a bit of luck. However, it is important to recognize that the more a person knows, the harder they work, often the luckier they get.

Risk management is the skill of handling the identified risks in the best possible method for the interests of the corporation.

Risk is often described by a mathematical formula:

$$\text{Risk} = \text{Threat} * \text{Vulnerability} * \text{Asset value}$$

This formula can be described and worked quite readily into the business environment using a common practical example. Using the example of the bully on the playground who threatens another child with physical harm outside the school gates after class, one can break down each component as follows:

- The threat is that of being beat up, and one can assign a likelihood to that threat. In this case, say that it is 80 percent likely that the bully will follow up on his threat unless something else intervenes (a countermeasure — discussed later).
- The vulnerability is the other child's weakness. The fact that the other child is unable to defend himself adequately against this physical attack means that the child is probably 100 percent likely to be harmed by a successful attack.

- The asset value is also easy to calculate. The cost of a new shirt or pants, because they will probably be hopelessly torn or stained as a result of an altercation and the resultant bloody nose, puts the value of the assets at $70.00.

Therefore, the total risk in this scenario is:

$$\text{Risk} = 80\% * 100\% * \$70.00.$$

$$\text{Risk} = \$56.00$$

Now one can ask: what is the value of this risk assessment? This assessment would be used to select and justify appropriate countermeasures and to take preventative action. The countermeasures could include hiring a bodyguard (a firewall) at a cost of $25.00, not going to school for the day (like shutting the system down and losing business), or taking out insurance to cover losses. The first of these primarily deals with strengthening the weakness(es) or vulnerabilities, while the third protects the asset value. Preventative action would include methods of reducing the threats, perhaps by befriending the bully, working out or learning karate, or moving out of town.

Thus, from this example, it is easy to describe a set of definitions in relation to risk management.

- Risk is any event that could impact a business and prevent it from reaching its corporate goals.
- Threat is the possibility or likelihood that the corporation will be exposed to an incident that has an impact on the business. Some experts have described a threat both in a positive sense as well as in a negative sense. Therefore, it is not certain that a threat will always have a negative impact; however, that is how it is usually interpreted.
- A vulnerability is the point of weakness that a threat can exploit. It is the soft underbelly or Achilles' heel where, despite the tough armor shielding the rest of the system, the attack is launched and may open the entire system or network to compromise. However, if risk is viewed as a potentially positive scenario, one should replace the term "vulnerability" with the term "opportunity" or "gateway." In this scenario, the key is to recognize and exploit the opportunity in a timely manner so that the maximum benefit of the risk is realized.
- The asset is the component that will be affected by the risk. From the example above, the asset was described as the clothing of the individual. This would be a typical quantitative interpretation of risk analysis. Quantitative risk analysis attempts to describe risk from a purely mathematical viewpoint, fixing a numerical value to every risk and using that as a guideline for further risk management decisions.

Quantitative Risk Analysis

Quantitative risk analysis has several advantages. It provides a rather straightforward result to support an accounting-based presentation to senior managers. It is also fairly simple and can easily follow a template type of approach. With support and input from all of the experts in the business groups and supporting research, much of the legwork behind quantitative analysis can be performed with minimal prior experience. Some of the steps of performing risk analysis are addressed later in this chapter.

However, it is also easy to see the weaknesses of quantitative risk analysis. While it provides some value from a budget or audit perspective, it disregards many other factors affected by an incident. From the previous example, how does one know the extent of the damage that would be caused by the bully? An assumption was made of generally external damage (clothing, scrapes, bruises, bloody nose), but the potential for damage goes well beyond that point. For example, in a business scenario, if a computer system is compromised, how does one know how far the damage has gone? Once the perpetrator is into a system and has the mind to commit a criminal act, what limits the duration or scope of the attack? What was stolen or copied? What Trojan horses, logic bombs, or viruses were introduced. What confidential information was exposed? And in today's most critical area, what private customer details or data were released. Because these factors are unknown, it is nearly impossible to put a credible number on the value of the damage to the asset.

This chapter, like most published manuscripts these days, is biased toward the perception of risk from a negative standpoint. On the other hand, when risk is regarded in a potentially positive situation, there is the difficulty of knowing the true benefit or timing of a successful exploitation of an opportunity. What would be the effect on the value of the asset if a person reacts today rather than tomorrow, or if the opportunity is

missed altogether and the asset (corporation) thereby loses its leading-edge initiative and market presence? A clear example of this is the stock market. It can be incredibly positive if a person or company knows the ideal time to act (seize a risk); however, it can be devastating to wait a day or an hour too long.

Some of the factors that are difficult to assess in a quantitative risk analysis include the impact on employees, shareholders or owners, customers, regulatory agencies, suppliers, and credit rating agencies.

From an employee perspective, the damage from a successful attack can be severe and yet unknown. If an attack has an effect on morale, it can lead to unrealized productivity losses, skilled and experienced employee retention problems, bad reactions toward customers, and dysfunction or conflict in the workplace. It can also inhibit the recruitment of new, skilled personnel.

Shareholders or owners can easily become disillusioned with their investments if the company is not performing up to expectations. Once a series of incidents occur that prevent a company from reaching its goals, the attraction to move an investment or interest into a different corporation can be overpowering. Despite the best excuses and explanations, this movement of capital can significantly impact the financial position of the corporation.

Customers are the key to every successful endeavor. Even the best product, the best sales plans, and the best employees cannot overcome the failure to attract and retain customers. Often, the thought can be that the strength of a company can rest in a superior product; however, that is of little value if no one is interested in the services or products a company is trying to provide. A company with an inferior product will often outperform the company with superior products that gets some "bad press" or has problems with credibility. A lifetime warranty is of no value if the company fails because the billing system being used is insecure.

Regulatory agencies are often very vulnerable to public pressure and political influence. Once a company has gained a reputation for insecure or vulnerable business processes, the public pressure can force "kneejerk" reactions from politicians and regulatory agencies that can virtually handcuff a firm and cause unreasonable costs in new controls, procedures, reports, and litigation.

One of the best lessons learned from companies that have faced serious disasters and incidents is to immediately contact all major customers and suppliers to reassure them that the company is still viable and business processes are continuing. This is critical to maintaining confidence among these groups. Once a company has been exposed to a serious incident, the reluctance of a supplier to provide new raw materials, support, and credit can cripple a firm from re-establishing its market presence.

Because of the possible impact of an incident on all of these groups, and the difficulty in gauging a numerical value for any of these factors, it has been asserted by many experts that a purely quantitative risk analysis is not possible or practical.

Qualitative Risk Analysis

The alternative to quantitative risk analysis is qualitative risk analysis. Qualitative risk analysis is the process of evaluating risk based on scenarios and determining the impact that such an incident would have.

For qualitative risk analysis, a number of brief scenarios of potential incidents are outlined and those scenarios are developed or researched to examine which areas of the corporation would be affected and what would be the probable extent of the damage experienced by those areas in the event that this scenario ever occurred. This is based on the best estimates of the personnel involved.

Instead of a numerical interpretation of a risk as done in a quantitative risk analysis, a ranking of the risk relative to the affected areas is prepared. The risk analysis team will determine what types of incidents may occur, based on the best knowledge they can gain about the business environment in which the company operates. This is similar to the financial modeling done by strategic planning groups and marketing areas. By rolling out the scenario and inputting the variables that influence the event, the risk analysis team will attempt to identify every area that might be affected by an incident and determine the impact on that group based on a simple graph like "High Impact," "Medium Impact," "Low Impact" — or through a symbolic designation like 3,2,1 or 0 for no impact. When all of the affected areas are identified, the value for each area is summarized to gauge the total impact or risk to the company of that scenario occurring. In addition to purely financial considerations, some of the areas to include in this analysis are productivity, morale, credibility, public pressure, and the possible impact on future strategic initiatives.

Whenever doing a risk analysis of an information system, it is important to follow the guidelines of the AIC triad. The risk analyst must consider the availability requirements of the system. Is it imperative that it operates continuously, or can it be turned down for maintenance or suffer short outages due to system failure without

causing a critical failure of the business process it supports? The integrity of the data and process controls and access controls around the systems and the underlying policies they are built on also need a thorough review. Probably no area has received as much negative publicity as the risk of data exposure from breaches of confidentiality in the past few years. A large, well-publicized breach of customer private information may well be a fatal incident for many firms.

One of the best methods to examine the relationship between the AIC triad and risk analysis is to perform general computer controls checks on all information systems. A short sample of a general computer controls questionnaire appears in Exhibit 63.1. This is a brief survey compiled from several similar documents available on the Internet. A proper general computer controls survey (see Exhibit 63.1) will identify weakness such as training, single points of failure, hardware and software support, and documentation. All of these are extremely valuable when assessing the true risk of an incident to a system.

However, qualitative risk analysis has its weaknesses just like quantitative risk analysis does. In the minds of senior managers, it can be too loose or imprecise and does not give a clear indication of the need or cost-benefit analysis required to spur the purchase of countermeasures or to develop or initiate new policies or controls.

For this reason, most companies now perform a combination of these two risk analysis methodologies. They use scenario-based qualitative risk analysis (see Exhibit 63.2) to identify all of the areas impacted by an incident, and use quantitative risk analysis to put a rough dollar figure on the loss or impact of the risk according to certain assumptions about the incident. This presumes, of course, a high level of understanding and knowledge about the business processes and the potential risks.

The Keys

If one were to describe three keys to risk analysis, they would be knowledge, observation, and business acumen.

Knowledge

Effective risk analysis depends on a thorough and realistic understanding of the environment in which a corporation operates. The risk manager must understand the possible threats and vulnerabilities that a corporation faces. These managers must have a current knowledge of new threats, trends, and system components, tools, and architectures in order to separate the hype and noise from the true vulnerabilities and solutions to their organization. To gain the cooperation of the business areas to perform risk analysis, and to be able to present the resulting credible recommendations to senior managers, the manager must be able to portray a realistic scenario of possible threats and countermeasures. This knowledge is gained through the continuous review of security bulletins, trade journals, and audits. For this reason, a Chief Risk Officer should also sit on the senior management team so that he or she has knowledge of corporate strategic direction and initiatives. The Chief Risk Officer should also receive regular updates of all ongoing incidents that may have an impact on the corporation.

Observation

Observation is the second key. We live in an age of overwhelming data and communication. Observation is the ability and skill to see through all of the outside influences and understand the underlying scenarios. Observation is to review all tools and reports routinely to notice if any abnormal conditions are being experienced. It is noteworthy that many excellent audit logs and output reports from tools are sitting unused on shelves because it is too difficult and time-consuming for most individuals to pick out the details. When a person first installs an intrusion detection system on his home PC, he suddenly becomes aware of the number of scans and hits he is exposed to. Did those just commence when he installed his IDS? No, it is just that he was able to observe them once he had purchased the correct tools. Therefore, observations and the use of tools are critical to understanding the characteristics and risks of the environment in which they operate.

Business Acumen

The main reason for risk analysis is to get results. Therefore, the third key is business acumen, that is, the ability to operate effectively in the business world — to sense and understand the methods and techniques to

EXHIBIT 63.1 General Computer Controls Guideline Questionnaire

Objective:

When an Auditor is involved in the analysis of a system or process that involves a software tool or computer system or hardware that may be unique to that department, we are requesting that the Auditor fill out this questionnaire, if possible, during the performance of the audit.

This will allow us to identify and monitor more of the systems in use throughout the company, and especially to assess the risk associated with these systems and indicate the need to include these systems in future audit plans.

Thanks for your assisance; if you have any questions, please contact either Alan or myslf.

System Name and Acronym: _____

Key Contact Person: _____

Area where system is used: _____

Questions for initial meeting:

Please describe the system function for us: _____

_____

What operating platform does it work on (hardware)? _____

Is it proprietary software? Yes ☐ No ☐ Who is the supplier? _____

Does MTS have a copy of the source code? Yes ☐ No ☐

In which department? _____

Who can make changes to the source code?_____

Are backups scheduled and stored offsite? Yes ☐ No ☐

How can we obtain a list of users of the systems and their privileges? _____

Is there a maintenance contract for software and hardware? Yes ☐ No ☐

Can we get a copy? Yes ☐ No ☐

Separation of Duties:

Can the same person change security and programming or perform data entry? Yes ☐ No ☐

Completeness and accuracy of inputs/processing/outputs:

Are there edit checks for inputs and controls over totals to ensure that all inputs are entered and processed correctly? Yes ☐ No ☐

Who monitors job processing and would identify job failures? _____

Who receives copies of outputs/reports? _____

Authorization levels

Who has high-level authorization to the system? _____

Security — physical and configuration

Are the hardware and data entry terminals secure? Can just anyone get to them, especially high-level user workstations? Yes ☐ No ☐

Maintenance of tables

Are there any tables associated with the system (i.e., tax tables, employee ID tables)? Yes ☐ No ☐

Who can amend these tables? _____

EXHIBIT 63.1 General Computer Controls Guideline Questionnaire (continued)

Documentation

Is the entire system and process documented? Yes ☐ No ☐

Where are these documents stored? _____

Training of end users

Who trains the users? _____

Who trains the system administrator? _____

Is there a knowledgeable backup person? _____

DRP of System

Has a Disaster Recovery Plan for this system been prepared and filed with Corporate Emergency Management? Yes ☐ No ☐

Please provide an example of an input/output. _____

Any other comments:

_____
_____
_____
_____
_____

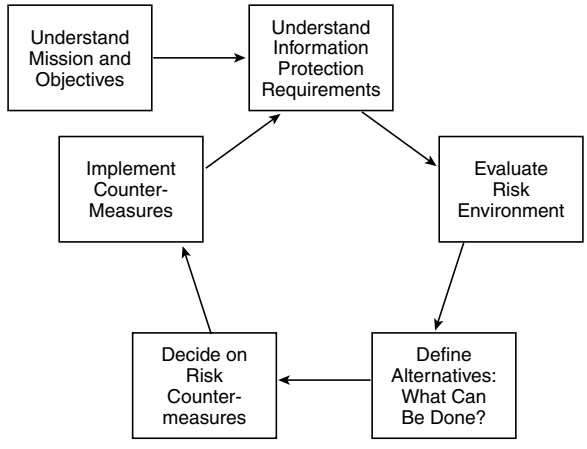


EXHIBIT 63.2 Risk analysis/management process.

use to achieve the desired results. Business acumen separates the average manager from the effective manager. With business acumen, they know how to get things done, how to make powerful and credible presentations, when to cry wolf, and when to withdraw. Because the whole foundation of risk analysis is based on understanding and addressing the mission of the business, risk managers must have the ability to set aside their traditional biases and understand the perspective of the business area managers at the same time they are evaluating risk and countermeasures. An ideal risk management solution requires the support of the users, the business area managers, and effective administration. This means that the solution must not be seen as too intrusive or cumbersome for the users nor having a significant performance or productivity impact on the supporting business systems or processes.

Risk Management

This is where the science of risk management comes into effect. Risk management is the careful balance between placing controls into the business processes and systems to prevent, detect, and correct potential incidents, and the requirement that the risk management solution not impede or restrict the proper flow and timeliness of the business.

Once the risk assessment has been completed, the result should be a concise overview of all possible threats to the organization. Included in this review will be a listing of all identified threats, areas potentially impacted by a threat, estimated cost or damage from an exposure (a threat actually being realized or occurring), and the key players in each business group.

From this assessment, the risk managers must evaluate whether or not the risk identified supports the adoption of some form of countermeasure. Usually, these countermeasures can be grouped into three categories: reduce, assign, and accept.

Reduce

To reduce the risk, most often some new control is adopted. These controls can be either administrative (balancing, edits, ID control, process change, or physical access rules) or technical (intrusion detection systems, firewalls, architecture, or new tools). By evaluating the true extent of the risk and the business requirements, the risk manager will develop a list of possible solutions to the risks. These solutions will then be evaluated on the basis of cost, effectiveness, and user acceptance before being presented for approval and implementation.

By this time in the risk analysis and management process, some of the initial fear or excitement that was driving the risk analysis process may be starting to wane. Personnel are moving on to new issues and can become desensitized to the threats that caused them sleepless nights only a few weeks before. This is where many risk management processes become derailed. Solutions are proposed and even purchased, but now the impetus to implement them dries up. The new tools sit ignored because no one has the time to look at them and learn all of their features. The controls are relaxed and become ineffective, and the budget does not provide the funding to continue the administrative support of the controls effectively. These can be dark days for the risk manager, and the result is often an incomplete risk analysis and management process. Now, at the very verge of implementation, the project silently subsides.

This is a challenge for the risk manager. The manager must rise to the occasion and create an awareness program, explain the importance of the new controls, and foster an understanding among the user community of how this risk solution can play a critical role in the future health of their department and the corporation.

Outsourcing

One alternate solution being explored by many companies today is a hybrid between the adoption of risk management tools and the assignment of risk management. This is the concept of outsourcing key areas of the risk management process. It is difficult for a corporation to maintain a competent, knowledgeable staff to maintain some of the tools and products needed to secure an information system. Therefore, they leverage the expertise of a vendor that provides risk management services to several corporations and has a skilled and larger staff that can provide 24-hour support. This relieves the corporation from a need to continually update and train an extensive internal staff group and at the same time can provide some proof of due diligence through the independent evaluation and recommendations of a third party. This does have significant challenges, however. The corporation needs to ensure that the promised services are being delivered, and that the knowledge and care of the corporate network entrusted to a third party are kept secure and confidential. Nothing is worse than hiring a fox to guard the chicken house. Through an outsourcing agreement, the risk manager must maintain the competence to evaluate the performance of the outsourcing support firm.

Assign

To assign the risk is to defer or pass some of the risk off to another firm. This is usually done through some insurance or service level agreement. Insurers will also require a fairly thorough check of the risks to the corporation they are ensuring to verify that all risks are acknowledged and that good practices are being followed. Such insurance should be closely evaluated to confirm that the corporation understands the limita-

tions that could affect a reimbursement from the insurer in the event of a failure. Some of the insurance that one will undoubtedly be seeing more of will be denial-of-service, E-business interruption, and Web site defacement insurance.

Accept

When a risk is either determined to be of an insignificant level, or it has been reduced through countermeasures to a tolerable level, acceptance of the residual risk is required. To accept a level of risk, management must be apprised of the risk analysis process that was used to determine the extent of the risk. Once management has been presented with these results, they must sign off on the acceptance of the risk. This presumes that a risk is defined to be at a tolerable level, either because it is of insignificant impact, countermeasure costs or processes outweigh the cost of the impact, or no viable method of risk prevention is currently available.

Summary

Risk analysis and management is a growing and exciting area. The ability of a corporation to identify risks and prevent incidents or exposures is a significant benefit to ensuring continued business viability and growth even in the midst of increasing threats and pressures. The ability of the risk managers to coordinate their efforts alongside the requirements of the business and to keep abreast of new developments and technologies will set the superb risk managers apart from the mundane and ineffective.

For further research into risk analysis and management, see the Information Assurance Technical Framework (IATF) at www.iatf.net.

64

New Trends in Information Risk Management

Brett Regan Young, CISSP, CBCP

Corporations have increased their investment in information security because critical business systems have moved into increasingly hostile territory. As the enterprise has embraced new technologies such as EDI/EFT, remote access, and sales automation, confidential data has gradually found itself in ever-riskier venues. Moving to the Internet is the latest — and riskiest — frontier. Nevertheless, forward-looking companies are willing to face the growing and unpredictable body of risks on the Internet to create competitive advantage.

Management of information risk is a new discipline, following on the heels of electronic information systems in general. To date, in the majority of organizations, information risk management has been done largely with a "seat of the britches" approach. The opinions of experts are often sought to assist with current protection needs while divining future threats. Electronic fortifications have been erected to improve an organization's defensive position. These measures, while allowing businesses to operate within the delicate balance of controls and risks, have had mixed success. This is not to say that organizations have not been hit by computer crime. The extent and frequency of such crimes have been historically low enough to give the impression that IS departments and security teams were managing information risk sufficiently well.

A Traditional Approach

Conventional risk analysis is a well-defined science that assists in decision support for businesses. The most common use of risk analysis is to lend order to apparently random events. By observing the frequency of an event factored by the magnitude of the occurrences, one can predict, with more or less accuracy, when and to what degree something might happen. Thus, one might expect ten earthquakes of a 7 magnitude to strike Yokohama within 100 years. When information is available to indicate the projected expense of each episode, then one can ascertain the ALE (annual loss expectancy). Conventional risk analysis is a powerful tool for managing risk, but it works best when analyzing static or slowly evolving systems such as human beings, traffic patterns, or terrestrial phenomena. Incidents that cause the loss of computing functions are difficult to map and even more difficult to predict. Two reasons for this are:

1. Trends in computing change so rapidly that it is difficult to collect enough historical data to make any intelligent predictions. A good example of this can be found in the area of system outages. An observer in California might predict that a server farm should suffer no more than one, three-hour outage in ten years. In 1996, that was plausible. Less than five years later and after an extended power crisis, that estimate was probably off by a factor of ten.
2. There is a contrarian nature to computer crime. Criminals tend to strike the least protected part of an enterprise. Because of the reactive nature of information security teams, it is most likely that one will

add protection where one was last hit. This relationship between attackers and attacked makes most attempts to predict dangerously off-track.

While information risk shares aspects with other types of business risks, it is also unique, making it difficult to analyze and address using conventional methods.

Doing Our Best

To protect their E-commerce operations, most businesses have relied primarily on an "avoidance" strategy, focusing on components such as firewalls and authentication systems. Daily reports of Internet exploits have shown that these avoidance measures, while absolutely essential, are not a sufficient defense. Avoidance strategies offer little recourse when incursions or failures do occur. And despite an organization's best efforts to avoid intrusions and outages, they will occur. In the high-stakes world of E-commerce, would-be survivors must understand this and they need to prepare accordingly.

Reports of Internet intrusions are frequent — and frightening enough to get the attention of management. Tragically, the most common response from corporate management and IS directors is a simple redoubling of current efforts. This reaction, largely driven by fear, will never be more than partially successful. It is simply not possible to out-maneuver Internet thugs by tacking new devices onto the perimeter.

The most telling metric of failed security strategies is financial. According to one source, funding for defensive programs and devices will increase an estimated 55 percent during the two years leading up to 2004, growing to a projected $19.7 billion for U.S. entities alone.[1] Keeping pace with rising computer security budgets are the material effects of computer crime. Dramatic increases in both the frequency and extent of damage were reported in the most recent annual Computer Security Institute (CSI)/FBI computer crime survey. The 273 respondents reported a total of $265 million in losses. These figures were up from the $120 million reported the previous year.[2] While the survey results are not an absolute measure of the phenomenon, it is a chilling thought to imagine that the enormous increases in security spending may not be keeping up with 50 percent and greater annual increases in material damage suffered as a result of computer-related crime.

The composite picture of rising costs for security chasing rising damages casts a dark shadow on the future of electronic commerce. Left unchecked, security threats coupled with security mismanagement could bring otherwise healthy companies to ruin. The ones that escape being hacked may succumb to the exorbitant costs of protection.

Common Sense

Who Let the Cows Out?

During the 1990s, a trend emerged among IS management to focus intensely on prevention of negative security events, often to the exclusion of more comprehensive strategies. There were three distinct rationales behind this emphasis:

1. *Experience has consistently shown that it is cheaper to avoid a negative incident than to recover from it.* This is most often expressed with a barnyard metaphor: "like shutting the gate after the cows are gone." The implication is that recovery operations (i.e., rounding up livestock after they have gotten loose) is infinitely more trouble than simply minding the latch on the gate.
2. *Loss of confidentiality often cannot be recovered, and there is, accordingly, no adequate insurance for it.* Valuing confidential information poses a paradox. All of the value of some types of confidential information may be lost upon disclosure. Conversely, the value of specific information can shoot up in certain circumstances, such as an IPO or merger. Extreme situations such as these have contributed to an "all-or-nothing" mentality.
3. *The "bastion" approach is an easier sell to management than recovery capability.* Information security has always been a hard sell. It adds little to the bottom line and is inherently expensive. A realistic approach, where contingencies are described for circumvented security systems, would not make the sale any easier.

The first argument makes sense: avoidance is cheaper than recovery in the long run. In theory, if new and better defenses are put in place with smarter and better-trained staff to monitor them, then the problem should

be contained. The anticipated results would be a more secure workplace; however, precisely the opposite is being witnessed, as evidenced by the explosive growth of computer crime.

The bastion approach has failed to live up to its expectations. This is not because the technology was not sufficient. The problem lies in the nature of the threats involved. One constant vexation to security teams charged with protecting a corporation's information assets is the speed with which new exploits are developed. This rapid development is attributable to the near-infinite amount of volunteer work performed by would-be criminals around the world. Attacks on the Internet are the ultimate example of guerilla warfare. The attacks are random, the army is formless, and communication between enemy contingents is almost instantaneous. There is simply no firewall or intrusion detection system that is comprehensive and current enough to provide 100 percent coverage. To stay current, a successful defense system would require the "perps" to submit their exploits before executing them. While this may seem ludicrous, it illustrates well the development cycle of defensive systems. Most often, the exploit must be executed, then detected, and finally understood before a defense can be engineered.

Despite the media's fascination with electronic criminals, it is the post-event heroics that really garner attention. When a high-volume E-commerce site takes a hit, the onlookers (especially affected shareholders) are less interested in the details of the exploit than they are in how long the site was down and whether there is any risk of further interruption. Ironically, despite this interest, spending for incident response and recovery has historically been shorted in security and E-commerce budgets.

It is time to rethink information protection strategies to bring them more in line with current risks. Organizations doing business on the Internet should frequently revise their information protection strategies to take into account the likelihood of having to recover from a malicious strike by criminals, a natural disaster, or other failures. Adequate preparation for recovery is expensive, but it is absolutely necessary for businesses that rely on the Internet for mission-critical (time-critical) services.

Exhibit 64.1 illustrates a simple hierarchy of information security defenses. The defenses garnering the most attention (and budget dollars) are in the lower three categories, with avoidance capturing the lion's share. Organizations will need to include recovery and bolster assurance and detection if they are to successfully protect their E-commerce operations.

A Different Twist: Business Continuity Management

Business continuity management (BCM) is a subset of information security that has established recovery as its primary method of risk management. Where other areas of information security have been preoccupied with prevention, BCM has focused almost exclusively on recovery. And just as security needs to broaden its focus on post-event strategies, business continuity needs to broaden its focus to include pre-event strategies. BCM in the E-commerce era will need to devise avoidance strategies to effectively protect the enterprise. The reason for this is time.

A review of availability requirements for Internet business reveals an alarming fact: there often is not enough time to recover from an outage without suffering irreparable damage. Where BCM has historically relied heavily on recovery strategies to maintain system availability, the demands of E-commerce may make recovery an unworkable option. The reason for this is defined by the fundamental variable of maximum tolerable downtime (MTD). The MTD is a measure of just how much time a system can be unavailable with the business still able to recover from the financial, operational, and reputational impacts.

EXHIBIT 64.1 Information Protection Model

Level	Examples
Recovery	Incident response, disaster recovery
Detection	Intrusion detection
Assurance	Vulnerability analysis, log reviews
Avoidance	Firewalls, PKI, policy and standards

Courtesy of Peter Stephenson of the Netigy Corporation.

E-commerce has shortened the MTD to almost nil in some cases. A few years back, a warehousing system might have had the luxury of several days' recovery time after a disaster. With the introduction of 24/7 global services, an acceptable downtime during recovery operations might be mere minutes. In this case, one is left with a paradox: the only workable alternatives to recovery are avoidance strategies.

Referring again to the information protection model, shown in Exhibit 64.1, BCM now requires more solutions at the lower avoidance area of the hierarchy. Discussing these solutions is not within the scope of this chapter, but examples of enhancing availability include system redundancy, duplexing, failover, and data replication across geographical distances. Another indication of the shift in focus is that business continuity now requires a greater investment in assurance and detection technologies. In 2002, was likely that a company's Web presence would fail as a result of a malicious attack because it is from a physical failure. Business continuity teams once relied on calls from end users or the helpdesk for notification of a system failure; but today, sophisticated monitoring and detection techniques are essential for an organization to respond to an attack quickly enough to prevent lasting damage.

The makeup of business continuity teams will likewise need to change to reflect this new reality. A decade ago, business continuity was largely the domain of subject matter experts and dedicated business continuity planners. The distributed denial-of-service attacks witnessed in February 2000 spawned ad hoc teams made up of firewall experts, router jocks, and incident management experts. The teams tackled what was, by definition, a business continuity issue: loss of system availability.

Reworking the Enterprise's Defenses

One only needs to look back as far as the mid-1990s to remember a time when it seemed that we had most of the answers and were making impressive progress in managing information risk. New threats to organizational security were sure to come, but technological advances would keep those in check — it was hoped. Then the rigorous requirements of protecting information within the insecurity of a wired frontier jarred us back to reality. Waves of malicious assaults and frequent outages suggest that it may be a long time before one can relax again. But one should take heart. A thorough review of the current risk terrain, coupled with renewed vigilance, should pull us through. It is quite clear, however, that organizations should not expect to improve the protection of their environments if they continue to use the same strategies that have been losing ground over the past several years. Coming out on top in the E-commerce age will require one to rethink positions and discard failed strategies.

It should be encouragement to us all that reworking the enterprise's defenses requires more rethinking than retooling. Many of the requisite techniques are already resident in the enterprise or can be readily obtained. In recommending a review of an organization's defensive strategy, four principal areas of analysis need to be applied. They are presented below.

Security Architecture

Building an appropriate security architecture for an organization requires a thorough understanding of the organization's primary business functions. This understanding is best obtained through interviews with business leaders within the organization. Once discovered, the primary business functions can be linked to information technology services. These, in turn, will require protection from outside attack, espionage, and systems outage. Protecting IS services and systems is accomplished using security practices and mechanisms. Thus, the results of a security architecture study relate the activities of the information security group back to the primary business of the company.

The results of a security architecture study are particularly enlightening to businesses that have recently jumped onto the Internet. Quite often, businesses will have security processes and mechanisms protecting areas of secondary criticality while new business-critical areas go unprotected. Devising an effective architecture model allows an organization to allocate sufficient resources to the areas that need the most protection.

An additional benefit of the results of a security architecture study lies in its bridging function. Security architecture tracks relationships between information security and business functions that it protects, demonstrating the value of information security to the enterprise. The resultant insights can prove quite valuable as a support tool for budgetary requests.

Business Impact Analysis

Business impact analysis (or BIA) has been used as an essential component of business continuity planning for some years. The BIA estimates the cost per time unit of an outage of a specific system. Once this cost is known for a specific system (e.g., $100,000 per day), then informed decisions can be made concerning the system's protection. In addition to the practical uses for such information, the cost of a potential outage is the type of information that makes corporate management less reluctant to budget for protective measures.

The BIA has been a tool employed almost exclusively by business continuity planners until very recently. As malicious attacks on E-commerce availability have become a costly form of computer crime, the BIA is receiving a broader base of attention.

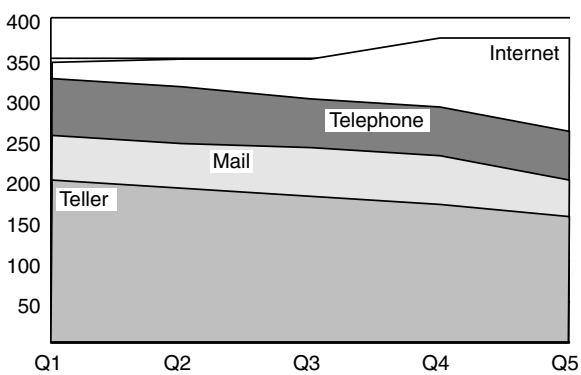
Two points must be made with respect to doing a BIA on E-commerce systems. First, as the MTD approaches zero, the potential business impact will appear absolute and infinite — much like an asymptote. Some understanding of the actual workings of the system may be indicated here. Unfortunately, because so many systems connected to the Internet host real-time activities, such as stock trading, the impact of a specific system outage may indeed be immediately devastating. This might be the case with a back-office, host-based system that previously had a more relaxed recovery requirement. Moving to a real-time Internet business model may put 7×24 requirements on legacy systems. The resulting dilemma may force decisions regarding the ability of the enterprise to run its business on certain platforms.

Second, a BIA that uses multiple revenue streams as a source of potential lost profit will need to be updated frequently as business shifts to the Internet. This is to say, for example, that a company trying to replace a telephone center with an Internet-based alternative should weight impacts to the telephone center with decreasing importance. This can be accomplished by frequently updating the BIA or by extrapolating future numbers using projections. An example for a bank transitioning to online services is shown in Exhibit 64.2.

The results of a BIA fasten perceived risks to a concrete anchor — money. As with a security architecture review, the information returned suggests a very potent tool. Obtaining resources to protect a business-critical system or process is far easier when the costs of an outage have been tallied and presented to management.

Risk Analysis

Risk analysis isolates and ranks individual risks to a specific system or process. In the past, quantitative risk analysis was time-consuming and not terribly accurate. In the area of E-commerce, risk analysis needs to be swift and decisive to be useful. In industries where marketing goals and production are expected to shift quickly to maximize profitability, risk analysis is the key to avoiding dangerous situations. It can provide the candid observations and raw ideas necessary to devise strategies to avoid and to resist threats.



A bank's transaction totals are shown in millions of dollars per quarter. Four types of transactions are added to obtain the total. As revenue streams shift from one revenue source to another, material impacts to the bank for failed systems in each of the four areas should increase or decrease in proportion to the change. Thus, the numbers used in a BIA must be extrapolated in anticipation of the changes.

EXHIBIT 64.2 Banking services over time.

The method known as facilitated risk analysis, taught by the CSI, offers a rapid, straightforward approach to ascertaining risks without getting bogged down in unnecessary details. Using this approach, a facilitator directs a small group (usually six to twelve people) through a series of questions designed to evoke the participant's impression of the threats to a particular system. Ideally, those participating will represent a diverse group of people, each having a unique view of the system. The process resembles a group interview, in that real-time peer review of each person's comments takes place. The results are a synthesized picture of the system's significant risks and a number of suggested controls for mitigating the risks.

As a process, the facilitated risk analysis is sufficiently lightweight that it could be repeated as often as required without overly taxing the affected group. Effective information security management depends on having a current, realistic assessment of risks to the enterprise's information. It also serves as a check to ensure that the mission of the information security team is in line with the customer's expectations.

Incident Response

Twenty years ago, incident response was exclusively the domain of disaster recovery and corporate (physical) security. If the incident was a massive system failure, the recovery team invoked a detailed, formal plan to recover the information asset. Had there been a reason to suspect wrongdoing, a fraud investigator would be enlisted to investigate the alleged crime.

Client/server and PC networks brought in their wake a wide range of vulnerabilities requiring proactive mechanisms to protect internal networks and hosts. As IS shops raced forward in the waves of new technologies, avoidance remained the preferred strategy — but ad hoc recovery became the new reality. In truth, most organizations make a dreadful mess of recovering from incidents.

In most shops today, incident response is the weakest tool of information defense. Incident recovery is the ability to detect and respond to an unplanned, negative incident in an organization's information systems. Most companies are woefully unprepared to respond to an incident because of their unwavering faith in avoidance. The approach of building an invincible wall around a trusted network was sold so well in the past decade that many organizations felt that spending on detection and recovery from computer crime was a frivolous waste of money. This is the same mix of technological faith and naiveté that supplied lifeboats for only half the passengers on the Titanic.

The appalling lack of incident response capability in most corporate environments is especially salient when one looks at a particularly embarrassing segment of computer crime: insider crime. The CSI/FBI computer crime survey presents an alarming picture of corporate crime that has not deviated very much over the past few years. The survey indicates that corporate insiders perpetrate approximately 70 percent of incidents reported in the survey. Even if overstated, the numbers underscore the need for increased detection and response capabilities. The criminals in these cases were found on the "friendly" side of the firewall. These threats are largely undeterred by the recent increases in funding for Internet security, and they require mechanisms for detection and response expertise to resolve.

Incident response brings an essential element to the arsenal of information security; that is, the organizational skill of having a group rapidly assess a complex situation and assemble a response. Properly managed, an incident response program can save the organization from disaster. The team needs a wide variety of experts to be successful, including legal, networking, security, and public relations experts. And the organization needs to exercise the team with frequent drills.

Conclusion

While the demand for security goods and services is experiencing boundless growth, the total cost of computer crime may be outpacing spending. This should prompt a thorough review of defensive strategies; instead, corporate IS managers seem prepared to increase funding to still higher levels in a futile attempt to build stronger perimeters. There is little reason for optimism for this program, given recent history.

The Internet now hosts live transaction business processes in almost every industry. Hitherto unthinkable exposures of technical, financial, and corporate reputations are the daily grist of the 21st-century information workers. Information risk management was once feasible with a small number of decision makers and security

technicians. Those days are gone. Risk management in an atmosphere that is so fraught with danger and constantly in flux requires clear thought and a broad base of experience. It requires that one take extraordinary measures to protect information while preparing for the failure of the same measures. It also requires wider participation with other groups in the enterprise.

It is incumbent on those who are in a position of influence to push for a more comprehensive set of defenses. Success in the Internet age depends on building a robust infrastructure that avoids negative incidents and is positioned to recover from them as well. Risk management in the 21st-century will require adequate attention to both pre-event (avoidance and assurance) measures as well as post-event (detection and recovery) measures. While the task seems daunting, success will depend on application of techniques that are already well-understood — but lamentably underutilized.

References

1. Prince, Frank, Howe, Carl D., Buss, Christian, and Smith, Stephanie, Sizing the Security Market, *The Forrester Report,* October 2000.
2. Computer Security Institute, *Issues and Trends: 2000 CSI/FBI Computer Crime and Security Survey,* Computer Security Institute, 2000.

65

Information Security in the Enterprise

Duane E. Sharp

The value of information to an organization cannot be overemphasized, particularly in today's knowledge-based economy. Information is probably *the* most valuable single asset of many organizations.

Corporate data assets are more distributed than in the past, both from a management and geographic location perspective. As well, the number of internal users requiring access to corporate data has increased and the traditionally solid IT perimeter has become much more easily accessed.

One objective of IT management is to provide high-value information services to its end users — its customers — in a timely fashion. Information is valuable in proportion to its timely availability and, in most cases, to its *secure* availability.

With a dizzying array of products and technologies available to provide secure information in various forms and in complex IT environments, the best solution is to develop a comprehensive security framework. This framework will integrate security in a cost-effective manner, subject to the needs of the entire enterprise.

Among the topics to be discussed in this chapter are the following:

- The need for security
- The requirements for implementing security
- Characteristics of an optimal security framework
- Key technology solutions to meet security requirements
- Building an effective security framework that matches technologies with requirements

The Need for Security: Accessing Corporate Data

In a number of geographic sectors of the globe, underground networks of hackers have developed and shared publicly some very sophisticated tools for intercepting and modifying data being transmitted over the Internet. These tools have even enabled successful interception of data behind the relative safety of the walls of corporate office buildings.

Some of the tools used for sniffing, hijacking, and spoofing are freely available on the Internet, a vast, loosely interconnected (and unsecured) network. Initially created as an open, accessible medium for the free exchange of information, it offers numerous opportunities to access data flowing through its global network. For example, a single e-mail message from one individual to a co-worker, buyer, vendor, client, doctor, patient, friend, or relative at a remote location may "hop" through several intermediate "nodes" before arriving at its final destination. At any point along the way, the contents of that e-mail could be visible to any number of people, including competitors, their agents, or individuals who would access the data for fraudulent purposes.

Over the past several years, the threat to organizations from hackers on the Internet has received wide publicity, as several major hacking incidents have interrupted the operations of both business and government. The fact is that although earlier surveys indicated that more than 50 percent of all intrusions occurred from *within* an organization, this trend seems to be reversing according to more recent analyses of hacking incidents.

These studies indicate that the majority of attacks are coming from *outside* the organization. It is not uncommon for such attacks to go unnoticed or unreported, so the statistics probably understate the seriousness of the threat.

In one recent analysis of 2213 Web sites of widely differing content, conducted by the Computer Security Institute, it was found that 28 percent of some commonly used sites were "highly vulnerable" to attack, 30 percent were somewhat vulnerable, and only 42 percent were considered safe. The sites surveyed were grouped into six categories: banks, credit unions, U.S. federal sites, newspapers, adult sites, and a miscellaneous group.

In another more recent study, companies reported annual increases of more than 35 percent in data or network sabotage incidents from 1997 to 1999. In this same survey, organizations reported annual increases of more than 25 percent in financial fraud perpetrated online. Insider abuse of network access increased by over 20 percent, resulting in losses of more than $8 million.

These studies point to the seriousness of the threat to organizations from financial fraud, through unauthorized access and use of corporate data flowing through the Internet and internal networks, and underline the requirement to provide a secure network environment.

Information Security Requirements

While security is a requirement at several levels in the handling of information, many security implementations focus on addressing a particular problem, as opposed to considering *all* levels. For example, most implementations have attempted to address problems such as authentication (ensuring that the users are who they say they are) or on protecting a specific resource such as the customer database. Taken by themselves, these solutions are often quite good at the job they do.

However, as with any assembly of unintegrated point products, these solutions will most likely be less than perfect, as well as being expensive to use and maintain due to their dissimilar user and administrative interfaces.

So, what should an information manager do to reduce the likelihood of significant loss from one of the enterprise's most valuable assets, without disrupting users, delaying current deliverables, and breaking the budget? The simple answer is: implement a comprehensive information security framework for the enterprise, one that stresses seamless integration with the existing IT environment; and implement it incrementally where it is most needed first.

Some of the specifics of a security framework are described later in this chapter. First, this chapter examines some of the requirements of an effective security framework and provides an overview of some of the techniques used to meet these requirements.

Primary Security Functions

The five primary functions of a good security framework include:

1. *Authentication*: to verify with confidence the identities of the users
2. *Access control*: to enable only authorized users to access appropriate resources
3. *Privacy*: to ensure confidentiality of communication among authorized parties and of data in the system
4. *Data integrity*: to ensure that communications, files, and programs are not tampered with
5. *Non-repudiation*: to provide undeniable proof that a certain user sent a certain message and to prevent the receiver from claiming that a different message was received

Functions such as virus protection are not specifically addressed because these are often combined with integrity, access control, and authentication functions.

Authentication

Authentication, the process of verifying the identity of a party or parties to an electronic communication, forces the party to produce proof of identity: something they know, something they have, or something they are. In situations where an individual is physically present to provide identification, these attributes can be provided through biometrics, a physical characteristic of the individual; for example, a fingerprint, voice print, or retinal scan. The first two categories are most commonly used because they are relatively inexpensive to implement.

In other situations where electronic communications are occurring without the facility to acquire a biometric form of identification, the easiest mechanism to implement is a simple password scheme. This mechanism forces the user to provide a known password in order to authenticate. To be effective, password authentication

requires the use of a secure channel through the network to transmit the encrypted password; otherwise, the password might be compromised by electronic eavesdroppers.

Passwords by themselves are not very secure. They are usually short and often easy to guess or observe, and they have been proven to be the weakest link in any system where a user participates in some form of digital commerce. Moreover, because users are increasingly being required to set numerous passwords for various systems, the tendency is to use a single password for all access requirements. They invariably either select from a very short list of known passwords or simply write down all passwords on a piece of paper near their computers. In either case, it is possible for someone to compromise several systems at once.

A cost-effective authentication scheme is to combine a password (something one knows) with an inexpensive smart-card token (something one has). A common example of this is the ATM (automatic teller machine) card. The ATM card is something an individual carries on their person, and the PIN (personal identification number) is something the individual knows. The combination provides improved protection (two-factor authentication) over just one or the other.

An important aspect of authentication is whether it is unilateral (sender authenticates to a server) or bilateral (user and server authenticate to each other). For example, using an ATM at a bank branch assumes that the ATM is legitimate. But can one be as confident when using an ATM sitting alone in a parking lot? There have been well-documented cases of thieves constructing extremely convincing but fraudulent ATMs in parking lots. Dozens of ATM card numbers and PINs, as well as cash, have been taken from unsuspecting customers. While these cases are admittedly rare, they do demonstrate the importance of *bilateral* authentication.

In an electronic environment, public key cryptography systems (usually referred to as PKI, for public key infrastructure), combined with digital certificates, provide a straightforward, secure mechanism for bilateral authentication. The success of public/private key systems and the trustworthiness of digital certificates lie in keeping the private key secret. If an individual's private key is stolen or accessed by an unauthorized party, then all communications to or from that person are compromised. The storage of private keys on PCs throughout an organization becomes a serious security risk. Because the private key is held on an individual's PC, the user must be authenticated on that PC to achieve security. The strongest security systems will store this information on a smart card and will require a PIN for access.

Access Control

Access control (or authorization), as the name implies, deals with ensuring that users only have access to appropriate resources (systems, directories, databases, even records) as determined by the security policy administrator. Technologies commonly used to enforce access control include trusted operating systems through the use of access control lists (ACLs), single sign-on products, and firewalls. Single sign-on products enable a user to authenticate to the environment once per session. The user will thus be authorized to access any of the appropriate resources without the need for additional authentication during that session.

Privacy

Privacy is the cornerstone of any security environment. Although the definition of privacy can vary significantly between users and owners, privacy issues are important for data with financial, personnel, or research value. Even on a corporate intranet, the privacy issue is important. However, extranet sites often face the greatest challenge in handling data because individuals and corporate data must be protected while multiple corporate entities are provided with some level of access to the data.

Depending on its sensitivity, information must be rendered indecipherable to unauthorized people, whether stored on disk or communicated over a network. Privacy can be implemented through physical isolation. In today's computing environments, however, this is generally too inefficient for most users. The ideal solution for most enterprises is to implement a decentralized cryptographic environment enabling users to maintain and exchange encrypted information.

The entire set of trust requirements for E-security (security of electronic data) builds on the foundation of encryption. There are numerous cryptographic systems available, both asymmetric and symmetric. However, asymmetric coding procedures typically have a severe disadvantage: they are computationally very expensive in comparison with symmetric procedures.

To minimize this problem, fast symmetric coding systems are usually combined with slower asymmetric ones, and a combination of public and private keys is used to decode and decrypt the message. In a secure environment, each user is assigned a user name, together with a public and private key. The public key is

published, that is, made available to all interested parties, together with the user name; the private key is only known to its key holder.

There are also efficient procedures to protect the integrity of information, by generating and verifying electronic signatures, combining asymmetric encoding with checksum algorithms, which are efficiently and easily implemented.

An interested partner can now authenticate the key holder through the capability of adding an electronic signature to data elements. However, this only ensures that the partner corresponds to that key; authentication of the partner by name requires a mechanism to guarantee that names and public keys belong together. The problem is comparable to that of a personal identity card, in that a match between the partner and the photo on the identity card does not mean that the partner's name is actually that shown on the identity card.

The idea of the identity card can be carried over to an electronic form. The corresponding "identity cards" are called certificates, which attest to the public key-name pair. It is also possible to distribute pairs of names and keys to the partners via secure channels and store them with write protection. If there are few subscribers, the names and public keys of all possible communication partners can be stored in a table in the electronic message handling system. To avoid a man-in-the-middle attack, it is necessary to ensure that the names and public keys actually belong together. In practice, this means that the pairs of names and keys must be distributed via secure channels and stored in the systems with write protection.

Data Integrity

Integrity involves the protection of data from corruption, destruction, or unauthorized changes. This requirement also extends to the configurations and basic integrity of services, applications, and networks that must be protected. Maintaining the integrity of information is critical. When information is communicated between two parties, the parties must have confidence that it has not been tampered with. Conceptually similar to checksum information, most cryptographic systems provide an efficient means for ensuring integrity.

Non-repudiation

This requirement is important for legal reasons. As more and more business, both internal and external, is conducted electronically, it becomes necessary to ensure that electronic transactions provide some form of legal proof of sender and message received when they are completed. This requirement goes hand-in-hand with the need to verify identity and control access.

IT Requirements

A good security framework must implement the functions discussed in the preceding chapter section, while at the same time supporting the following requirements:

- The varying security robustness requirements throughout the enterprise
- Integration with point security products such as security gateways and firewalls already in place in the IT infrastructure
- The heterogeneous platforms, applications, networks, networking equipment, and tools found in all IT departments
- The availability and performance requirements of the users and system administrators
- Cross-departmental, cross-geographical, and potentially inter-enterprise interaction
- The ease-of-use requirements of the users
- Flexible and cost-effective implementation under the control of the IT organization
- Stepwise implementation and deployment throughout the enterprise

The best security is transparent to the user community. When a dignitary makes a public visit, the security agents one actually sees are generally only a small fraction of the security forces deployed for that person's protection. Information security should also be largely invisible to the user community and most of the security framework should be behind the scenes.

In the open system environments commonly found in IT departments, transparency can be problematic. Consuming technology from multiple vendors enhances the value of a solution by enabling selection of best-of-breed technology and by creating competition. It is important, however, that technologies from multiple vendors fit together seamlessly, based on open standards, or this benefit is lost in the integration effort.

Ultimately, the IT organization "owns the problem." Rather than buying a number of unintegrated security "point" products such as firewalls and smart cards, it is better to implement an integrated security framework. Framework components would be designed to inter-operate seamlessly, using standard programming interfaces, with each other and with the existing IT application base. The products may still come from multiple sources, but they should plug into a framework that represents the requirements of the entire enterprise.

In today's electronic economy, organizations need to communicate transparently with other organizations, a factor that has contributed to the commercial explosion of the Internet. The world's networking environment is now laced with intranets and extranets, many of which are interwoven with the Internet, that enhance the capabilities of organizations to communicate with each other. Throughout all of these networking environments, the security of data must be maintained.

The following components form the essential framework of an integrated, comprehensive, security system, designed to protect corporate data from unauthorized access and misuse.

Encryption: The Key to Security

Encryption refers to the process of transforming messages and documents from cleartext to ciphertext using a secret code known only to the sender and the intended recipient. Decryption is the inverse process — restoring the cleartext from the ciphertext. There are a number of methods available for document encryption. These generally fall into the categories of symmetric and asymmetric cryptography.

Symmetric key structures, such as the Data Encryption Standard (DES), use a single key shared between sender and receiver. This key, when applied to the cleartext, yields the ciphertext and, when applied to the ciphertext, yields the cleartext. With symmetric keys, both the sender and receiver must share the same key. Symmetric keys tend to perform well, but the sharing protocol may not scale well in a large environment as more and more users need to communicate encrypted information to one another.

Asymmetric key structures use a public and private key-pair, a different key to encrypt and decrypt. The significance of public key encryption technology is that only the user has access to his private key; that user gives out his public key to others. Other people encrypt documents with the public key for communication to the user, and the user encrypts the documents with his private key.

There is a strict inverse mathematical relationship between public and private keys that ensures that only the user with his private key can decrypt messages encrypted with his public key. As well, with that private key, the user with his private key could have encrypted messages, while other people can decrypt with his public key. This characteristic enables the use of keys to "sign" documents digitally.

Strong Encryption: A Necessary Requirement

One security technology in widespread use today, so much so that it has become a *de facto* standard, is the RSA strong public/private key-pairs with digital certificates. Strong encryption refers to the use of encryption technology that is nearly impossible to break within an amount of time that would enable the information to be of any value. The distinction is made between strong and weak encryption, due in part to the running debate over restrictions the U.S. Government has placed on the exportability of message encryption technologies.

The technology of providing strong encryption is considered a munition and its export from the United States is, for the most part, prohibited. The export of weaker encryption is permitted with certain restrictions.

Cleartext e-mail messages and other documents sent over the Internet can be intercepted by hackers, as experience shows. If encryption is the solution, then what prevents a hacker from guessing someone's key and being able to decrypt that person's encrypted messages? In most cases, nothing more than time.

One method used by hackers is to take a sample of cleartext and corresponding encrypted text and repeatedly try random bit sequences by brute force to reconstruct the key used to encrypt the text. Therefore, all the hacker needs is a fast computer or network of computers working together and samples of the clear and encrypted texts. To protect against these brute-force attacks, cryptographic keys must be "strong."

Assessing the Strength of an Encryption System

In a strong encryption scenario, the hacker's strategy will be to use high-powered computing resources to try to crack the encryption key. The solution to this hacking process is to generate sufficiently large keys such that it will take the hacker too long to break them. It is important to remember that computing speeds are doubling roughly every 18 months. The size of a key must be large enough to prevent hacking now and in the future. Also, one does not want to have to change one's key very often.

How large should a key be? Strong encryption means encryption based on key sizes large enough to deter a brute-force attack. Thus, hackers using even a large number of powerful computers should not be able to break the key within a useful amount of time; that is, on the order of many, many years. Key sizes of 56 bits or less are considered weak. Key sizes in excess of 128 bits are considered very strong. One rule of thumb for key sizes is that keys used to protect data today should be at least 75 bits long. To protect information adequately for the next 20 years in the face of expected advances in computing power, keys in newly deployed systems should be at least 90 bits long.

Key Management

Managing keys securely is extremely important and there are a number of products from the security industry that address this issue. Most attacks by hackers will involve an attempt to compromise the key management versus the keys themselves, because a brute-force attack would require a long time to break a key with 128 or more bits.

There are several key management considerations for users. They must be able to:

- Create or obtain their own keys in a highly secure and efficient manner.
- Distribute their keys to others.
- Obtain other people's keys with confidence in the identity of the other party.

Without secure key management, a hacker could tamper with keys or impersonate a user. With public/private key-pairs, a form of "certification" is used, called digital certificates, to provide confidence in the authenticity of a user's public key.

Using Keys and Digital Certificates

Digital certificates must be secure components in the security framework. That is, it must not be possible to forge a certificate or obtain one in an unsecured fashion. Nor should it be possible to use legitimate certificates for illegitimate purposes. A secure infrastructure is necessary to protect certificates, which in turn attest to the authenticity of public keys.

One of the important functions of the certificate infrastructure is the revocation of certificates. If someone's private key is lost or stolen, people communicating with that individual must be informed. They must no longer use the public key for that individual nor accept digitally signed documents from that individual with the invalid private key. This is analogous to what happens when one loses, or someone steals, a credit card.

When keys are generated, they receive an expiration date. Keys need to expire at some point or they can be compromised due to attrition. The expiration date must be chosen carefully, however, as part of the set of security policies in force in the environment. Because other users must be made aware of the expiration, having keys expire too frequently could overload the certificate and key management infrastructure.

Digital Signatures and Certification

Encryption works to ensure the privacy of communication; but how is authentication handled? That is, how can a person, as the receiver of a document, be sure that the sender of that document really is who he says he is? And vice versa? The authentication of both parties is accomplished by a combination of a digital signature and certification mechanism.

A digital certificate from a mutually trusted third party verifies the authenticity of the individual's public key. This party is the Certificate Authority (CA), and operates in a similar manner to a notary public in the nonelectronic world. The certificate contains some standard information about the individual and holds that individual's public key. The CA digitally "signs" the individual's certificate, verifying his or her digital identity and the validity of his or her public key.

Digital signatures have legal significance for parties to an electronic transaction. Encrypting creates the signature using the private key of the signatory, information that is verifiable by both parties. The signature provides proof of the individual's identity: only the owner of the private key could have encrypted something that could be decrypted with his or her public key.

In the case of the CA signing a digital certificate, the CA uses its private key to encrypt select information stored in the digital certificate — information such as the person's name, the name of the issuing CA, the serial number and valid dates of the certificate, etc. This information is called the message authentication code (MAC). Both the sender and the receiver of the transmission have access to the certificate; thus, the MAC information is verifiable by both parties.

Anyone can verify a digital certificate by fetching the public key of the CA that signed it. When sending an encrypted document, one exchanges certificates with the other party as a separate step from the actual document exchange, to establish trust and verification.

As an example, consider a two-party exchange of private messages between Jane and Sam, and the mutual verification process. If Jane wants to send an encrypted document to Sam, she first gets Sam's digital certificate, which includes his public key signed by his CA. Jane also gets the CA's public key to verify the CA's signature, and now has confidence that the public key she has does indeed belong to Sam, because the CA's private key was used to sign it. Sam invokes a similar procedure when he receives Jane's certificate.

Of course, most of the work in this process is software controlled, transparent to the user, and, given current technology, performs with nearly imperceptible delay.

Certification Infrastructure

The previous description of a transaction between two parties describes the public key cryptography and certification process. This is a simplified example because even within the same enterprise, security policy might dictate segregating certificate management along departmental or geographic lines to provide a fine-grained level of security control and accountability.

One solution to this problem is to have a single master CA issue all certificates throughout the world. This business model has been attempted; however, the CA quickly becomes a bottleneck for organizations needing fast access to hundreds or thousands of certificates. An important fundamental in the security foundation is the capability to control one's own resources. A critical responsibility such as managing certificates should not be left to third parties.

One solution that some organizations have adopted is to establish a hierarchical certification infrastructure. A single "Top CA" within the organization is identified to certify the lower level user or departmental CAs (UCAs). The Top CA maintains certificate revocation lists (CRLs) for the organization, but would otherwise not be involved in day-to-day certificate management. The UCAs handle this stage in the certification process. Outside the organization, a top-level CA is appointed, the Policy Certificate Authority (PCA), who certifies all Top CAs and manages inter-organization CRLs to provide trust between enterprises. Finally, all PCAs in the world are certified by an Internet PCA Registration Authority, ensuring trust between certification infrastructures.

Implementing the Enterprise Security Framework

Implementing an enterprise information security environment is a major, complex task, one that will be different within each enterprise. The implementation should be done in stages, with the first stage being to establish a set of security regulations and a design for the framework, both of which need to be structured to meet both current and future needs as well as budgetary considerations. Consideration must also be given to the inter-enterprise requirements: who are the vendors, partners, and customers involved in the exchange of electronic data? In what order of priority does one wish to secure these communications?

The following chapter sections provide a reasonably comprehensive description of the tasks generally involved in implementing a secure IT environment for the enterprise.

The Security Audit

The security implementation should begin with a security audit by a qualified firm. The roles of the audit are to:

- Map out the current IT environment.

- Understand all aspects of the security mechanisms currently in place — physical security as well as software and hardware solutions.
- Obtain a detailed and confidential analysis of security breaches that have or may have already occurred.
- Provide an assessment of the current security mechanisms with specific emphasis on deficiencies as compared with other organizations.
- Provide an independent assessment as to the root causes of previous incidents.
- Provide recommendations for improvements to the security infrastructure.

Business Analysis and Development of Security Policy

The next step is to conduct an in-depth security analysis along with a business analysis based on the audit findings. Then a set of security policies to meet the needs of the enterprise can be developed. The security framework will be adapted to adhere to these policies. This process encompasses the following multi-stage process:

1. *Establish the organizational relationship between security personnel and the IT organization*. Is there a separate security organization; and if so, how are its policies implemented in the IT organization? What is the security budget, and how are resources shared?
2. *Define the security and IT distribution models*. Does the headquarters organization set policy and implement at all sites, or do remote sites have authority and accountability for their own IT environments?
3. *Understand the security goals at a business level.* Determine the key resources requiring protection and from whom. Who are the typical users of these resources, and what do they do with them? What auditing mechanisms are in place, and what are the physical isolation versus hardware/software considerations?
4. *Assess the IT-vendor business issues*: dealing with a single vendor versus several, buying product and service from different vendors, experience with training and support, etc.
5. *List the applications, data files, and server and client systems* that need to be enhanced with security.
6. *Plan the current, near-term, and longer-term IT environment*: addressing issues such as major data flows between business components, platform, hardware, network topology, third-party electronic interaction requirements, space planning, and physical security.
7. *Propose a high-level security paradigm* for defining and controlling access to corporate data, for example, access control server with firewall, smart-card tokens versus single-factor authentication, centralized versus peer-to-peer certification, etc.
8. *Develop a high-level set of security policies for the enterprise*, including site security personnel, access control, certification, and the interaction of the security infrastructure with other enterprise resources.
9. *Analyze and document key dependencies* within the security framework and between the framework and the applications.

Project Planning

Once high-level security policies and a framework have been established, the project plan will have a basic structure. The next stage is to break down the framework into tasks that can be sized, cost-justified, and scheduled for delivery and deployment.

There is no single, definable approach to the planning phase — it will consume resources from potentially many different groups. There are various trade-offs that can be made in terms of an implementation model, such as cost based or complexity based. A project manager should be identified at this stage. This individual should have a broad technical background in IT development projects and a fairly deep knowledge of security implementations.

Selecting an Implementation Model

It is difficult to advise on the selection of an implementation model because so much depends on other work going on in the IT organization. For example, if the organization is about to embark on a major software program, implementing a thorough security program would be a prudent approach because all systems may be open for modification and enhancement with security components. Conversely, if resources are already over-allocated, few large security-related programs can be implemented.

A recommended guideline for rolling out a security implementation is to proceed in stages from a localized client/server (group) level, to a site-wide deployment, to the full enterprise, and finally to an inter-enterprise configuration. At each stage, the issues can be tackled in a similar manner. For example, it might be best to start by installing a basic authentication and access control implementation that provides basic security for individual devices and the network perimeter.

The next stage would be an enhanced level of authentication and access control with centralized services at the network level, as well as a cryptographic environment for privacy and integrity. Finally, truly robust security can be provided at the network perimeter and inside the network with access control, strong cryptography, certification, token management, and non-repudiation.

It is good design practice to start the design form with legacy systems because:

- These systems tend to transcend the many organizational changes common to business today.
- They are often at the core of the business.
- Modifying these applications may be sufficiently onerous that it is considered a better strategy to "surround" the system with security, versus adding it in.

There are two basic approaches to the development of a security framework. One approach is to begin with the servers and work outward. This approach has the advantage of integrating security into the enterprise at the primary data source. However, for a decentralized IT organization, a better approach might be to build up the levels of security from the clients inward.

One technique that can be used for the client-inward approach is to incorporate smart-card readers into all client PCs up front and require local authentication via the smart card. This functionality could later be expanded to provide single-sign-on access to the network and other features. The disadvantage of this approach is that the client side is difficult to measure, in terms of numbers, because it is usually a changing number, as clients may be added, removed, or receive software upgrades fairly frequently in some organizations. This approach may not catch strategically important server data, which needs protection.

Skills Assessment

Once the implementation model is chosen, a skills inventory needs to be developed. This inventory of skills will be used to determine the appropriate staff resources available and the training requirements. The use of open, standards-based security tools is essential in minimizing the need for extensive training in a proprietary environment.

It is advisable to prepare a high-level workflow diagram to identify the affected organizations requiring representation on the project teams. All affected organizations should be identified and staffing resources within each organization nominated. If physical equipment isolation requiring space planning is required, for example, building operations may need to become involved.

At this stage, project teams can be formed with representation from IT and other organizations. To ensure that all departments have input to the security framework design, end-user departments should have representatives on project teams.

Sizing and Resource Planning

The next major stage in the design process is to prepare project sizing estimates and a resource plan. This is the point at which the security framework project should dovetail with any other planned IT projects. A complete IT budget and staffing plan review may be necessary. In some cases, project priorities will need to be adjusted. The security implementation sub-tasks should be prioritized in groups, where dependencies were identified in the framework, to ensure that the priorities are consistent.

As for any major IT project, price/performance trade-offs within a sub-task need to be analyzed. In the analysis of business requirements and the development of security policy performed previously, the determination might have been made that the enterprise had numerous physically isolated resources, relative to the threat of attack. In this situation, a hardware/software technology solution might better optimize resources across the enterprise, while still providing more than adequate security.

Local authentication processes can range from simply verifying the user's network address to sophisticated biometrics with varying degrees of robustness and cost.

It is also important to evaluate price-performance trade-offs for hardware/software combinations. It might, for example, be more cost-effective to implement a Windows NT®-based firewall and accept somewhat less performance scaling than to use a more powerful UNIX product. These decisions will be influenced by the technical skill sets available within the IT organization.

Selecting the Technology Vendor

Once high-level security policies and the project plan have been established, it is time to approach the security product vendor community to assess product offerings. A system integrator may also be required to supplement the local IT resources and ensure the proper interface of all components. RFPs and RFIs for security products can be quite extensive and should include the following criteria, which are the most important characteristics required from a security product vendor:

- *Performance, scalability.* How much delay is incurred in implementing security, and how does the solution scale as users and resources are added to the system?
- *Robustness.* How secure is the solution against a complex attack?
- *Completeness.* How broad and deep is the solution, and for what type of environment is the solution best suited?
- *Interoperability.* How well does the solution integrate into the proposed environment?
- *Support, availability.* How available is the solution, and what are the support and maintenance characteristics?

In support of these five basic and fundamental characteristics, the following set of extensive questions on vendor products should form part of the vendor evaluation process, along with any other concerns involving a specific IT environment.

1. Which of the five primary security functions — access control, authentication, privacy, integrity, and non-repudiation — are provided by the products, and how do they work?
2. Describe the types of attacks the products are designed to thwart.
3. What is the level of network granularity (client only, local client/server, site-wide, full enterprise, inter-enterprise) for which the products are best suited?
4. What type, if any, of encryption do the products use?
5. Does the encryption technology ship from the United States? If so, when messages travel between countries, is encryption "weakened," and down to what level?
6. Do the products use certification and signing? Describe the architecture.
7. Who conducted the security audit? Present the results.
8. To what standards do the products conform, and where have proprietary extensions been added?
9. With which third-party security offerings do the products inter-operate "out of the box"?
10. How precisely do the products interface with one's existing security products, such as security gateways and network managers? Where are modifications required?
11. On which of the proposed platforms, applications, and tools will the products work without modification?
12. Does the product function identically on all supported platforms, or will separate support and training be required?
13. What are the availability levels of the products (e.g., routine maintenance required, periodic maintenance required (7×24)?
14. How are the products managed, and can they be easily integrated with the rest of the proposed system and network management infrastructure?
15. Is the product support provided by the vendor, or is it outsourced? Will the vendor support mission-critical environments around the clock?
16. Do the products support cross-departmental, cross-geographical, and potentially inter-enterprise interaction? How exactly? Does the vendor have reference sites available with this functionality running? How easy are the products to use based on references?

17. Does one need to deploy the products all at once, or can they be phased in? That is, do the products run in a hybrid environment, enabling communication, for example, between secure and unsecured users?
18. Provide quantitative information on the scalability of the solution as users and secured resources are added.

Implementation and Testing

The project implementation will always reveal issues not identified in the planning stages. For this reason, it is important to develop a well-thought-out framework for the implementation, and to choose manageable, well-defined tasks for the project plan. As the implementation progresses from design to development, testing, and eventually deployment, it is important that new requirements not be introduced into the process, potentially resulting in major delays. New requirements should be collected for a revision to the project that would go through the same methodology.

When the project has exited the testing phase, to ensure a smooth transition, a localized pilot test that does not interfere with mission-critical systems should be performed. The pilot should match as closely as possible the live configuration in a controlled environment and should last as long as necessary to prove the technology, as well as the processes and practices used in developing the security framework.

Conclusion

The major stages of an IT security implementation — audit, requirements analysis, framework construction, project planning, and implementation — have been described in this chapter, with a focus on some of the approaches that can be used to implement a security framework, as well as some of the key issues to consider.

This chapter on enterprise security provided an overview of the security requirements needed to protect corporate information from unwarranted access, and a detailed process for designing and implementing an enterprisewide security framework. Some of the solutions available to implement this security framework were described, along with recommendations for the appropriate processes and guidelines to follow for a successful, effective implementation.

The security of information in the enterprise must be viewed from five perspectives:

1. Authentication
2. Access control
3. Privacy
4. Integrity
5. Non-repudiation

An effective enterprise security framework will integrate these functions with the existing IT environment, and the final system will have the following characteristics:

- Flexible enough to provide IT management with the capability to control the level of security
- Minimal disruption to users
- Cost-effective to implement
- Usable in evolving enterprise network topologies, spanning organizational and geographic boundaries; the security framework must also provide interoperability with organizations outside the enterprise

66

Managing Enterprise Security Information

Matunda Nyanchama, Ph.D., CISSP and Anna Wilson, CISSP, CISA

Today's business and computing environments have blurred traditional boundaries between what is considered trusted and untrusted. As a result, organizations are taking measures to protect their information assets. Information from various sources in an organization's network is key to managing security. Such information comes from a number of security devices (intrusion detection systems and firewalls), operating systems, and network devices such as switches and routers.

In general, each of these devices performs a function that contributes to the overall enterprise needs and hence its security posture. Moreover, each of these technologies has a responsibility in the overall security management of the computing environment. Collectively, these devices produce a large amount of information.

The challenge before us is to make sense of all this information and to mange it in a way that is useful in protecting the computing environment, and in a manner that will benefit the entire enterprise. To achieve this, one must first understand the technology one is dealing with, how it collects and interprets information, and at what point one needs to intervene in the overall process. Having this understanding will allow one to set out the best strategy in one's approach to the management of enterprise security information.

This chapter discusses issues pertaining to challenges of managing security information for purposes of improving an organization's security posture through aggregation, analysis, and correlation.

This chapter discusses the sources of information that are useful for security management and the nature of the information they produce. Among technologies discussed are intrusion detection systems (IDSs), firewalls, routers, switches, and operating systems. Also explored are their primary function in security management, the manner in which they collect information, and how this information can be analyzed, collectively, to offer an enterprisewide security view. Ways of collecting this information and how it informs the security management process are also discussed.

Having this appreciation, one can look into the various strategies available in the overall management of this security information. In addition, there is a quick overview of the issues of security management and the challenges for managing this information in a manner that raises security effectiveness. This knowledge is intended to empower security information security practitioners in planning the most effective way in which to blend technology and man, ongoing efforts to keep business environments secure.

The material in this chapter should be read in conjunction with suggested references. In discussing various network and security technologies, the authors do so with a view to understanding the nature of information they produce and how this information can be used to ensure enterprise security. Specific technologies are not discussed in sufficient detail to make this chapter a stand-alone technical reference with respect to the said technologies. However, the chapter does include discussions of the following:

- The need for and sources of enterprise security information, including:
  - — IDSs
  - — Firewalls
  - — System logs
  - — Switches and routers

Some strategies for enterprise security management are discussed; these include approaches to collection and analysis strategies. The section also touches on the challenges of associating vulnerability data to business risk. The final section offers a summary and pointers to future challenges for managing security information.

Sources of Enterprise Security Information

This chapter section focuses on the need for security information, sources of such information, and how this information helps with the management of enterprise security. Understanding the need to collect security information is important because it is this need that determines the nature of desirable information, the means of collection, and the necessary manipulation that helps in security management.

The Need for Security Information

The past decade has seen tremendous growth of issues of information security, including technology, skilled professionals, and security-related information. This growth, spurred by the central role computers and networking continue to play in all aspects of daily endeavors and commerce, has resulted in reaches beyond physical boundaries. Networking has extended this reach into areas outside organizations' and individuals' immediate control. Further, it has contributed to the development of today's commercially available security technologies, in an effort to assert control over one's "territory."

On the other hand, networking has resulted in complex systems composed of differing network devices. The ensuing systems produce information used in the ongoing management of the networks and computing resources. This information must be analyzed to better understand the environment in which it is produced.

On the whole, security information forms a component of the total information produced in entire systems within organizations. Such security information is important for making security-related decisions, without which the situation would be tantamount to getting behind the steering wheel of a car, blindfolded, and hoping for the best. The value of information from security systems and devices is useful for making informed, cost-effective choices about how best to protect and manage computing environments.

Be it an audit log, firewall log, or intrusion information, such information is useful in many different ways. It can be used for performing system audits to determine the nature of activity in the system. It can also be used for the diagnosis required from time to time, especially in cases of a security incident; and is also useful for forensic analysis, which forms a core component of incident resolution. In general, security information is useful to determine an enterprise's security.

Sources of Security Information

To be effective, information security management requires varied pieces of information across an enterprise. This information that originates from various sources contributes to an enterprise's information security jigsaw puzzle. Each piece of information is useful for what it reveals. Aggregation of these information pieces contributes to a better understanding of the overall security posture.

Perhaps the most familiar source of security information is operating system logs. Operating system logs have been a common feature of computers for a long time, even before security administration became entrenched as it is today. During this time, system administrators have used system logs to manage computing environments, specifically pertaining to determining who was doing what, where, and when, in a fairly detailed manner.

With the growth of inter-networking, the need to connect internal networks to other external networks has continued to grow. Invariably, connecting to untrusted networks creates a need to control communication between internal and external networks. This is the role filled by the use of firewalls as they act as gateways between internal and external networks. Firewalls are a common feature in today's networking. And just as operating systems produce logs pertaining to system activity, firewalls track activity at the gateway.

With operating system logs recording activity on systems and the firewalls controlling and logging activity through the gateway, one might assume that a network would be secure against external attacks. Right? Not exactly! This is because those who venture on the dark side are also pretty clever. They have a knack of getting around established defenses and backdoors, exploiting weaknesses in communication protocols, applications, and operating systems.

This creates a need for intrusion monitoring to supplement the firewalls defenses. Intrusion detection systems (IDSs) provide information about flagged events on systems and networks. IDSs track suspicious

network activity, which may indicate attack attempts, probing, or successful intrusion. Information provided by an IDS may reveal missed or unforeseen weaknesses or holes in internal systems and the gateway. This affords the opportunity to harden or close the exposures caused by the security weakness and holes.

Systems, firewalls, and IDSs play different but complementary roles in enforcing security. Each of these is responsible for a specified role in the computing infrastructure. In practice, however, their boundaries may not be as distinct as may be suggested here. And given the different roles they play, there exist differences in the type of information they produce, the way it is collected, and how it is analyzed and interpreted.

Security information also comes from routers and switches in a network. Routers and switches play a critical role in networking and are critical to the availability of infrastructure segments.

When combined, this information from diverse sources provides a holistic picture of enterprise security. The resulting aggregation benefits from the power of correlation and may yield useful patterns and trends in a manner that informs the security management process. For example, such information can improve the management of security incidents and ensure that lessons learned will help improve future management of similar incidents, helping shift the management of incidents from a reactive to proactive mode.

Security information, if used and managed appropriately, can offer the prescription for the total security "health"[1] of our computing environments.

Intrusion Detection Systems (IDSs)

This chapter section focuses on IDSs, what they are, and their role in the management of enterprise security. IDSs can be seen as devices that monitor the pulse of the enterprise health. They depend on anomalies and known attacks to raise alarms about potential attacks and intrusions. They are limited to the extent that they can recognize anomalies or associate an activity to a known attack based on activity signature.

Introduction

IDSs play an important role in the monitoring and enforcement of security in an enterprise. They are usually deployed at vantage points where they detect activity and take action as desired, including logging the associated activity, raising an alarm or a pager, or sending an e-mail message to specified users for attention.

IDSs detect flagged activity that is deemed suspicious for which they generate specified action. Whether monitoring traffic on a network or watching for suspicious changes on a specific host, IDSs form part of the "active security" components in an organization.

IDSs continue to evolve as they face ever-growing security challenges. These challenges include keeping up with hacker exploits that evade IDS detection. IDSs must also contend with denial-of-service attacks intended to bring them down.

In general, intrusion detection technology is relatively young. Although there are minor differences among security professionals as to what constitutes an IDS, there is substantial agreement on the role that IDSs play in enterprise security management. Further, there is concurrence on the need for analysis of IDS information as an aid to security management. In general, IDSs are a major source of information which, when analyzed and acted upon, helps improve enterprise security.

An ideal IDS has several automated components that define its functionality, including:

- Providing information about events on a computer system or network
- Analyzing the information in a manner that aids the security process
- Logging and storing security-sensitive event information for future use, specifically for making improvements
- Acting on that information in a manner that improves security
- Performing all of the above in a flawless and timely manner

The above list is an ultimate IDS dream for all security professionals. Whether such a system exists is a matter for another discussion.

There are two key approaches to IDS monitoring. These include knowledge-based and anomaly detection IDSs. Moreover, there are two key strategies for IDS deployment; that is, on the network or on a host. These are discussed individually, along with an overview of incident response, given the close relationship between IDS and incident response.

Knowledge-Based Intrusion Detection Systems

Misuse detection-based IDSs, also called knowledge-based IDSs, are the most widely used today. Such IDSs contain accumulated knowledge about known attacks and vulnerabilities based on signatures. Using this knowledge base of attack signatures of exploits, the IDS matches patterns of events to the attack signatures. When an attack attempt is detected, the IDS may trigger an alarm, log the event, raise a pager, or send an e-mail message.

Knowledge-based IDSs are easy to implement and manage due to their simplicity. They are very effective at quickly and reliably detecting attacks. With continued tuning and update of signatures, it is possible to lower the false alarm rate. In the process, this enables security professionals to respond to incidents very effectively, regardless of their level of expertise.

There is a downside to misusing detection-based IDSs; they are most effective when the information in their knowledge base remains current. The predefined rules or attack signatures must be continuously updated. Moreover, there is usually a time lag between the time an exploit is publicized and when an associated attack signature is available. This leaves a window of opportunity for a new, undetectable attack. Such IDSs can be seen to be blind to potentially many attacks that they do not "know" about, especially where there is a substantial time lag in the update of the IDS' knowledge base.

Anomaly Detection (Behavior-Based IDS)

Anomaly detection, or behavior-based, IDSs operate on the premise of identifying abnormal or unusual behavior. Anomalies on a host or network stand apart from what is considered normal or legitimate activity. These differences are used to identify what could be an attack.

To determine what an anomaly is, systems develop profiles representing normal user activity on hosts or networks over a period of time. The system collects event data and uses various metrics to determine if an activity being monitored is a deviation from what is considered "normal behavior."

To determine such normal behavior, some, all, or a combination of the following techniques are used:

- Rules
- Statistical measurements
- Thresholds

However, these systems are subject to false alarms because patterns of activity considered to be normal behavior can change and vary dramatically.

The key advantage of behavior-based IDSs is that they are able to detect new attack forms without previous specific knowledge of the attacks. Further, there is the possibility of using the information produced by anomaly detection to define attack patterns for use in knowledge-based IDSs.

Host-Based Intrusion Detection Systems

Host-based IDSs are installed on specific hosts on which they perform monitoring. A host-based IDS can be seen as system specific. It uses the system's audit, system, and application logs for IDS information. Using the system's various logs lends to the quality of the information available to the IDS. Given that it is dealing with a specific operating system, the accuracy of the associated information will be substantially high because the operating system retains a good sense of activity on the host on which it is installed.

A host-based IDS responds when flagged events happen on the host. These events could pertain to file changes, privilege escalation, or any such activity deemed security sensitive. This makes a host-based IDS very effective in detecting integrity attacks. Using an operating system's audit trails, a detected inconsistency in a process could be an indication of a Trojan horse or some other similar attack.

Additional advantages of a host-based system include the ability to detect attacks that go undetected by network-based systems. Depending on where information sources are generated, host-based systems can operate in environments in which network traffic is encrypted. Where switching technology is utilized on a network, host-based systems remain unaffected.

Host-based IDSs suffer some drawbacks. Given that they are usually designed for specific systems and applications, host-based systems may not be very portable. Moreover, an IDS that supports one platform may not support another. In a complex environment in which there are varied systems and applications, there may be a temptation to install a different host-based IDS on each of the systems. This would result in a complex

environment with many different IDSs, which, in turn, presents a challenge in the monitoring and management of all the resulting information from the diverse systems.

Despite these disadvantages, a host-based IDS remains an important tool, as the resources on those hosts are the targets for many attackers — which leads to yet another disadvantage. Suppose a specific host on a network running an IDS is under attack. What will be the first target on that host?

Network-Based Intrusion Detection Systems

A network-based IDS monitors network traffic in the network segment on which it is installed. It functions by analyzing every packet to detect any anomalies or performing pattern matching against captured packets based on known attack signatures. Based on the information in the packet, the IDS attempts to identify anything that may be considered hostile or patterns that match what may have been defined as hostile activity.

Packet analysis presents a challenge in that the information may no longer be as revealing as in the host-based system. Indeed, a substantial degree of inference is required to determine whether observed patterns or detected signatures constitute hostile activity. This is because one can determine the physical source of a packet but one may not know who is behind it.

Network-based IDSs have some key advantages, including their nonintrusive stealth nature. As well, unlike host-based IDSs that may impact hosts on which they reside, network-based IDS performance does not impact systems. As well, network-based IDS packet analysis is beneficial over the host-based system when under some type of fragmentation attack.

One major disadvantage of network-based IDSs is the inability to scale well with respect to network traffic. The ability to inspect every packet under high traffic conditions offers a challenge to IDSs. The result is packet loss. Where such packet loss is substantial, there may be less IDS information to manage but that information may be critical to the desired security.

After examining the pros and cons of the various IDS technologies, one can clearly see that the most effective use of an IDS would be to use some combination of all.

IDS Selection and Deployment

The selection and deployment of an IDS must take a number of factors into consideration, including the:

- Purpose for which it is intended: host- or network-based intrusion detection
- Ability to scale up to high volumes of traffic if it is a network-based IDS
- Scope of attack signatures, where it is knowledge-based, or the ability to perform accurate anomaly detection

Other factors that determine deployment include the volume of information being analyzed, the degree of analysis desired, and the significance of the intrusions or attacks one wants to monitor.

The physical location of an IDS is determined by the type of activity intended to be monitored. Placing a network-based IDS outside the security perimeter (e.g., outside the firewall) will monitor for attacks targeted from outside, as well as attacks launched from inside but targeted outside the perimeter. On the other hand, placing an IDS inside the security perimeter will monitor for successful intrusions. Placing the IDS on either side of the firewall will effectively monitor the firewall rules (policy), because it will offer the difference between activity outside the firewalls and successful intrusions.

In deploying an IDS, one must select the mode of operation, which can be either real-time (in which IDS information is passed in real-time for analysis) or interval-based (also known as batch) mode (in which information is sent in intervals for offline analysis). Real-time analysis implies immediate action from the IDS due to the constant flow of information from its various sources. Interval-based or offline analysis refers to the storage of intrusion-related information for future analysis.

The choice of one of these methods over the other depends on the need for the IDS information. Where immediate action is desirable, real-time mode is used; where analysis can wait, batch-mode collection of information would be advantageous.

Incidence Response

IDSs are useful for detecting suspicious activity. As discussed in the previous chapter section, IDSs log and transmit intrusion-related information. Security management requires that this information be transformed

into suitable format for storage and analysis. Potentially, anything identified by an IDS — whether it is an attack, intrusion, or even a false alarm — represents an incident requiring analysis and action. The materiality of the incident depends on the threat posed by the incident.

In cases where an intrusion is thought to have occurred, the security organization must respond quickly and act urgently to contain the intrusion, limit the damage caused by the intrusion, repair any damage, and restore the system to full function.

Once things have calmed down, it is important to perform a root cause analysis to determine the nature of the attack and then use this information to improve defenses against future attack. Without applying the "lessons learned" into the process of security enforcement, an organization risks future attack and exploitation.

In general, the incident response process should take a system approach based on detection, response, repair, and prevent. The IDS performs detection, raising the alert to an incident. Human intervention must respond to the incident, perform the repair, and ensure that the lessons learned help improve security.

IDSs can be configured to help manage incidents better based on how they are configured to respond to attacks and intrusions. These responses can be passive (e.g., logging) or active (e.g., generating a page to the security administrator).

Active responses involve automated actions based on the type of intrusion detected. In some cases, IDSs can be configured to attempt to stop an attack, for example, through actively killing the offensive packets. It can also involve terminating the attacker's connection by reconfiguring routers and firewalls to block ports, services, or protocols being utilized by the attacker. Further, network traffic can also be blocked based on the source or destination address and, if necessary, all connections through a particular interface.

The least offensive approach for an active response is to raise the attention of a security administrator, who will then review the logged information about the attack. The analysis will show the nature of the attack and the associated response necessary. Based on the outcome of this analysis, the sensitivity of the IDS can be adjusted to reflect the need for response.

This can be accomplished by increasing the sensitivity of the system to collect a broader scope of information, assisting in the diagnosis of whether an attack actually occurred. Collection of this information will also support further investigation into an attack and provide evidence for legal purposes, if necessary.

There are other approaches to responding to perceived attacks, including fighting back. This involves actively attempting to gain information about the attacker or launching an attack against them. Despite being appealing to some, this type of approach should be used only to the extent of gathering information about the attacker. Actively launching an attack against a perceived attack has a number of potential perils. For example, suppose the source IP has been spoofed, and the last hop of the attack has been just a launch pad rather than originating the attack. Moreover, this has legal implications. As such, professionals should be very clear about their legal boundaries and take care not to cross them.

Most of today's commercially available IDSs depend on the passive responses by logging attack information and raising alarms. Invariably, this requires human intervention to respond to the information provided by the IDS. These come in the form of alarms, notifications, and SNMP traps. An alarm or notification is triggered when an attack is detected and can take the form of a pop-up window or an on-screen alert, e-mail notification, or an alert sent to a cellular phone or pager. To some degree, some commercial IDSs give users the options to do "active kills" of suspicious traffic.

To send alarm information across the network, many systems use SNMP traps and messages to send alarms to a network management system. One is beginning to see security-focused management systems coming onto the market that consolidate security events and manage them through a single console. The benefit of such a system is its holistic nature, allowing the entire network infrastructure to play a role in the response to an attack. Many of the recognized network management systems have incorporated security-specific modules into their systems to meet this demand.

Is IDS Technology Sufficient for Security?

Given an understanding of the role of the IDS and the role of incidence response in security management, IDS technology can only go so far; that is, cause alerts, log security-sensitive events, and to a limited degree, perform active kills of offensive traffic. The information generated by IDSs across an enterprise must then be used to make informed decisions intended for security improvements.

Even if we have a state-of-the-art IDS deployed, the analysis of incidents by experts provides critical data for the enhancement of the response and management process. Once an alerted incident has been identified

and determined to be, in fact, a critical incident, the response team will react quickly to ensure the event is contained and the network and systems are protected from any further possible damage. At this point, the role of forensics comes into play. The forensics experts will conduct a detailed analysis to establish the cause and effect of the incident, and the resulting data from this forensic analysis will provide the information necessary to find a solution. Taking this information and organizing it into various categories such as hostile attacks, denial-of-service, or misuse of IT resources, to name a few, allows for statistical reporting to improve the handling and response of future incidents.

Finally, one needs to use this information to address any weaknesses that may have been identified during the analysis. These can range from technical vulnerabilities or limitations on the systems and network, to administrative controls such as policies and procedures. One must effectively inoculate against possible future incidents to prevent them from occurring again. Case in point: how many security professionals have to repeatedly deal with the effects of the same virus being released as a variant, simply because the lessons from a previous infection were not learned? These post-mortem activities will serve to improve one's security posture, contribute to lessons learned, and heighten security awareness.

Other IDS management issues include ensuring that the IDSs are updated and constantly tuned to catch the most recent attacks and also filter false alarms. Like all systems, IDSs require constant maintenance to ensure the usefulness of the information they collect.

Firewalls: Types and Role in Security Enforcement

This chapter section reviews firewalls and their role in protecting information, including the different firewall types and advantages and limitations in security management.

Introduction

A firewall is a device that provides protection between different network zones by regulating access between the zones. Typically, a firewall filters specific services or applications based on specified rules, and provides protection based on this controlled access between network segments.

Firewalls have major advantages, including:

- The ability to consolidate security via a common access point; where there is no firewall, security is solely the function of the specific hosts or network devices. This consolidation allows for centralized access management to protected segments.
- Being a single access point, the firewall provides a point for logging network traffic. Firewall logs are useful in many ways as log reviews can offer major insights into the nature of traffic transiting at the firewall. Such traffic could be intrusion related, and its analysis helps to understand the nature of associated security threats.
- The capability to hide the nature of the internal network behind the firewall, which is a major boon to privacy.
- The ability to offer services behind a firewall without the threat of external exploitation.[2]

While providing a core security function, a firewall cannot guarantee security for the organization. Effective firewall security depends on how it is administered, including associated processes and procedures for its management. Further, there must be trained personnel to ensure proper configuration and administration of the firewall.

Although overall, firewalls help to enhance organization security, they have some disadvantages. These include hampered network access for some services and hosts, and being a potential single point of failure.[3]

There are two major approaches to firewall configuration, namely:

1. Permit all (e.g., packets or services) except those specified as denied
2. Deny all (packets or services) except those specified as allowed

The "permit all" policy negates the desired restrictive need for controlled access. Typically, most firewalls implement the policy of "deny all except those specified as allowed."

Firewall Types

Packet Filters

Packet filtering firewalls function at the IP layer and examine packet types, letting through only those packets allowed by the security policy while dropping everything else. Packet filtering can be filtering based on packet type, source and destination IP address, or source and destination TCP/UDP ports. Typically, packet filtering is implemented with routers.

The major advantage of packet filters is that they provide security at a relatively inexpensive price as well as high-level performance. As well, their use remains transparent to users.

Packet filters have disadvantages, however. These include:

- They are more difficult to configure, and it is more difficult to verify configurations. The high potential for misconfiguration increases the risk of security holes.
- They neither support user-level authentication nor access based on the time of day.
- They have only limited auditing capability and have no ability to hide the private network from the outside world.
- They are susceptible to attacks targeted at protocols higher than the network layer.

Application Gateways

Application gateways function at the application layer and examine traffic in more detail than packet filters. They allow through only those services for which there is a specified proxy. In turn, proxy services are configured to ensure that only trusted services are allowed through the firewall. New services must have their proxies defined before being allowed through.

In general, application gateways are more secure than packet filtering firewalls.

The key advantages of firewalls based on application gateways are:

- They provide effective information hiding because the internal network is not "visible" from the outside. In effect, application gateways have the ability to hide the internal network architecture.
- They allow authentication, logging, and can help centralize internal mail delivery.
- Their auditing capability allows tracking of information such as source and destination addresses, size of information transferred, start and end times, as well as user identification.
- It is also possible to refine the filtering on some commands within a service. For example, the FTP application gateway has the ability to filter **put** and **get** commands.

The downside of application gateways is that the client/server connection is a two-stage process. Their functioning is not transparent to the user. Moreover, because of the extent of traffic inspection employed, application gateways are usually slower than packet filters.

Firewall Management Issues

Good security practices require that firewall activity be logged. If all traffic into and out of the secured network passes through the firewall, log information can offer substantial insight into the nature of traffic, usage patterns, and sources and destinations for different types of network traffic. Analysis of log information can provide valuable statistics — not only for security planning, but also with respect to network usage.

Where desirable, a firewall can provide a degree of intrusion detection functionality. When properly configured with appropriate alarms, the firewall can be a good source of information about whether the firewall and network are being probed or attacked. This plays a complementary role when used in conjunction with an IDS.

Network usage statistics and evidence of probing can be used for several purposes. Of primary importance is the analysis of whether or not the firewall can withstand external attacks, and determining whether or not the controls on the firewall offer robust protection. Network usage statistics are a key input into network requirements studies and risk analysis activities.

More recent techniques for study of attacks and intrusions use "honey pots" for studying traffic patterns, potential attacks, and the nature of these attacks on enterprise. Here, a honey pot with known vulnerabilities is deployed to capture intrusion attempts, their nature, their success, and the source of the attacks. Further

analysis of the honey pot traffic can help determine the attackers motives and the rate of success of specific types of attacks.[4]

Is Firewall Security Sufficient?

There are many organizations that install a firewall, configure it, and move on, feeling confident that their information is secure. In real life, a firewall is like that giant front door through which most intruders are likely to come should they find holes they can exploit. In reality, there are many ways an intruder can evade or exploit the firewall to gain access to the internal network. This includes exploitation of protocol or application-specific weaknesses or circumventing the firewall where there are alternate routes for traffic into and out of the internal network.[5]

In reality, the issues that guarantee maximum security pertain to processes, people, and technology. The technology must be right; there must be trained people to manage the technology; and processes must be in place for managing the security and the people enforcing security.

Key processes include:

- Applying updates or upgrades of the software
- Acquiring and applying the patches
- Properly configuring the firewall to include collection of logs and log information
- Reviewing log information for security-sensitive issues
- Correlating the log information with information from other security devices in the network
- Determining the findings from the security information and acting on the findings
- Repeating the cycle

Operating System Logs

This chapter section reviews system logs, what they are, and why they are required; different means of collecting log information; strategies for managing system logs; and the challenges of managing log information and its impact on system security.

Introduction

Operating system logs are an important and very useful tool in the gathering and analysis of information about systems. They serve to provide valuable detailed information regarding system activity. Logs are divided into several categories responsible for recording information about specific activities, including user, security, and system, and application related events. They can support ongoing operations and provide a trail of the activity on a system, which can then be used to determine if the system has been compromised and, in the event of criminal activity, provide important evidence in a court of law.

Types of Logs, Their Uses, and Their Benefits

The auditing of operating systems logs information about system activity, application activity, and user activity. They may function under different names depending on the operating system but each is responsible for recording activity in its category. A system can log activity in two ways: event oriented or recording every keystroke on the system (keystroke monitoring).

The event-oriented log contains information related to activities of the system, an application, or user, telling us about the event, when it occurred, the user ID associated with the event, what program was used to initiate it, and the end result.

Keystroke monitoring is viewed as a special type of system logging, and there can be legal issues surrounding it that must be understood prior to its use. Using this form of auditing, a user's keystrokes are recorded as they are entered, and sometimes the computer's response to those keystrokes are also recorded. This type of system logging can be very useful for system administrators for the repair of damage that may have been caused by an intruder.

System log information is used for monitoring system performance. Activities such as drivers loading, processes and services starting, and throughput can provide valuable information to the system administrator for fine-tuning the system. In addition, these logs can capture information about access to the system and what programs were invoked.

Events related to user activity establish individual accountability and will record both successful and unsuccessful authentication attempts. These logs will also contain information about commands invoked by a user and what resources, such as files, were accessed. If additional granularity is required, the detailed activity within an application can be recorded, such as what files were read or modified. The application logs can also be used to determine if there are any defects within the application and whether any application-specific security rules were violated.

The benefits of these audit logs are numerous. By recording user-related events, not only does one establish individual accountability but, in addition, users may be less likely to venture into forbidden areas if they are aware their activities are being recorded. The system logs can also work with other security mechanisms such as an access control mechanism or an intrusion detection system to further analyze events. In the event operations cease, the logs are very useful in determining activity leading up to the event and perhaps even revealing the root cause.

Of course, for these logs to be useful, they must be accurate and available, reinforcing the need for appropriate controls to be placed on them. Protection of the integrity of log records is critical to its usefulness, and the disclosure of this information could have a negative impact if vulnerabilities or flaws recorded in the logs are disclosed to the wrong parties. In many situations, the audit or operating system logs may be a target of attack by intruders or insiders.

Challenges in Management and Impact

Without a doubt, the operating system logs are a very important aspect of our systems, but the amount information being collected can be very difficult to manage effectively. The information contained in the logs is virtually useless unless it is reviewed on a regular basis for anomalous activities. This can be an arduous task for a group of individuals, let alone one person. The reviewers must know what they are looking for to appropriately interpret the information and take action. They must be able to spot trends, patterns, and variances that might indicate unusual behavior among the recorded events. When one considers the amount of information recorded in logs each day, and adds the responsibility of managing it in an effective manner to an already busy security professional, the challenge becomes all too apparent. This could easily become a full-time job for one person or a group of individuals.

If the management of this vast amount of information can cause a security professional to reach for pain relief, imagine the impact on a system during collection of this information — not to mention the additional overhead for storage and processing.

Fortunately, there are analysis tools designed to assist in the ongoing management of all this information. Audit reduction tools will reduce the amount of data by removing events that have little consequence on security, such as activities related to normal operations, making the remaining information more meaningful. Trends/variance detection and attack-signature detection tools similar to the functionality associated with intrusion detection systems will extract useful information from all the available raw data.

Conclusion

Operating system logs can provide a wealth of useful information in the ongoing security management of an organization's systems and resources, but not without a price. Managing this information in a meaningful way requires a commitment of time, computing resources, and perseverance, making it an ongoing challenge.

Other: Routers and Switches

Routers and switches play a critical role in enterprise networks. Routers connect different network segments and mediate in routing traffic from one segment to another. They can be considered as "sitting" at critical points of the network.

Routers are the glue that connects the pieces of a network. Even in the simplest networks, this is not a simple task.

Like routers, switches are critical components in networks. Switches sort out traffic intended for one network, while allowing separation of network segments.

Switches and routers continue to evolve into fairly complex devices with substantial computing power. Further, given their criticality in the network function, their impact on security is critical. Routers have evolved into highly specialized computing platforms, with extremely flexible but complex capabilities. Such complexity lends itself to vulnerabilities and attacks.

Issues pertaining to routers and switches deal with:

- *Access:* who has what access to the device
- *Configuration:* what kind of configuration ensures security of the device
- *Performance:* once deployed, how well it performs to meet intended requirements

It is of interest to track information on the above to ensure the "health" of the network devices and their performance. Ensuring device health means that the device is kept functioning based on its intended purposes. Not only must one keep track of changes and performance of the device, but one must also determine whether the changes are authorized and the impact on the security of the device.

Issues of managing routers and switches are similar to those pertaining to network devices such as firewalls and IDSs. Like firewalls and IDSs, switches and routers require due care; that is, logging suspicious activity, generating alarms where necessary, and constant reviewing of logged activity for purposes of improving their protection.

Similar to using firewalls and IDSs, users must ensure that routers and switches are not deployed with default configurations and that there exist processes for updating and patching the devices.

Typically, switches and routers are used in an internal network. For many, this may suggest a lower level of protection than that required for devices on the perimeter. Indeed, there are some who may feel that once the perimeter is secured, the degree of protection on the inside must be lower. This would be a false sense of security, considering that most attacks arise from sources in the internal network. Moreover, in the case of a successful attack, the intruder will have a free reign where there is insufficient protection on internal network devices.

There is more. The distinction between the inside and outside of a network continues to blur. Typically, an enterprise network is composed of intranets, extranets, the internal network, as well as the Internet. It takes the weakest link to break the security chain. And this could be the switch or router through which one links the business partner. As such, as much attention must be paid to managing these devices securely as is required for devices on the perimeter.

Security information from routers and switches should be part of total enterprise security information. This will help define a holistic picture of the enterprise security posture.

Strategies for Managing Enterprise Information

Managing security information presents a number of challenges to enterprise security and risk managers. The challenges include the potentially overwhelming amounts of information generated by a diverse number of network devices, analyzing the information, correlating security events, and relating technical risks to business risk. Moreover, security and risk managers must continuously perform these security-related activities to be aware of the organization's security posture while finding ways to improve this posture.

To have meaningful insight into an enterprise's security posture, information from diverse sources must be aggregated and correlated in a meaningful manner. Subsequent analysis would show underlying patterns, trends, and metrics associated with the information.

Patterns can be indicators of profiles of system usage. These profiles, in turn, may be due to the nature of the project and maintenance process for security in the enterprise. Trends, on the other hand, indicate variation in various security aspects over time. They may be indicators of improvements realized; they may also indicate problem areas that need improvements. Metrics and patterns are also useful for root cause analysis and problem resolution.

The most challenging tasks for security and risk managers include analyzing information collected across an enterprise from diverse network devices — devices that are used for different but complementary security functions. For example, information coming from firewalls, intrusion systems, and syslogs is complementary in nature with respect to security. Defining a suitable association for information from these diverse sources remains a key test.

Aside from dealing with the volumes of data collected, specific analysis techniques are required to ease the analysis process. These techniques can be based on anomalies, correlation, association with known exposures, trends, and user profiles. These techniques act as filters and aggregators for security information, converting massive data elements to useful information for decision-making.

In general, the range of issues is technical, process, people, and business related. They must be viewed with this totality for the information to be meaningful for improving security posture.

The remainder of this chapter section offers an overview of security information management issues and approaches to meeting the challenges of the complexity of managing the security information. While many practices identified in this chapter section are useful in improving an organization's security predisposition, their total application is key to improved enterprise security bearing. Practitioners of information security realize that security is both social and technical. As such, practices and norms in an organization, especially those pertaining to self-improvement, are core to improving the organization's security.

Security Data Collection and Log Management Issues

Collection, storage, and analysis of security-related information across an enterprise are crucial to understanding an organization's security predisposition. Managers must determine ways of managing the potentially huge amounts of information in a manner that makes business sense. Specifically, how does one translate technical vulnerability data to business risk?

There is a potential danger of being overwhelmed by the amount of information generated if proper filtering is not applied to ensure that only critical security-related information gets collected. The choice of filters for security-related information is borne out of experience and depends on the nature of the environment to which the information pertains.

There remains the challenge of collecting information in a manner that retains security-sensitive information and yet eliminates the amount of "noise" in the information collected. As an example, most intrusion detection systems raise a lot of false positives based on the configuration of the IDS sensors. In practice, a lot of information they generate can be classified as "white noise" and is of little value to security management. Security managers are faced with the challenge of designing appropriate filters to enable the filtering of white noise and thus lessen the burden of managing the collected information.

There are other fundamental issues pertaining to collecting security information. These include ensuring sufficient storage space for log information and periodic transmittal of collected information to a central log collection host. In many cases, projects are executed without sufficient planning for collection of log data, a fact that makes it extremely difficult to do root cause analysis when incidents occur.

Other log management issues pertain to the process in place for reviewing log information. In many organizations, information is logged but hardly examined to discern whether any serious security breach might have taken place. Given that security is more than technology, the process of managing security is as important as the technology used and the qualification of the people charged with managing that security. Technically proficient people managing security not only will understand the technology they are managing, but also appreciate security-related issues pertaining to their technology, including the role of the technology in ensuring security in the organization.

Issues of log management and associated technical personnel to execute them must be part of an organization's security management plan arising from an enterprise's security policy.

Data Interchange and Storage

The lack of an industry standard for exchange of security information presents a major problem for management of security information. Although the XML standard promises to close this gap, it has yet to be adopted as widely in industry as desirable. Thus, while users wait for vendors to adopt a standard for exchanging information, they must live with managing security from diverse sources in the different formats presented by vendors.

A security information exchange standard such as XML is one step, however. A long-term challenge is to find a common classification of security information from different products in the same security space. For example, IDSs would classify data the same way so that a security event generated by one IDS would be treated the same way as a similar event generated by an IDS from a different vendor.

Although there is no industry standard for data interchange yet, most products have the ability to store security-related information in a database. Being Open Database Connectivity (ODBC) compliant allows for data interchange between different programs and databases.

Storage issues include the determination of the amount of data collected and the format of storage. Typically, a database schema must be designed that makes sense with respect to the information collected. The schema will determine the nature of the breakdown of security events and their storage.

In designing storage requirements, security managers must incorporate such known concepts as backup and restoration properties. Others include high availability and remote access provision.

Correlation and Analysis

To get an enterprisewide security view, security information must be aggregated across the enterprise. This includes information from a diverse range of devices (intrusion detection systems, firewalls, system hosts, applications, routers, switches, and more) in the enterprise network. The above information, along with vulnerability data, can help discern an organization's security posture.

Log Data Analysis

Ultimately, the analysis of security information is intended to better understand the associated technical risk. Better still, the information would be useful for managing business risk.[6]

The information aggregation principle is based on the fact that the sum of the information from individual parts is less than the information obtained from the whole composed of the parts. Given the number of potential security-related events generated in an enterprise, there is a big challenge to associate, in a meaningful manner, related security-sensitive information or events.

A security event detected by an IDS can be related to a similar event recorded by the firewall or a Web server in an enterprise's DMZ. Indeed, such an event can be associated with specific activity in back-end systems behind the DMZ.

Event correlation has the power to give insight into the nature of a security-related event. One can play out different scenarios of how an event manifests itself once it hits an organization's network.

Take an event that has been detected by an IDS sitting on the DMZ. Now suppose that it was not detected by the firewall. This may be due to the failure to configure the firewall appropriately; and if the event is detected and blocked by the firewall, it is well and good. However, if it is not detected, it may require investigation. And if picked up by the Web server at the DMZ, then there is cause for concern. Correlation also allows for incorporation of the desirable response based on the criticality of the device under attack.[7]

It makes sense to associate security events seen at the firewall with those seen by the firewall and (if necessary) those happening in the DMZ and even the backend applications behind the DMZ. The collective picture gained is powerful and offers a more holistic view of security-related events in an organization.

Event correlation requires an enterprisewide strategy to become meaningful. Exhibit 66.1 depicts one possible way to organize collection and correlation of information. In this example, there are collections agents that can be configured in peer-to-peer or master-slave modes. Peer-to-peer agents have similar control in the communication. The master-slave relationship retains control within the matter.

To be effective and offer a totality of an organization's security posture, the agents must be capable of handling information from a diverse range of sources, including event logs, syslogs, intrusion systems, firewalls, routers, and switches. Special agents can also be deployed for specific network devices (e.g., firewalls) to offer a view of the security configuration of firewalls.[8]

The above example shows a possible scenario in which collection agents are deployed across the enterprise but organized along the way the enterprise is organized. This ensures that business units can collect their information and pass up the chain only specific flagged information, in the form of aggregates, that contributes to the overall picture of enterprise security posture.

There may be other models along which information is organized. For example, collection agents can be deployed as peer-to-peer, master-slave, or a mix of both. Organizations must determine which model best suits them.

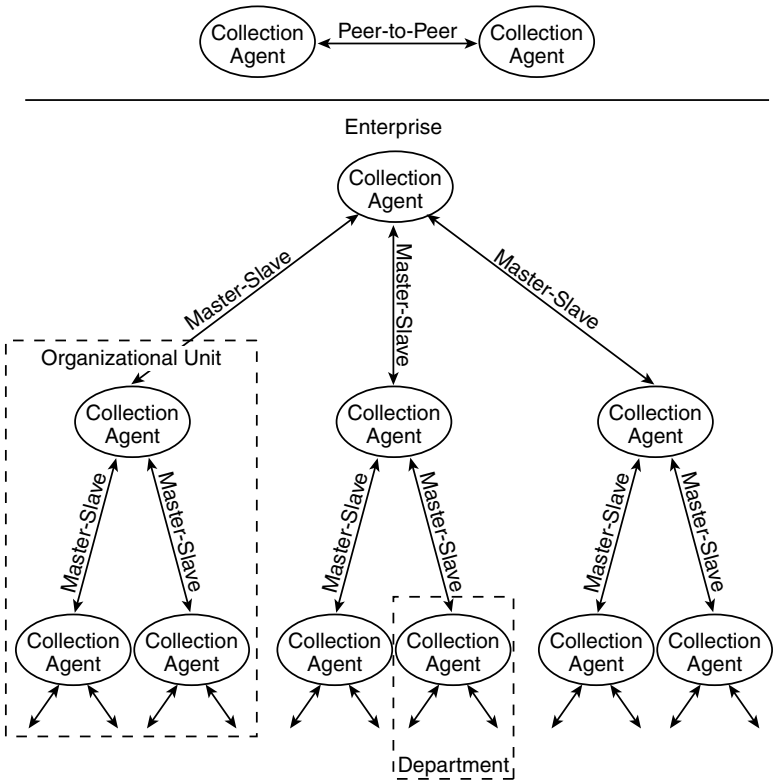


EXHIBIT 66.1 Collection and correlation of information.

Vulnerability Data

Log data analysis and correlation alone is not sufficient to ensure enterprise security posture, although it is important as a component of the "active security" of an organization. Typically, further analysis will include vulnerability data from network assessments.

Vulnerability data usually pertains to scans targeted at discovering such things as the number of ports open, the types of services running, the kind of exposures said services are vulnerable to, and the potential severity of these exposures.

There are few guidelines on the market indicating how vulnerability data should be manipulated. However, creating data mining can give indications on such aspects as the following:

- Risk profiles (e.g., per network, department, etc.) based on the number of vulnerabilities in that network segment
- Metrics about proportions of vulnerabilities regarded as high risk versus those with high risk
- Indication of trends of vulnerability data based on scans taken at different periods of time; interpolation and extrapolation of such trends will offer insight into any improvements in the security posture and whether or not there are improvements

Specific risk profiles will be useful in root cause analysis. It may be that certain vulnerability risk profiles indicate specific weaknesses pertaining to a number of factors such as the process of security planning, design and implementation, as well as the strength of the security process.

For security practitioners, the challenge is to determine the best way to present vulnerability data so as to help improve the way security is managed; specifically, lessons learned from correlation, trends in vulnerability data, and the metrics in performance as well as root cause analysis. And while these insights can be useful in managing security, the ultimate goal would be to associate technical vulnerability information to business risk. The data is not in yet but it is possible that certain vulnerability data profiles suggest specific types of likely business risks. Others, such as Donn Parker,[9] argue that such an approach is not

suitable. Instead, Parker advocates the concept of due care based on the fact that one cannot quantify the cost of avoiding potential security failure.

Summary and Conclusions

In an enterprise, there are diverse sources of security information that comes from devices that perform various network functions. Technologies such as firewalls and IDSs play a key role in enforcing security, while information from routers, switches, and system logs helps in providing a view of an organization's security posture.

Security managers face the challenge of collecting, storing, and analyzing the information in an effective manner that informs and improves the security management process. It must be understood that while security depends a lot on technology, it remains an issue pertaining to people and processes around managing security.

Strategies for the collection of information include the application of filters at strategic locations, complete with filters that pass only that information which must be passed on. This may use intelligent agents that correlate and aggregate information in useful ways to help minimize the amount of information collected centrally.

Security management is also faced with the challenge of creating measures for various aspects of enterprise security. These include metrics such as the percentage of network devices facing particular risks. This requires comparative criteria for measuring technical risk. Further, the said metrics can be used for root cause analysis to both identify and solve problems in a computing environment. Future challenges include being able to associate technical and business risk measures.

There is more. Taking technical risk numbers over time can be used to obtain trends. Such trends would indicate whether there are improvements to the organization's security posture over time.

Security and risk managers, apart from understanding the function of network technologies, face other major challenges. Coming to grips with the reality of the complexity of security management is one step. Defining clear processes and means of managing the information is a step ahead. Yet, using information in a manner that informs the security process and contributes to improvement of the security posture would be a major achievement. Finally, defining metrics and trends useful for managing business risk will be a critical test in the future.

Notes

1. We deliberately use the term "health" because an organization's security posture can be seen in terms of health.
2. This is true for a single access point network. This claim is questionable for networks with multiple access points that blur the concept of what is in and what is out.
3. Most vendors offer high-availability solutions. This, however, is additional cost to network infrastructure. As well, load balancing is a challenge for many firewall vendors.
4. The Honeynet Project (www.honeynet.org) has taken this concept further by creating typical environments for attack and using forensic methodologies to study attacks, their sources, and the motives of attackers.
5. This is the case where users have dialup access to the Internet while logged on to the internal network.
6. Few organizations have been successful in showing the link between technical vulnerability/risk data and associated business risk.
7. Event correlation is dealt with in greater detail in Matunda Nyanchama and Paul Sop's Enterprise Security Management: Managing Complexity, in *Information Systems Security,* January/February 2001.
8. There are products on the market that claim to perform event correlation across different network devices. As of writing this chapter (March 2001), there is no a single product with convincing performance to warrant such a claim.
9. See Risk Reduction Out, Enablement and Due Care In, in *CSI Computer Security Journal*, Vol. XVI, #4, Fall 2000.

Bibliography

Zwicky, E.D., Cooper, S., and Chapman, D.B., *Building Internet Firewalls,* 2nd edition, O'Reilly, 2000.
http://csrc.nist.gov/publications/nistpubs/800-7/node155.html.

Wack, J. and Carnhan, L., Keeping Your Site Comfortably Secure: An Introduction to Internet Firewalls. NIST Special Publication 800-10. U.S. Department of Commerce. National Institute of Standards and Technology, February 1995, http://csrc.nist.gov/publications/nistpubs/800-10/main.html.

Ballew, S.M., *Managing IP Networks with Cisco Routers,* 1st edition, O'Reilly, 1997.

Goncalves, M., *Firewalls Complete,* McGraw-Hill, 1998.

Syslog the UNIX System Logger, http://www.scrambler.net/syslog.htm.

http://njlug.rutgers.edu/projects/syslog/.

Explanation and FAQ for RME Syslog Analyzer, http://www.cisco.com/warp/public/477/RME/rme_syslog.html.

Marshall, V.H., Intrusion Detection in Computers. Summary of the Trusted Information Systems (TIS) Report on Intrusion Detection Systems, January 1991.

Carson, M. and Zink, M., NIST Switch: A Platform for Research on Quality of Service Routing, 1998, http://www.antd.nist.gov/itg/nistswitch/qos.spie.ps.

Parker, D., Risk Reduction Out, Enablement and Due Care In, in *CSI Computer Security Journal,* Vol. XVI, #4, Fall 2000.

Nyanchama, M. and Sop, P., Enterprise Security Management: Managing Complexity, in *Information Systems Security,* January/February 2001.

Base, R. and Mell, P., NIST Special Publication on Intrusion Detection Systems.

Security Portal; The Joys of the Incident Handling Response Process.

Ranum, M., Intrusion Detection Ideals, Expectations and Realities.

NIST Special Publication, 800-12, Introduction to Computer Security, *The NIST Handbook.*

67

Risk Analysis and Assessment

Will Ozier

There are a number of ways to identify, analyze, and assess risk, and there is considerable discussion of "risk" in the media and among information security professionals. But, there is little real understanding of the process and metrics of analyzing and assessing risk. Certainly everyone understands that "taking a risk" means "taking a chance," but a risk or chance of what is often not so clear.

When one passes on a curve or bets on a horse, one is taking a chance of suffering harm/injury or financial loss — undesirable outcomes. We usually give a degree of more or less serious consideration to such an action before taking the chance, so to speak. Perhaps we would even go so far as to calculate the odds (chance) of experiencing the undesirable outcome and, further, take steps to reduce the chance of experiencing the undesirable outcome.

To effectively calculate the chance of experiencing the undesirable outcome, as well as its magnitude, one must be aware of and understand the elements of risk and their relationship to each other. This, in a nutshell, is the process of risk analysis and assessment.

Knowing more about the risk, one is better prepared to decide what to do about it — accept the risk as now assessed (go ahead and pass on the blind curve or make that bet on the horse), or mitigate the risk. To mitigate the risk is to do something to reduce the risk to an acceptable level (wait for a safe opportunity to pass or put the bet money in a savings account with interest).

There is a third choice; to transfer the risk, that is, buy insurance. However prudent good insurance may be, all things considered, having insurance will not prevent the undesirable outcome. Having insurance will only serve to make some compensation — almost always less than complete — for the loss. Further, some risks — betting on a horse — are uninsurable.

The processes of identifying, analyzing and assessing, mitigating, or transferring risk are generally characterized as Risk Management.

There are a few key questions at the core of the risk management process:

1. What could happen (threat event)?
2. If it happened, how bad could it be (threat impact)?
3. How often could it happen (threat frequency, annualized)?
4. How certain are the answers to the first three questions (recognition of uncertainty)?

These questions are answered by analyzing and assessing risk.

Uncertainty is the central issue of risk. Sure, one might pass successfully on the curve or win big at the races, but does the gain warrant taking the risk? Do the few seconds saved with the unsafe pass warrant the possible head-on collision? Are you betting this month's paycheck on a long shot to win? Cost/benefit analysis would most likely indicate that both of these examples are unacceptable risks.

Prudent management, having analyzed and assessed the risks by securing credible answers to these four questions, will almost certainly find there to be some unacceptable risks as a result. Now what? Three questions remain to be answered:

1. What can be done (risk mitigation)?
2. How much will it cost (annualized)?
3. Is it cost effective (cost/benefit analysis)?

Answers to these questions, decisions to budget and execute recommended activities, and the subsequent and ongoing management of all risk mitigation measures — including periodic reassessment — comprise the balance of the Risk Management process.

Managing the risks associated with information in the information technology (IT) environment, information risk management, is an increasingly complex and dynamic task. In the budding Information Age, the technology of information storage, processing, transfer, and access has exploded, leaving efforts to secure that information effectively in a never-ending catch-up mode. For the risks potentially associated with information and information technology to be identified and managed cost-effectively, it is essential that the process of analyzing and assessing risk is well understood by all parties — and executed on a timely basis. This chapter is written with the objective of illuminating the process and the issues of risk analysis and assessment.

Terms and Definitions

To discuss the history and evolution of information risk analysis and assessment, several terms whose meanings are central to this discussion should first be defined.

Annualized Loss Expectancy (ALE)

This discrete value is derived, classically, from the following algorithm (see also the definitions for single loss expectancy [SLE] and annualized rate of occurrence [ARO] below):

$$\begin{matrix}\text{SINGLE LOSS}\\ \text{EXPECTANCY}\end{matrix} \times \begin{matrix}\text{ANNUALIZEDRATE}\\ \text{OFOCCURRENCE}\end{matrix} = \begin{matrix}\text{ANNUALIZED LOSS}\\ \text{EXPECTANCY}\end{matrix}$$

To effectively identify the risks and to plan budgets for information risk management, it is helpful to express loss expectancy in annualized terms. For example, the preceding algorithm will show that the **ALE** for a threat (with an **SLE** of $1,000,000) that is expected to occur only about once in 10,000 years is ($1,000,000 divided by 10,000) only $100.00. When the expected threat frequency (**ARO**) is factored into the equation, the significance of this risk factor is addressed and integrated into the information risk management process. Thus, the risks are more accurately portrayed, and the basis for meaningful cost/benefit analysis of risk reduction measures is established.

Annualized Rate of Occurrence (ARO)

This term characterizes, on an annualized basis, the frequency with which a threat is expected to occur. For example, a threat occurring once in 10 years has an **ARO** of 1/10 or 0.1; a threat occurring 50 times in a given year has an **ARO** of 50.0. The possible range of frequency values is from 0.0 (the threat is not expected to occur) to some whole number whose magnitude depends on the type and population of threat sources. For example, the upper value could exceed 100,000 events per year for minor, frequently experienced threats such as misuse of resources. For an example of how quickly the number of threat events can mount, imagine a small organization — about 100 staff members — having logical access to an information processing system. If each of those 100 persons misused the system only once a month, misuse events would be occurring at the rate of 1200 events per year. It is useful to note here that many confuse **ARO** or frequency with the term and concept of probability (defined below). While the statistical and mathematical significance of these frequency and probability metrics tends to converge at about 1/100 and become essentially indistinguishable below that level of frequency or probability, they become increasingly divergent above 1/100 to the point where probability stops — at 1.0 or certainty — and frequency continues to mount undeterred, by definition.

Exposure Factor (EF)

This factor represents a measure of the magnitude of loss or impact on the value of an asset. It is expressed as a percent, ranging from 0 to 100 percent, of asset value loss arising from a threat event. This factor is used in the calculation of single loss expectancy (**SLE**), which is defined below.

Information Asset

This term, in general, represents the body of information an organization must have to conduct its mission or business. A specific information asset may consist of any subset of the complete body of information, that is, accounts payable, inventory control, payroll, etc. Information is regarded as an intangible asset separate from the media on which it resides. There are several elements of value to be considered: first is the simple cost of replacing the information; second is the cost of replacing supporting software; and third through fifth is a series of values that reflect the costs associated with loss of the information's confidentiality, availability, and integrity. Some consider the supporting hardware and netware to be information assets as well. However, these are distinctly tangible assets. Therefore, using tangibility as the distinguishing characteristic, it is logical to characterize hardware differently than the information itself. Software, on the other hand, is often regarded as information.

These five elements of the value of an information asset often dwarf all other values relevant to an assessment of information-related risk. It should be noted that these elements of value are not necessarily additive for the purpose of assessing risk. In both assessing risk and establishing cost-justification for risk-reducing safeguards, it is useful to be able to isolate the value of safeguard effects among these elements.

Clearly, for an organization to conduct its mission or business, the necessary information must be present where it is supposed to be, when it is supposed to be there, and in the expected form. Further, if desired confidentiality is lost, results could range from no financial loss if confidentiality is not an issue, to loss of market share in the private sector, to compromise of national security in the public sector.

Qualitative/Quantitative

These terms indicate the (oversimplified) binary categorization of risk metrics and information risk management techniques. In reality, there is a spectrum across which these terms apply, virtually always in combination. This spectrum may be described as the degree to which the risk management process is quantified. If all elements — asset value, impact, threat frequency, safeguard effectiveness, safeguard costs, uncertainty, and probability — are quantified, the process can be characterized as fully quantitative.

It is virtually impossible to conduct a purely quantitative risk management project because the quantitative measurements must be applied to the qualitative properties, that is, characterizations of vulnerability, of the target environment. For example, "failure to impose logical access control" is a qualitative statement of vulnerability. However, it is possible to conduct a purely qualitative risk management project. A vulnerability analysis, for example, may identify only the absence of risk-reducing countermeasures, such as logical access controls. Even this simple qualitative process has an implicit quantitative element in its binary — yes/no — method of evaluation. In summary, risk analysis and assessment techniques should be described not as either qualitative or quantitative but in terms of the degree to which such elementary factors as asset value, exposure factor, and threat frequency are assigned quantitative values.

Probability

This term characterizes the chance or likelihood, in a finite sample, that an event will occur or that a specific loss value may be attained should the event occur. For example, the probability of getting a six on a single roll of a die is 1/6, or 0.16667. The possible range of probability values is 0.0 to 1.0. A probability of 1.0 expresses certainty that the subject event will occur within the finite interval. Conversely, a probability of 0.0 expresses certainty that the subject event will not occur within the finite interval.

Risk

The potential for harm or loss, best expressed as the answer to those four questions:

- What could happen? (What is the threat?)
- How bad could it be? (What is the impact or consequence?)
- How often might it happen? (What is the frequency?)
- How certain are the answers to the first three questions? (What is the degree of confidence?)

The key element among these is the issue of uncertainty captured in the fourth question. If there is no uncertainty, there is no "risk," per se.

Risk Analysis

This term represents the process of analyzing a target environment and the relationships of its risk-related attributes. The analysis should identify threat vulnerabilities, associate these vulnerabilities with affected assets, identify the potential for and nature of an undesirable result, and identify and evaluate risk-reducing countermeasures.

Risk Assessment

This term represents the assignment of value to assets, threat frequency (annualized), consequence (i.e., exposure factors), and other elements of chance. The reported results of risk analysis can be said to provide an assessment or measurement of risk, regardless of the degree to which quantitative techniques are applied. For consistency in this chapter, the term "risk assessment" hereafter is used to characterize both the process and the results of analyzing and assessing risk.

Risk Management

This term characterizes the overall process. The first phase, risk assessment, includes identification of the assets at risk and their value, risks that threaten a loss of that value, risk-reducing measures, and the budgetary impact of implementing decisions related to the acceptance, mitigation, or transfer of risk. The second phase of risk management includes the process of assigning priority to, budgeting, implementing, and maintaining appropriate risk-reducing measures. Risk management is a continuous process.

Safeguard

This term represents a risk-reducing measure that acts to detect, prevent, or minimize loss associated with the occurrence of a specified threat or category of threats. Safeguards are also often described as controls or countermeasures.

Safeguard Effectiveness

This term represents the degree, expressed as a percent, from 0 to 100 percent, to which a safeguard may be characterized as effectively mitigating a vulnerability (defined below) and reducing associated loss risks.

Single Loss Expectancy or Exposure (SLE)

This value is classically derived from the following algorithm to determine the monetary loss (impact) for each occurrence of a threatened event:

$$\text{ASSET VALUE} \times \text{EXPOSURE FACTOR} = \text{SINGLE LOSS EXPECTANCY}$$

The **SLE** is usually an end result of a business impact analysis (BIA). A BIA typically stops short of evaluating the related threats' **ARO** or their significance. The **SLE** represents only one element of risk, the expected impact, monetary or otherwise, of a specific threat event. Because the BIA usually characterizes the massive losses resulting from a catastrophic event, however improbable, it is often employed as a scare tactic to get management attention — and loosen budgetary constraints — often unreasonably.

Threat

This term defines an event (e.g., a tornado, theft, or computer virus infection), the occurrence of which could have an undesirable impact.

Uncertainty

This term characterizes the degree, expressed as a percent, from 0.0 to 100 percent, to which there is less than complete confidence in the value of any element of the risk assessment. Uncertainty is typically measured inversely with respect to confidence; that is, if confidence is low, uncertainty is high.

Vulnerability

This term characterizes the absence or weakness of a risk-reducing safeguard. It is a condition that has the potential to allow a threat to occur with greater frequency, greater impact, or both. For example, not having a fire suppression system could allow an otherwise minor, easily quenched fire to become a catastrophic fire. The expected frequency (**ARO**) and the exposure factor (**EF**) for major and catastrophic fire are both increased as a consequence of not having a fire suppression system.

Central Tasks of Information Risk Management

The following sections describe the tasks central to the comprehensive information risk management process. These tasks provide concerned management with credible decision support information regarding the identification and valuation of assets potentially at risk, an assessment of risk, and cost-justified recommendations for risk reduction. Thus, the execution of well-informed management decisions whether to accept, mitigate, or transfer risk cost-effectively is supported. The degree of quantitative orientation determines how the results are characterized, and, to some extent, how they are used. Each of these tasks is discussed below.

Establish Information Risk Management (IRM) Policy

A sound IRM program is founded on a well-thought-out IRM policy infrastructure that effectively addresses all elements of information security. Generally Accepted Information Security Principles (GASSP), currently being developed, based on an Authoritative Foundation of supporting documents and guidelines, will be helpful in executing this task.

IRM policy should begin with a high-level policy statement and supporting objectives, scope, constraints, responsibilities, and approach. This high-level policy statement should drive subordinate policy, from logical access control to facilities security to contingency planning.

Finally, IRM policy should be communicated effectively — and enforced — to all parties. Note that this is important for both internal control and external control — EDI, the Web, and the Internet — for secure interface with the rest of the world.

Establish and Fund an IRM Team

Much of the IRM functionality should already be in place — logical access control, contingency planning, etc. However, it is likely that the central task of IRM, risk assessment, has not been built into the established approach to IRM or has, at best, been given only marginal support.

At the most senior management level possible, the tasks and responsibilities of IRM should be coordinated and IRM-related budgets cost-justified based on a sound integration and implementation of the risk assessment process. At the outset, the IRM team may be drawn from existing IRM-related staff. The person charged with responsibility for executing risk assessment tasks should be an experienced IT generalist with a sound understanding of the broad issues of information security and the ability to "sell" these concepts to management. This person will need the incidental support of one who can assist at key points of the risk assessment task, that is, scribing a Modified Delphi information valuation (see below for details).

In the first year of an IRM program, the lead person could be expected to devote 50 to 75 percent of his or her time to the process of establishing and executing the balance of the IRM tasks, the first of which follows immediately below. Funds should be allocated (1) according to the above minimum staffing, and (2) to acquire, and be trained in the use of, a suitable automated risk assessment tool — $25 to 35K.

Establish IRM Methodology and Tools

There are two fundamental applications of risk assessment to be addressed: (1) determining the current status of information security in the target environment(s) and ensuring that associated risk is managed (accepted, mitigated, or transferred) according to policy, and (2) assessing risk strategically. Strategic assessment assures that the risks associated with alternative strategies are effectively considered before funds are expended on a specific change in the IT environment, a change that could have been shown to be "too risky." Strategic assessment allows management to effectively consider the risks associated with various strategic alternatives in its decision-making process and weigh those risks against the benefits and opportunities associated with each alternative business or technical strategy.

With the availability of proven automated risk assessment tools, the methodology is, to a large extent, determined by the approach and procedures associated with the tool of choice. An array of such tools is listed at the end of this chapter. Increasingly, management is looking for quantitative results that support a credible cost/benefit analysis and budgetary planning.

Identify and Measure Risk

Once the IRM policy, team, and risk assessment methodology and tools are established and acquired, the first risk assessment will be executed. This first risk assessment should be scoped as broadly as possible, so that (1) management is provided with a good sense of the current status of information security, and (2) management has a sound basis for establishing initial risk acceptance criteria and risk mitigation priorities.

Project Sizing

This task includes the identification of background, scope, constraints, objectives, responsibilities, approach, and management support. Clear project sizing statements are essential to a well-defined and well-executed risk assessment project. It should also be noted that a clear articulation of project constraints (what is not included in the project) is very important to the success of a risk assessment.

Threat Analysis

This task includes the identification of threats that may adversely impact the target environment. This task is important to the success of the entire IRM program and should be addressed, at least initially, by risk assessment experts to ensure that all relevant risks are adequately considered. One without risk management and assessment experience may fail to consider a threat, whether of natural causes or the result of human behavior, that stands to cause substantial harm or loss to the organization. Some risk assessment tools, such as BDSS™, help to preclude this problem by assuring that all threats are addressed as a function of expert system knowledge bases.

Asset Identification and Valuation

This task includes the identification of assets, both tangible and intangible, their replacement costs, and the further valuing of information asset availability, integrity, and confidentiality. These values may be expressed in monetary (for quantitative) or nonmonetary (for qualitative) terms. This task is analogous to a BIA in that it identifies the assets at risk and their value.

Vulnerability Analysis

This task includes the qualitative identification of vulnerabilities that could increase the frequency or impact of threat event(s) affecting the target environment.

Risk Evaluation

This task includes the evaluation of all collected information regarding threats, vulnerabilities, assets, and asset values in order to measure the associated chance of loss and the expected magnitude of loss for each of an array of threats that could occur. Results are usually expressed in monetary terms on an annualized basis (ALE) or graphically as a probabilistic "risk curve" for a quantitative risk assessment. For a qualitative risk assessment, results are usually expressed through a matrix of qualitative metrics such as ordinal ranking (low, medium, high or 1, 2, 3).

Interim Reports and Recommendations

These key reports are often issued during this process to document significant activity, decisions, and agreements related to the project:

- *Project sizing*. This report presents the results of the project sizing task. The report is issued to senior management for their review and concurrence. This report, when accepted, assures that all parties understand and concur in the nature of the project before it is launched.
- *Asset identification and valuation*. This report may detail (or summarize) the results of the asset valuation task, as desired. It is issued to management for their review and concurrence. Such review helps prevent conflict about value later in the process. This report often provides management with their first insight into the value of the availability, confidentiality, or integrity of their information assets.
- *Risk evaluation*. This report presents management with a documented assessment of risk in the current environment. Management may choose to accept that level of risk (a legitimate management decision) with no further action or to proceed with risk mitigation analysis.

Establish Risk Acceptance Criteria

With the results of the first risk assessment — through the risk evaluation task and associated reports (see below) — management, with the interpretive help from the IRM leader, should establish the maximum acceptable financial risk. For example, "Do not accept more than a 1 in 100 chance of losing $1,000,000," in a given year. And, with that, and possibly additional risk acceptance criteria, such as "Do not accept an ALE greater than $500,000," proceed with the task of risk mitigation.

Mitigate Risk

The first step in this task is to complete the risk assessment with the risk mitigation, costing, and cost/benefit analysis. This task provides management with the decision support information necessary to plan for, budget, and execute actual risk mitigation measures. In other words, fix the financially unacceptable vulnerabilities. The following risk assessment tasks are discussed in further detail under the section "Tasks of Risk Assessment" later in this chapter.

Safeguard Selection and Risk Mitigation Analysis

This task includes the identification of risk-reducing safeguards that mitigate vulnerabilities and the degree to which selected safeguards can be expected to reduce threat frequency or impact. In other words, this task comprises the evaluation of risk regarding assets and threats before and after selected safeguards are applied.

Cost/Benefit Analysis

This task includes the valuation of the degree of risk mitigation that is expected to be achieved by implementing the selected risk-mitigating safeguards. The gross benefit less the annualized cost for safeguards selected to achieve a reduced level of risk, yields the net benefit. Tools such as present value and return on investment are often applied to further analyze safeguard cost-effectiveness.

Final Report

This report includes the interim reports' results as well as details and recommendations from the safeguard selection and risk mitigation analysis, and supporting cost/benefit analysis tasks. This report, with approved recommendations, provides responsible management with a sound basis for subsequent risk management action and administration.

Monitor Information Risk Management Performance

Having established the IRM program, and gone this far — recommended risk mitigation measures have been acquired/developed and implemented — it is time to begin and maintain a process of monitoring IRM performance. This can be done by periodically reassessing risks to ensure that there is sustained adherence to good control or that failure to do so is revealed, consequences considered, and improvement, as appropriate, duly implemented.

Strategic risk assessment plays a significant role in the risk mitigation process by helping to avoid uninformed risk acceptance and having, later, to retrofit (typically much more costly than built-in security or avoided risk) necessary information security measures.

There are numerous variations on this risk management process, based on the degree to which the technique applied is quantitative and how thoroughly all steps are executed. For example, the asset identification and valuation analysis could be performed independently. This task is often characterized as a business impact analysis. The vulnerability analysis could also be executed independently.

It is commonly but incorrectly assumed that information risk management is concerned only with catastrophic threats, that it is useful only to support contingency planning and related activities. A well-conceived and well-executed risk assessment can, and should, be used effectively to identify and quantify the consequences of a wide array of threats that can and do occur, often with significant frequency, as a result of ineffectively implemented or nonexistent IT management, administrative, and operational controls.

A well-run information risk management program — an integrated risk management program — can help management to significantly improve the cost-effective performance of its information technology environment, whether it is mainframe, client/server, Internet, or any combination, and to ensure cost-effective compliance with applicable regulatory requirements.

The integrated risk management concept recognizes that many often uncoordinated units within an organization play an active role in managing the risks associated with the failure to assure the confidentiality, availability, and integrity of information. The following quote from FIPSPUB-73, published June 30, 1980, is a powerful reminder that information security was long ago recognized as a central, not marginal issue:

> "Security concerns should be an integral part of the entire planning, development, and operation of a computer application. Much of what needs to be done to improve security is not clearly separable from what is needed to improve the usefulness, reliability, effectiveness, and efficiency of the computer application."

Resistance and Benefits

"Why should I bother with doing risk assessment?" "I already know what the risks are!" "I've got enough to worry about already!" "It hasn't happened yet...." Sound familiar? Most resistance to risk assessment boils down to one of three conditions:

1. Ignorance
2. Arrogance
3. Fear

Management is often ignorant, except in the most superficial context, of the risk assessment process, the real nature of the risks, and the benefits of risk assessment. Risk assessment is not yet a broadly accepted element of the management toolkit, yet virtually every "Big 5" consultancy, and other major providers of information security services, offer risk assessment in some form.

Arrogance of the bottom line often drives an organization's attitude about information security, and therefore about risk assessment. "Damn the torpedoes, full speed ahead!" becomes the marching order. If it can not readily be shown to improve profitability, do not do it. It is commendable that IT has become so reliable that management could maintain that attitude for more than a few giddy seconds. Despite the fact that a well-secured IT environment is also a well-controlled, efficient IT environment, management often has difficulty seeing how sound information security can and does affect the bottom line in a positive way.

This arrogance is often described euphemistically as an "entrepreneurial culture."

Finally, there is the fear factor — fear of discovering that the environment is not as well-managed as it could be — and having to take responsibility for that; fear of discovering, and having to address, risks not already known; and fear of being shown to be ignorant or arrogant.

While good information security may seem expensive, inadequate information security will be not just expensive, but, sooner or later, catastrophic.

Risk assessment, while still a young science, with a certain amount of craft involved, has proven itself to be very useful in helping management understand and cost-effectively address the risks to their information and IT environments.

Finally, with regard to resistance, when risk assessment had to be done manually, or could be done only qualitatively, the fact that the process could take many months to execute (and that it was not amenable to revision or "what-if" assessment) was a credible obstacle to its successful use. But that is no longer the case.

Some specific benefits are described below:

- Risk assessment helps management understand:
  1. What is at risk
  2. The value at risk — as associated with the identity of information assets and with the confidentiality, availability, and integrity of information assets
  3. The kinds of threats that could occur and their financial consequences annualized
  4. Risk mitigation analysis; what can be done to reduce risk to an acceptable level
  5. Risk mitigation costs (annualized) and associated cost/benefit analysis; whether suggested risk mitigation activity is cost-effective
- Risk assessment enables a strategic approach to information risk management. In other words, possible changes being considered for the IT environment can be assessed to identify the least-risk alternative before funds are committed to any alternative. This information complements the standard business case for change and may produce critical decision support information that could otherwise be overlooked.

- "What-if" analysis is supported. This is a variation on the strategic approach to information risk management. Alternative approaches can be considered and their associated level of risk compared in a matter of minutes.
- Information security professionals can present their recommendations with credible statistical and financial support.
- Management can make well-informed information risk management decisions.
- Management can justify, with credible quantitative tools, information security budgets/expenditures that are based on a reasonably objective risk assessment.
- Good information security, supported by quantitative risk assessment, will ensure an efficient, cost-effective IT environment.
- Management can avoid spending that is based solely on a perception of risk.
- An information risk management program based on the sound application of quantitative risk assessment can be expected to reduce liability exposure and insurance costs.

Qualitative versus Quantitative Approaches

Background

As characterized briefly above, there are two fundamentally different metric schemes applied to the measurement of risk elements: qualitative and quantitative. The earliest efforts to develop an information risk assessment methodology were reflected originally in the National Bureau of Standards (now the National Institute of Standards and Technology [NIST]) FIPSPUB-31, Automated Data Processing Physical Security and Risk Management, published in 1974. That idea was subsequently articulated in detail with the publication of FIPSPUB-65, Guidelines for Automated Data Processing Risk Assessment, published in August of 1979. This methodology provided the underpinnings for OMB A-71, a federal requirement for conducting "quantitative risk assessment" in the federal government's information processing environments.

Early efforts to conduct quantitative risk assessments ran into considerable difficulty. First, because no initiative was executed to establish and maintain an independently verifiable and reliable set of risk metrics and statistics, everyone came up with their own approach; second, the process, while simple in concept, was complex in execution; third, large amounts of data were collected that required substantial and complex mapping, pairing, and calculation to build representative risk models; and fourth, with no software and desktop computers, the work was done manually — a very tedious and time-consuming process. Results varied significantly.

As a consequence, while some developers launched and continued efforts to develop credible and efficient automated quantitative risk assessment tools, others developed more expedient qualitative approaches that did not require independently objective metrics — and OMB A-130, an update to OMB A-71, was released, lifting the "quantitative" requirement for risk assessment in the federal government.

These qualitative approaches enabled a much more subjective approach to the valuation of information assets and the scaling of risk. In Exhibit 67.1, for example, the value of the availability of information and the associated risk were described as "low," "medium," or "high" in the opinion of knowledgeable management, as gained through interviews or questionnaires.

Often, when this approach is taken, a strategy is defined wherein the highest risk exposures (darkest shaded areas) require prompt attention, the moderate risk exposures (lightly shaded areas) require plans for corrective attention, and the lowest risk exposures (unshaded areas) can be accepted.

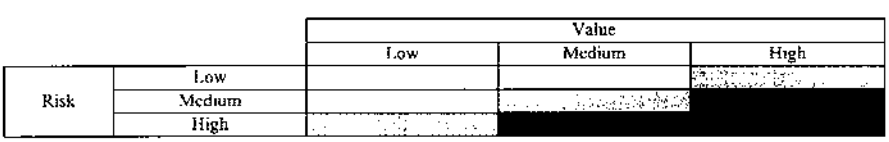


EXHIBIT 67.1 Value of the availability of information and the associated risk.

Elements of Risk Metrics

There are six primitive elements of risk modeling to which some form of metric can be applied:

1. Asset Value
2. Threat Frequency
3. Threat Exposure Factor
4. Safeguard Effectiveness
5. Safeguard Cost
6. Uncertainty

To the extent that each of these elements is quantified in independently objective metrics such as the monetary replacement value for Asset Value or the Annualized Rate of Occurrence for Threat Frequency, the risk assessment is increasingly quantitative. If all six elements are quantified with independently objective metrics, the risk assessment is fully quantified, and the full range of statistical analyses is supported.

Exhibit 67.2 relates both the quantitative and qualitative metrics for these six elements.

Note: The baseline approach makes no effort to scale risk or to value information assets. Rather, the baseline approach seeks to identify in-place safeguards, compare those with what industry peers are doing to secure their information, then enhance security wherever it falls short of industry peer security. A further word of caution is appropriate here. The baseline approach is founded on an interpretation of "due care" that is at odds with the well-established legal definition of due care. Organizations relying solely on the baseline approach could find themselves at a liability risk with an inadequate legal defense should a threat event cause a loss that could have been prevented by available technology or practice that was not implemented because the baseline approach was used.

The classic quantitative algorithm, as presented in FIPSPUB-65, that laid the foundation for information security risk assessment is simple:

$$\text{Annualized Loss Expectancy} = (\text{Asset Value} \times \text{Exposure Factor} = \text{Single Loss Exposure}) (\text{Annualized R of O})$$

For example, let's look at the risk of fire. Assume the asset value is \$1M, the Exposure Factor is 50%, and the annualized rate of occurrence is 1/10 (once in ten years). Plugging these values into the algorithm yields the following:

$$(\$1M \times 50\% = \$500K) \times 1/10 = \$50K$$

Using conventional cost/benefit assessment, the \$50K ALE represents the cost/benefit break-even point for risk mitigation measures. In other words, the organization could justify spending up to \$50K per year to prevent the occurrence or reduce the impact of a fire.

It is true that the classic FIPSPUB-65 quantitative risk assessment took the first steps toward establishing a quantitative approach. However, in the effort to simplify fundamental statistical analysis processes so that everyone could readily understand, the algorithms developed went too far. The consequence was results that had little credibility for several reasons, three of which follow:

1. The classic algorithm addresses all but two of the elements, recommended safeguard effectiveness and uncertainty. Both of these must be addressed in some way, and uncertainty, the key risk factor, must be addressed explicitly.
2. The algorithm cannot distinguish effectively between low frequency/high-impact threats (such as "fire") and high-frequency/low impact threats (such as "misuse of resources"). Therefore, associated risks can be significantly misrepresented.
3. Each element is addressed as a discrete value, which, when considered with the failure to address uncertainty explicitly, makes it difficult to actually model risk and illustrate probabilistically the range of potential undesirable outcomes.

Yes, this primitive algorithm did have shortcomings, but advances in quantitative risk assessment technology and methodology to explicitly address uncertainty and support technically correct risk modeling have largely done away with those problems.

	Quantitative Metrics				Qualitative Metrics			
Risk Element	Monetary Value	Percent Factors (%)	Annualized Rate of Occurrence	Bounded Distribution (Range)	Low, Medium & High	Ordinal Ranking	Vital, Critical, Important, etc.	Baseline
Asset Value	x			x	x	x	x	
Threat Frequency (Annualized)			x	x	x	x		
Threat Exposure Factor		x		x	x	x		
Recommended Safeguard Effectiveness		x		x	x	x		
Safeguard Cost (Annualized)	x			x	x	x		
Uncertainty (Confidence Factor)		x		x	x	x		

EXHIBIT 67.2 Quantitative and qualitative metrics for the six elements.

Pros and Cons of Qualitative and Quantitative Approaches

In this brief analysis, the features of specific tools and approaches will not be discussed. Rather, the pros and cons associated in general with qualitative and quantitative methodologies will be addressed.

Qualitative — Pros

- Calculations, if any, are simple and readily understood and executed.
- It is usually not necessary to determine the monetary value of information (its availability, confidentiality, and integrity).
- It is not necessary to determine quantitative threat frequency and impact data.
- It is not necessary to estimate the cost of recommended risk mitigation measures and calculate cost/benefit.
- A general indication of significant areas of risk that should be addressed is provided.

Qualitative — Cons

- The risk assessment and results are essentially subjective in both process and metrics. The use of independently objective metrics is eschewed.
- No effort is made to develop an objective monetary basis for the value of targeted information assets. Hence, the perception of value may not realistically reflect actual value at risk.
- No basis is provided for cost/benefit analysis of risk mitigation measures, only subjective indication of a problem.
- It is not possible to track risk management performance objectively when all measures are subjective.

Quantitative — Pros

- The assessment and results are based substantially on independently objective processes and metrics. Thus, meaningful statistical analysis is supported.
- The value of information (availability, confidentiality, and integrity), as expressed in monetary terms with supporting rationale, is better understood. Thus, the basis for expected loss is better understood.
- A credible basis for cost/benefit assessment of risk mitigation measures is provided. Thus, information security budget decision-making is supported.
- Risk management performance can be tracked and evaluated.
- Risk assessment results are derived and expressed in management's language, monetary value, percentages, and probability annualized. Thus, risk is better understood.

Quantitative — Cons

- Calculations are complex. If they are not understood or effectively explained, management may mistrust the results of "black-box" calculations.
- It is not practical to attempt to execute a quantitative risk assessment without using a recognized automated tool and associated knowledge bases. A manual effort, even with the support of spreadsheet and generic statistical software, can easily take ten to twenty times the work effort required with the support of a good automated risk assessment tool.
- A substantial amount of information about the target information and its IT environment must be gathered.
- As of this writing, there is not yet a standard, independently developed and maintained threat population and threat frequency knowledge base. Thus, users must rely on the credibility of the vendors who develop and support extant automated tools or do threat research on their own.

Business Impact Analysis versus Risk Assessment

There is still confusion as to the difference between a Business Impact Analysis (BIA) and risk assessment. It is not unusual to hear the terms used interchangeably, but that is not correct. A BIA, at the minimum, is the equivalent of one task of a risk assessment — asset valuation, a determination of the value of the target body of information and its supporting IT resources. At the most, the BIA will develop the equivalent of a Single

Loss Exposure, with supporting details, of course, usually based on a worst case scenario. The results are most often used to convince management that they should fund development and maintenance of a contingency plan.

Information security is much more than contingency planning. A BIA often requires 75 to 100 percent or more of the work effort (and associated cost) of a risk assessment, while providing only a small fraction of the useful information provided by a risk assessment. A BIA includes little if any vulnerability assessment, and no sound basis for cost/benefit analysis.

Target Audience Concerns

Risk assessment continues to be viewed with skepticism by many in the ranks of management. Yet those for whom a well-executed risk assessment has been done have found the results to be among the most useful analyses ever executed for them.

To cite a few examples:

- In one case, involving an organization with multiple large IT facilities — one of which was particularly vulnerable — a well-executed risk assessment promptly secured the attention of the Executive Committee, which had resisted all previous initiatives to address the issue. Why? Because IT management could not previously supply justifying numbers to support its case. With the risk assessment in hand, IT management got the green light to consolidate IT activities from the highly vulnerable site to another facility with much better security. This was accomplished despite strong union and staff resistance. The move was executed by this highly regulated and bureaucratic organization within three months of the quantitative risk assessment's completion! The quantitative risk assessment provided what was needed, credible facts and numbers of their own.
- In another case, a financial services organization found, as a result of a quantitative risk assessment, that it was carrying four to five times the amount of insurance warranted by its level of exposure. It reduced coverage by half, still retaining a significant cushion, and has since saved hundreds of thousands of dollars in premiums.
- In yet another case, management of a relatively young but rapidly growing organization had maintained a rather "entrepreneurial" attitude toward IT in general, until presented with the results of a risk assessment that gave them a realistic sense of the risks inherent to that posture. Substantial policy changes were made on the spot, and information security began receiving real consideration, not just lip service.
- Finally, a large energy industry organization was considering relocating its IT function from its original facility to a bunkered, tornado-proof facility across town that was being abandoned by a major insurance company. The energy company believed that it could reduce its IT-related risk substantially. The total cost of the move would have run into the millions of dollars. Upon executing a strategic risk assessment for the alternatives, it was found that the old facility was sound, and relocating would not significantly reduce the organizations risk. In fact, it was found that the biggest risks were being taken in its failure to maintain good management practices.

Some specific areas of concern are addressed below.

Diversion of Resources

That organizational staff will have to spend some time providing information for the risk assessment is often a major concern. Regardless of the nature of the assessment, there are two key areas of information gathering that will require staff time and participation beyond that of the person(s) responsible for executing the risk assessment:

1. Valuing the intangible information asset's confidentiality, integrity, and availability
2. Conducting the vulnerability analysis

These tasks will require input from two entirely different sets of people in most cases.

Valuing the Intangible Information Asset

There are a number of approaches to this task, and the amount of time it takes to execute will depend on the approach as well as whether it is qualitative or quantitative. As a general rule of thumb, however, one could expect all but the most cursory qualitative approach to require one to four hours of continuous time from two to five key knowledgeable staff for each intangible information asset valued.

Experience has shown that the Modified Delphi approach is the most efficient, useful, and credible. For detailed guidance, refer to the "Guideline for Information Valuation" (GIV) published by the Information System Security Association (ISSA). This approach will require (typically) the participation of three to five staff members knowledgeable in various aspects of the target information asset. A Modified Delphi meeting routinely lasts four hours; so, for each target information asset, key staff time of 12 to 16 hours will be expended in addition to about 20 to 36 hours total for a meeting facilitator (four hours) and a scribe (16 to 32 hours).

Providing this information has proven to be a valuable exercise for the source participants, and the organization, by giving them significant insight into the real value of the target body of information and the consequences of losing its confidentiality, availability, or integrity. Still, this information alone should not be used to support risk mitigation cost/benefit analysis.

While this "Diversion of Resources" may be viewed initially by management with some trepidation, the results have invariably been judged more than adequately valuable to justify the effort.

Conducting the Vulnerability Analysis

This task, which consists of identifying vulnerabilities, can and should take no more than five workdays (about 40 hours) of one-on-one meetings with staff responsible for managing or administering the controls and associated policy (e.g., logical access controls, contingency planning, change control, etc.). The individual meetings — actually guided interviews, ideally held in the interviewees' workspace — should take no more than a couple of hours. Often, these interviews take as little as five minutes. Collectively, however, the interviewees' total diversion could add up to as much as 40 hours. The interviewer will, of course, spend matching time, hour for hour. This one-on-one approach minimizes disruption while maximizing the integrity of the vulnerability analysis by assuring a consistent level-setting with each interviewee.

Credibility of the Numbers

Twenty years ago, the task of coming up with "credible" numbers for information asset valuation, threat frequency and impact distributions, and other related risk factors was daunting. Since then, the GIV was published, and significant progress has been made by some automated tools' handling of the numbers and their associated knowledge bases — the knowledge bases that were developed on the basis of significant research to establish credible numbers. And, credible results are provided if proven algorithms with which to calculate illustrative risk models are used.

However, manual approaches or automated tools that require the users to develop the necessary quantitative data are susceptible to a much greater degree of subjectivity and poorly informed assumptions.

In the past couple of years, there have been some exploratory efforts to establish a Threat Research Center tasked with researching and establishing:

1. A standard information security threat population
2. Associated threat frequency data
3. Associated threat scenario and impact data

and maintaining that information while assuring sanitized source channels that protect the providers of impact and scenario information from disclosure. As recognition of the need for strong information security and associated risk assessment continues to increase, the pressure to launch this function will eventually be successful.

Subjectivity

The ideal in any analysis or assessment is complete objectivity. Just as there is a complete spectrum from qualitative to quantitative, there is a spectrum from subjective to increasingly objective. As more of the elements of risk are expressed in independently objective terms, the degree of subjectivity is reduced accordingly, and the results have demonstrable credibility.

Conversely, to the extent a methodology depends on opinion, point of view, bias, or ignorance (subjectivity), the results will be of increasingly questionable utility. Management is loath to make budgetary decisions based on risk metrics that express value and risk in terms such as low, medium, and high.

There will always be some degree of subjectivity in assessing risks. However, to the extent that subjectivity is minimized by the use of independently objective metrics, and the biases of tool developers, analysts, and knowledgeable participants are screened, reasonably objective, credible risk modeling is achievable.

Utility of Results

Ultimately, each of the above factors (diversion of resources, credibility of the numbers, subjectivity, and, in addition, timeliness) plays a role in establishing the utility of the results. Utility is often a matter of perception. If management feels that the execution of a risk assessment is diverting resources from its primary mission inappropriately, if the numbers are not credible, if the level of subjectivity exceeds an often intangible cultural threshold for the organization, or if the project simply takes so long that the results are no longer timely, then the attention — and trust — of management will be lost or reduced along with the utility of the results.

A risk assessment executed with the support of contemporary automated tools can be completed in a matter of weeks, not months. Developers of the best automated tools have done significant research into the qualitative elements of good control, and their qualitative vulnerability assessment knowledge bases reflect that fact. The same is true with regard to their quantitative elements. Finally, in building these tools to support quantitative risk assessment, successful efforts have been made to minimize the work necessary to execute a quantitative risk assessment.

The bottom line is that it makes very little sense to execute a risk assessment manually or build one's own automated tool except in the most extraordinary circumstances. A risk assessment project that requires many work-months to complete manually (with virtually no practical "what-if" capability) can, with sound automated tools, be done in a matter of days, or weeks at worst, with credible, useful results.

Tasks of Risk Assessment

In this section, we will explore the classic tasks of risk assessment and key issues associated with each task, regardless of the specific approach to be employed. The focus is, in general, primarily on quantitative methodologies. However, wherever possible, related issues in qualitative methodologies are also discussed.

Project Sizing

In virtually all project methodologies, there are a number of elements to be addressed to ensure that all participants, and the target audience, understand and are in agreement about the project. These elements include:

- Background
- Purpose
- Scope
- Constraints
- Objective
- Responsibilities
- Approach

In most cases, it would not be necessary to discuss these individually, as most are well-understood elements of project methodology in general. In fact, they are mentioned here for the exclusive purpose of pointing out the importance of (1) ensuring that there is agreement between the target audience and those responsible for executing the risk assessment, and (2) describing the constraints on a risk assessment project. While a description of the scope, *what is included*, of a risk assessment project is important, it is equally important to describe specifically, in appropriate terms, *what is not included*. Typically, a risk assessment focuses on a subset of the organization's information assets and control functions. If what is not to be included is not identified, confusion and misunderstanding about the risk assessment's ramifications may result.

Again, the most important point about the project sizing task is to ensure that the project is clearly defined and that a clear understanding of the project by all parties is achieved.

Threat Analysis

In manual approaches and some automated tools, the analyst must determine what threats to consider in a particular risk assessment. Because there is not, at present, a standard threat population and readily available threat statistics, this task can require a considerable research effort. Of even greater concern is the possibility that a significant local threat could be overlooked and associated risks inadvertently accepted. Worse, it is possible that a significant threat is intentionally disregarded.

The best automated tools currently available include a well-researched threat population and associated statistics. Using one of these tools virtually assures that no relevant threat is overlooked, and associated risks are accepted as a consequence.

If, however, a determination has been made not to use one of these leading automated tools and instead to do the threat analysis independently, there are good sources for a number of threats, particularly for all natural disasters, fire, and crime (oddly enough, not so much for computer crime), even falling aircraft. Also, the console log is an excellent source for in-house experience of system development, maintenance, operations, and other events that can be converted into useful threat event statistics with a little tedious review. Finally, in-house physical and logical access logs (assuming such are maintained) can be a good source of related threat event data.

However, gathering this information independently, even for the experienced risk analyst, is no trivial task. Weeks, if not months, of research and calculation will be required, and, without validation, results may be less than credible.

For those determined to proceed independently, the following list of sources, in addition to in-house sources previously mentioned, will be useful:

- Fire — National Fire Protection Association (NFPA)
- Flood, all categories — National Oceanic and Atmospheric Administration (NOAA) and local Flood Control Districts
- Tornado — NOAA
- Hurricane — NOAA and local Flood Control Districts
- Windstorms — NOAA
- Snow — NOAA
- Icing — NOAA
- Earthquakes — U.S. Geological Survey (USGS) and local university geology departments
- Sinkholes — USGS and local university geology departments
- Crime — FBI and local law enforcement statistics, and your own in-house crime experience, if any
- Hardware failures — vendor statistics and in-house records

Until an independent Threats Research Center is established, it will be necessary to rely on automated risk assessment tools, or vendors, or your own research for a good threat population and associated statistics.

Asset Identification and Valuation

While all assets may be valued qualitatively, such an approach is useless if there is a need to make well-founded budgetary decisions. Therefore, this discussion of asset identification and valuation will assume a need for the application of monetary valuation.

There are two general categories of assets relevant to the assessment of risk in the IT environment:

1. Tangible assets
2. Intangible assets

Tangible Assets

The tangible assets include the IT facilities, hardware, media, supplies, documentation, and IT staff budgets that support the storage, processing, and delivery of information to the user community. The value of these assets is readily determined, typically, in terms of the cost of replacing them. If any of these are leased, of course, the replacement cost may be nil, depending on the terms of the lease.

Sources for establishing these values are readily found in the associated asset management groups, that is, facilities management for replacement value of the facilities, hardware management for the replacement value for the hardware — from CPUs to controllers, routers and cabling, annual IT staff budgets for IT staff, etc.

Intangible Assets

The intangible assets, which might be better characterized as information assets, are comprised of two basic categories:

1. Replacement costs for data and software
2. The value of the confidentiality, integrity, and availability of information

Replacement Costs

Developing replacement costs for data is not usually a complicated task unless source documents do not exist or are not backed up reliably at a secure off-site location. The bottom line is that "x" amount of data represents "y" keystrokes — a time-consuming but readily measurable manual key entry process.

Conceivably, source documents can now be electronically "scanned" to recover lost, electronically stored data. Clearly, scanning is a more efficient process, but it is still time-consuming. However, if neither source documents nor off-site backups exist, actual replacement may become virtually impossible, and the organization faces the question of whether such a condition can be tolerated. If, in the course of the assessment, this condition is found, the real issue is that the information is no longer available, and a determination must be made as to whether such a condition can be overcome without bankrupting the private-sector organization or irrevocably compromising a government mission.

Value of Confidentiality, Integrity, and Availability

In recent years, a better understanding of the values of confidentiality, integrity, and availability and how to establish these values on a monetary basis with reasonable credibility has been achieved. That understanding is best reflected in the ISSA-published GIV referenced above. These values often represent the most significant "at-risk" asset in IT environments. When an organization is deprived of one or more of these with regard to its business or mission information, depending on the nature of that business or mission, there is a very real chance that unacceptable loss will be incurred within a relatively short time.

For example, it is well-accepted that a bank that loses access to its business information (loss of availability) for more than a few days is very likely to go bankrupt.

A brief explanation of each of these three critical values for information is presented below.

1. *Confidentiality.* Confidentiality is lost or compromised when information is disclosed to parties other than those authorized to have access to the information. In the complex world of IT today, there are many ways for a person to access information without proper authorization if appropriate controls are not in place. Without appropriate controls, that access or theft of information could be accomplished without a trace. Of course, it still remains possible to simply pick up and walk away with confidential documents carelessly left lying about or displayed on an unattended, unsecured PC.
2. *Integrity.* Integrity is the condition that information in or produced by the IT environment accurately reflects the source or process it represents. Integrity can be compromised in many ways, from data entry errors to software errors to intentional modification. Integrity may be thoroughly compromised, for example, by simply contaminating the account numbers of a bank's demand deposit records. Because the account numbers are a primary reference for all associated data, the information is effectively no longer available. There has been a great deal of discussion about the nature of integrity. Technically, if a single character is wrong in a file with millions of records, the file's integrity has been compromised.

   Realistically, however, some expected degree of integrity must be established. In an address file, 99 percent accuracy (only one out of 100 is wrong) may be acceptable. However, in the same file, if each record of 100 characters had only one character wrong — in the account number — the records would meet the poorly articulated 99 percent accuracy standard, but be completely compromised. In other words, the loss of integrity can have consequences that range from trivial to catastrophic. Of course, in a bank with one million clients, 99 percent accuracy means at best that the records of 10,000 clients are in error. In a hospital, even one such error could lead to loss of life!
3. *Availability.* Availability, the condition that electronically stored information is where it needs to be, when it needs to be there, and in the form necessary, is closely related to the availability of the information processing technology. Whether because the process is unavailable, or the information itself is somehow unavailable, makes no difference to the organization dependent on the information to conduct its business or mission. The value of the information's availability is reflected in the costs incurred, over time, by the organization, because the information was not available, regardless of cause. A useful tool (from the Modified Delphi method) for capturing the value of availability, and articulating uncertainty, is illustrated in Exhibit 67.3. This chart represents the cumulative cost, over time, of the best-case and worst-case scenarios, with confidence factors, for the loss of availability of a specific information asset.

INTERVAL	LOS	HI$	CF %	INTERVAL	LOS	HI$	CF %
0-1 HR				4 DAYS			
2 HR				8 DAYS			
4 HR				16 DAYS			
8 HR				1 MONTH			
16 HR				2 MONTHS			
1 DAY				3 MONTHS			
2 DAY				6 MONTHS			

EXHIBIT 67.3 Capturing the value of availability (Modified Delphi method).

Vulnerability Analysis

This task consists of the identification of vulnerabilities that would allow threats to occur with greater frequency, greater impact, or both. For maximum utility, this task is best conducted as a series of one-on-one interviews with individual staff members responsible for developing or implementing organizational policy through the management and administration of controls. To maximize consistency and thoroughness, and to minimize subjectivity, the vulnerability analysis should be conducted by an interviewer who guides each interviewee through a well-researched series of questions designed to ferret out all potentially significant vulnerabilities.

It should be noted that establishment and global acceptance of Generally Accepted System Security Principles (GASSP), as recommended in the National Research Council report "Computers at Risk" (December 1990), the National Information Infrastructure Task Force (NIITF) findings, the Presidential National Security and Telecommunications Advisory Council (NSTAC) report (December 1996), and the President's Commission on Critical Infrastructure Protection (PCCIP) report (October 1997), all of which were populated with a strong private sector representation, will go far in establishing a globally accepted knowledge base for this task. The "Treadwell Commission" report published by the American Institute of Certified Public Accountants (AICPA) Committee of Sponsoring Organizations (COSO) in 1994, "Internal Control, Integrated Framework" now specifically requires that auditors verify that subject organizations assess and manage the risks associated with IT and other significant organizational resources. The guiding model characterized in the requirement represents quantitative risk assessment. Failure to have effectively implemented such a risk management mechanism now results in a derogatory audit finding.

Threat/Vulnerability/Asset Mapping

Without connecting — mapping — threats to vulnerabilities and vulnerabilities to assets and establishing a consistent way of measuring the consequences of their interrelationships, it becomes nearly impossible to establish the ramifications of vulnerabilities in a useful manner. Of course, intuition and common sense are useful, but how does one measure the risk and support good budgetary management and cost/benefit analysis when the rationale is so abstract?

For example, it is only good common sense to have logical access control, but how does one justify the expense? I am reminded of a major bank whose management, in a cost-cutting frenzy, came very close to terminating its entire logical access control program! With risk assessment, one can show the expected risk and annualized asset loss/probability coordinates that reflect the ramifications of a wide array of vulnerabilities. Exhibit 67.4 carries the illustration further with two basic vulnerabilities.

Applying some simple logic at this point will give the reader some insight into the relationships between vulnerabilities, threats, and potentially affected assets.

No Logical Access Control

Not having logical access control means that anyone can sign on the system, get to any information they wish, and do anything they wish with the information. Most tangible assets are not at risk. However, if IT staff productivity is regarded as an asset, as reflected by their annual budget, that asset could suffer a loss (of productivity) while the staff strives to reconstruct or replace damaged software or data. Also, if confidentiality is compromised by the disclosure of sensitive information (competitive strategies or client information), substantial competitive advantage and associated revenues could be lost, or liability suits for disclosure of private information could be very costly. Both could cause company goodwill to suffer a loss.

VULNERABILITY	MAPPED THREAT(S)	AFFECTED ASSETS (At minimum)[a]
No Logical Access Control	Sabotage of Software	Software Goodwill
	Sabotage of Data/Information	Information Integrity Goodwill
	Theft of Software	Software Goodwill
	Theft of Data/Information	Information Confidentiality Goodwill
	Destruction of Software	Software Goodwill
	Destruction of Data/Information	Information Availability Goodwill
No Contingency Plan	Fire Hurricane Earthquake Flood Terrorist Attack	Facilities Hardware Media and Supplies IT Staff Budgets Software Information Availability Goodwill
	Toxic Contamination[b]	IT Staff Budgets Software Information Availability Goodwill

[a] In each case it is assumed that the indicated vulnerability is the only vulnerability; thus, any impact on other information assets is expected to be insignificant. Otherwise, without current backups, for example, virtually every threat on this chart could have a significant impact on information availability.

[b] Tangible assets are not shown as being impacted by a toxic contamination, aside from the IT staff budgets, because it is assumed that the toxic contamination can be cleaned up and the facilities and equipment restored to productive use.

EXHIBIT 67.4 Two basic vulnerabilities.

Because the only indicated vulnerability is not having logical access, it is reasonable to assume monetary loss resulting from damage to the integrity of the information or the temporary loss of availability of the information is limited to the time and resources needed to recover with well-secured, off-site backups. Therefore, it is reasonable to conclude, all other safeguards being effectively in place, that the greatest exposure resulting from not having logical access control is the damage that may result from a loss of confidentiality for a single event. But without logical access control, there could be many such events!

What if there was another vulnerability? What if the information was not being backed up effectively? What if there were no usable backups? The loss of availability — for a single event — could become overwhelmingly expensive, forcing the organization into bankruptcy or compromising a government mission.

No Contingency Plan

Not having an effective contingency plan means that the response to any natural or man-made disaster will be without prior planning or arrangements. Thus, the expense associated with the event is not assuredly contained to a previously established maximum acceptable loss. The event may very well bankrupt the organization or compromise a government mission. This is without considering the losses associated with the tangible assets! Studies have found that organizations hit by a disaster and not having a good contingency plan are likely (4 out of 5) to be out of business within two years of the disaster event.

What if there were no usable backups — another vulnerability? The consequences of the loss of information availability would almost certainly be made much worse, and recovery, if possible, would be much more costly. The probability of being forced into bankruptcy is much higher.

By mapping vulnerabilities to threats to assets, we can see the interplay among them and understand a fundamental concept of risk assessment:

Vulnerabilities allow threats to occur with greater frequency or greater impact. Intuitively, it can be seen that the more vulnerabilities there are, the greater is the risk of loss.

Risk Metrics/Modeling

There are a number of ways to portray risk: some qualitative, some quantitative, and some more effective than others.

In general, the objective of risk modeling is to convey to decision makers a credible, usable portrayal of the risks associated with the IT environment, answering (again) these questions:

- What could happen? (threat event)
- How bad would it be? (impact)
- How often might it occur? (frequency)
- How certain are the answers to the first three questions? (uncertainty)

With such risk modeling, decision-makers are on their way to making well-informed decisions — either to accept, mitigate, or transfer associated risk.

The following brief discussion of the two general categories of approach to these questions, qualitative and quantitative, will give the reader a degree of insight into the ramifications of using one or the other approach:

Qualitative

The definitive characteristic of the qualitative approach is the use of metrics that are subjective, such as ordinal ranking — low, medium, high, etc. (see Exhibit 67.5). In other words, independently objective values such as objectively established monetary value, and recorded history of threat event occurrence (frequency) are not used.

Quantitative

The definitive characteristic of quantitative approaches is the use of independently objective metrics and significant consideration given to minimizing the subjectivity that is inherent in any risk assessment. Exhibit 67.6 was produced from a leading automated tool, BDSS™, and illustrates quantitative risk modeling.

The graph shown in Exhibit 67.6 reflects the integrated "all threats" risk that is generated to illustrate the results of risk evaluation in BDSS™ before any risk mitigation. The combined value of the tangible and intangible assets at risk is represented on the "Y" axis, and the probability of financial loss is represented on the "X" axis. Thus, reading this graphic model, there is a 1/10 chance of losing about $0.5M over a one-year period.

The graph shown in Exhibit 67.7 reflects the same environment after risk mitigation and associated cost/benefit analysis. The original risk curve (Exhibit 67.6) is shown in Exhibit 67.7 with the reduced risk curve and associated average annual cost of all recommended safeguards superimposed on it, so the reader can see the risk before risk mitigation, the expected reduction in risk, and the cost to achieve it. In Exhibit 67.7, the risk at 1/10 and 1/100 chance of loss is now minimal, and the risk at 1/1000 chance of loss has been reduced from about $2.0M to about $0.3M. The suggested safeguards are thus shown to be well justified.

Management Involvement and Guidance

Organizational culture plays a key role in determining, first, whether to assess risk, and second, whether to use qualitative or quantitative approaches. Many firms' management organizations see themselves as "entrepreneurial" and have an aggressive bottom-line culture. Their basic attitude is to minimize all costs, take the chance that nothing horrendous happens, and assume they can deal with it if it does happen.

		Value		
		Low	Medium	High
Risk	Low			
	Medium			
	High			

EXHIBIT 67.5 Value of the availability of information and the associated risk.

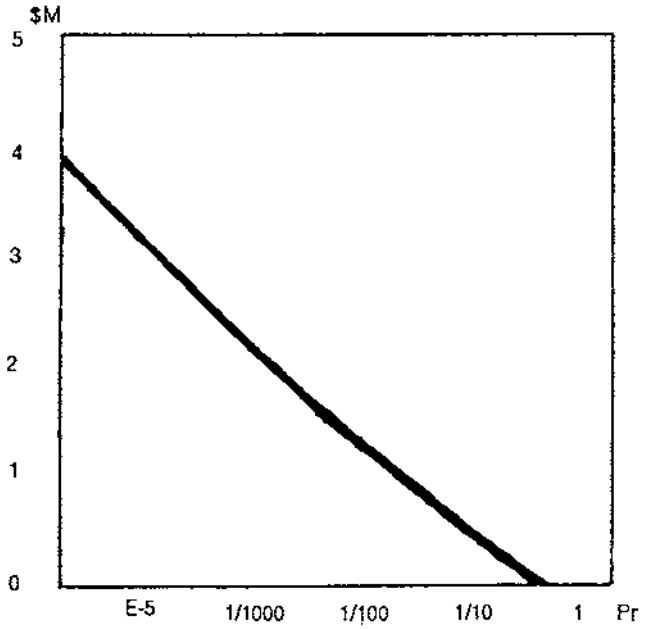


EXHIBIT 67.6 Results of risk evaluation in BDSS™ before any risk mitigation.

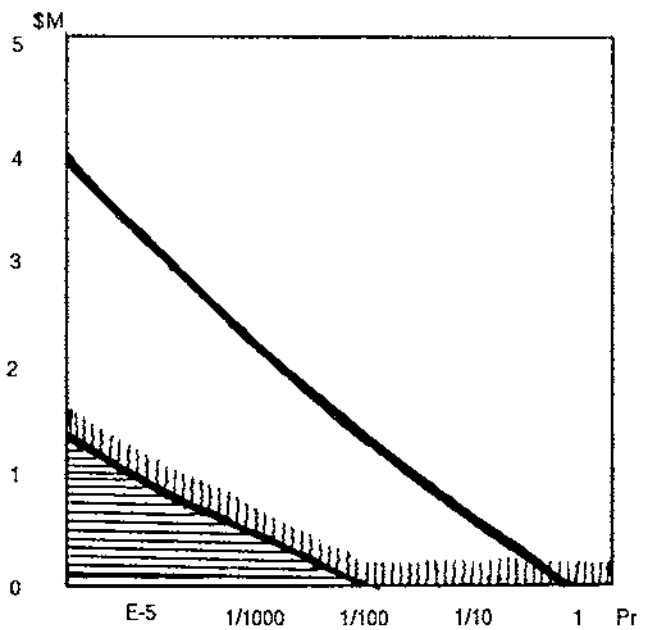


EXHIBIT 67.7 Results of risk evaluation after risk mitigation and associated cost/benefit analysis.

Other firms, particularly larger, more mature organizations, will be more interested in a replicable process that puts results in management language such as monetary terms, cost/benefit assessment, and expected loss. Terms that are understood by business management will facilitate the creation of effective communication channels and support sound budgetary planning for information risk management.

It is very useful to understand the organizational culture when attempting to plan for a risk assessment and get necessary management support. While a quantitative approach will provide, generally speaking, much more useful information, the culture may not be ready to assess risk in significant depth.

In any case, with the involvement, support, and guidance of management, more utility will be gained from the risk assessment, regardless of its qualitative or quantitative nature. And, as management gains understanding of the concepts and issues of risk assessment and begins to realize the value to be gained, reservations about quantitative approaches will diminish, and it will increasingly look toward those quantitative approaches to provide more credible, defensible budgetary support.

Risk Mitigation Analysis

With the completion of the risk modeling and associated report on the observed status of information security and related issues, management will almost certainly find some areas of risk that it is unwilling to accept and for which it wishes to see a proposed risk mitigation analysis. In other words, management will want answers to the last three questions for those unacceptable risks:

1. What can be done?
2. How much will it cost?
3. Is it cost effective?

There are three steps in this process:

1. Safeguard analysis and expected risk mitigation
2. Safeguard costing
3. Safeguard cost/benefit analysis

Safeguard Analysis and Expected Risk Mitigation

With guidance from the results of the risk evaluation, including modeling and associated data collection tasks, and reflecting management concerns, the analyst will seek to identify and apply safeguards that could be expected to mitigate the vulnerabilities of greatest concern to management. Management will, of course, be most concerned about those vulnerabilities that could allow the greatest loss expectancies for one or more threats, or those subject to regulatory or contractual compliance. The analyst, to do this step manually, must first select appropriate safeguards for each targeted vulnerability; second, map or confirm mapping, safeguard/vulnerability pairs to all related threats; and third, determine, for each threat, the extent of asset risk mitigation to be achieved by applying the safeguard. In other words, for each affected threat, determine whether the selected safeguard(s) will reduce threat frequency, reduce threat exposure factors, or both, and to what degree.

Done manually, this step will consume many days or weeks of tedious work effort. Any "what-if" assessment will be very time-consuming as well. When this step is executed with the support of a knowledge-based expert automated tool, however, only a few hours to a couple of days are expended, at most.

Safeguard Costing

To perform a useful cost/benefit analysis, estimated costs for all suggested safeguards must be developed. While these cost estimates should be reasonably accurate, it is not necessary that they be precise. However, if one is to err at this point, it is better to overstate costs. Then, as bids or detailed cost proposals come in, it is more likely that cost/benefit analysis results, as shown below, will not overstate the benefit.

There are two basic categories of costing for safeguards:

1. Cost per square foot, installed
2. Time and materials

In both cases, the expected life and annual maintenance costs must be included to get the average annual cost over the life of the safeguard. An example of each is provided in Exhibits 67.8 and Exhibit 67.9.

These average annual costs represent the break-even point for safeguard cost/benefit assessment for each safeguard. In these examples, discrete, single-point values have been used to simplify the illustration. At least one of the leading automated risk assessment tools, BDSS™, allows the analyst to input bounded distributions with associated confidence factors to articulate explicitly the uncertainty of the values for these preliminary cost estimates. These bounded distributions with confidence factors facilitate the best use of optimal probabilistic analysis algorithms.

EXHIBIT 67.8 Cost per Square Foot, Installed, for a Robust New IT Facility

Cost per square foot	$165.00	
Total Square feet	50,000	
Total		$8,250,000
Safeguard Life expectancy	10 years	
Annualized cost (8,250,000/10)		$825,000
Annual Maintenance	$250,000	
Average Annual Cost		$1,075,000

Safeguard Cost/Benefit Analysis

The risk assessment is now almost complete, though this final set of calculations is, once again, not trivial. In previous steps, the expected value of risk mitigation — the annualized loss expectancy (ALE) before safeguards are applied, less the ALE after safeguards are applied, less the average annual costs of the applied safeguards — is conservatively represented individually, safeguard by safeguard, and collectively. The collective safeguard cost/benefit is represented first, threat by threat with applicable selected safeguards; and, second, showing the overall integrated risk for all threats with all selected safeguards applied. This may be illustrated as follows:

$$\text{Safeguard } 1 \rightarrow \text{Vulnerability } 1 \rightarrow n \rightarrow \text{Threat } 1 \rightarrow n$$

One safeguard may mitigate one or more vulnerabilities to one or more threats. A generalization of each of the three levels of calculation is represented below.

For the Single Safeguard

A single safeguard may act to mitigate risk for a number of threats. For example, a contingency plan will contain the loss for disasters by facilitating a timely recovery. The necessary calculation includes the integration of all affected threats' risk models *before* the safeguard is applied, less their integration *after* the safeguard is applied to define the gross risk reduction benefit. Finally, subtract the safeguard's average annual cost to derive the net annual benefit.

$$[(\text{RB(T)1 RA(T)1}) - (\text{RB(T)n RA(T)n}) = \text{GRRB}] \text{ — SGAAC} = \text{NRRB}$$

where:

RB(T) = the risk model for threats1-n *before* the safeguard is applied
RA(T) = the risk model for threats1-n *after* the safeguard is applied
GRRB = Gross Risk Reduction Benefit
NRRB = Net Risk Reduction Benefit
SGAAC = Safeguard Average Annual Cost

This information is useful in determining whether individual safeguards are cost effective. If the net risk reduction (mitigation) benefit is negative, the benefit is negative (i.e., not cost effective).

For the Single Threat

Any number of safeguards may act to mitigate risk for any number of threats. It is useful to determine, for each threat, how much the risk for that threat was mitigated by the collective population of safeguards selected that act to mitigate the risk for the threat. Recognize at the same time that one or more of these safeguards can also act to mitigate the risk for one or more other threats.

$$[(\text{AALEB} - \text{AALEA} = \text{GRRB}) - \text{SGAACSG1–n}] = \text{NRRB}$$

Cost per labor hour	$65.00	
Labor hours	480	
Implementation cost, labor		$31,200
Purchase/materials for an automated DRP tool	$29,000	
Total acquisition and implementation cost		$70,200
Safeguard life expectancy	8 years	
Annualized acquisition and implementation cost ($70,200/8)		$8,775
Annual maintenance:	$4,350	
DRP license maintenance	$32,500	
DRP staff, .5 work year (65,000 x .5)		$36,850
Average Annual Cost		$45,625

EXHIBIT 67.9 Time and materials for acquiring and implementing a disaster recovery plan (DRP).

where:

AALEB = Average Annual Loss Expectancy *before* safeguards

AALEA = Average Annual Loss Expectancy *after* safeguards

In this case, NRRB refers to the combined benefit of the collective population of safeguards selected for a specific threat. This process should be executed for each threat addressed. Still, these two processes alone should not be regarded as definitive decision support information. There remains the very real condition that the collective population of safeguards could mitigate risk very effectively for one major threat while having only a minor risk mitigating effect for a number of other threats relative to their collective SGAAC.

In other words, if looked at out of context, the selected safeguards could appear, for those marginally affected risks, to be cost prohibitive — their costs may exceed their benefit for those threats. Therefore, the next process is essential to an objective assessment of the selected safeguards overall benefits.

For All Threats

The integration of all individual threat risk models for *before* selected safeguards are applied and for *after* selected safeguards are applied shows the gross risk reduction benefit for the collective population of selected safeguards as a whole. Subtract the average annual cost of the selected safeguards, and the net risk reduction benefit as a whole is established.

This calculation will generate a single risk model that accurately represents the combined effect of all selected safeguards in mitigating risk for the array of affected threats. In other words, an executive summary of the expected results of proposed risk-mitigating measures is generated.

Final Recommendations

After the risk assessment is complete, final recommendations should be prepared on two levels: (1) a categorical set of recommendations in an executive summary, and (2) detailed recommendations in the body of the risk assessment report. The executive summary recommendations are supported by the integrated risk model reflecting all threats risks before and after selected safeguards are applied, the average annual cost of the selected safeguards, and their expected risk mitigation benefit.

The detailed recommendations should include a description of each selected safeguard and its supporting cost benefit analysis. Detailed recommendations may also include an implementation plan. However, in most cases, implementation plans are not developed as part of the risk assessment report. Implementation plans are typically developed upon executive endorsement of specific recommendations.

Automated Tools

The following products represent a broad spectrum of automated risk assessment tools ranging from the comprehensive, knowledge based expert system BDSS™, to RiskCalc, a simple risk assessment shell with provision for user-generated algorithms and a framework for data collection and mapping.

- ARES: Air Force Communications and Computer Security Management Office, Kelly AFB, TX
- @RISK: Palisade Corp, Newfield, NY
- Bayesian Decision Support System (BDSS™), OPA, Inc.: The Integrated Risk Management Group, Petaluma, CA
- Control Matrix Methodology for Microcomputers: Jerry FitzGerald & Associates, Redwood City, CA
- COSSAC: Computer Protection Systems Inc., Plymouth, MI
- CRITI-CALC: International Security Technology, Reston, VA
- CRAMM: Executive Resources Association, Arlington, VA
- GRA/SYS: Nander Brown & Co., Reston, VA
- IST/RAMP: International Security Technology, Reston, VA
- JANBER: Eagon. McAllister Associates Inc., Lexington Park, MD
- LAVA: Los Alamos National Laboratory, Los Alamos, NM
- LRAM: Livermore National Laboratory, Livermore, CA

- MARION: Coopers & Lybrand (UK-based), London, England
- Micro Secure Self Assessment: Boden Associates, East Williston, NY
- Predictor: Concorde Group International, Westport, CT
- PRISM: Palisade Corp., Newfield, NY
- QuikRisk: Basic Data Systems, Rockville, MD
- RA/SYS: Nander Brown & Co., Reston, VA
- RANK-IT: Jerry FitzGerald & Associates, Redwood City, CA
- RISKCALC: Hoffman Business Associates Inc., Bethesda, MD
- RISKPAC: Profile Assessment Corp., Ridgefield, CT
- RISKWATCH: Expert Systems Software Inc., Long Beach, CA
- The Buddy System Risk Assessment and Management System for Microcomputers: Countermeasures, Inc., Hollywood, MD

Summary

While the dialogue on risk assessment continues, management increasingly is finding utility in the technology of risk assessment. Readers should, if possible, given the culture of their organization, make every effort to assess the risks in the subject IT environments using automated, quantitatively oriented tools. If there is strong resistance to using quantitative tools, then proceed with an initial approach using a qualitative tool. But do start the risk assessment process!

Work on automated tools continues to improve the utility and credibility. More and more of the "Big Accounting Firms" and other major consultancies, including those in the insurance industry, are offering risk assessment services using, or planning to use, quantitative tools. Managing risk is the central issue of information security. Risk assessment with automated tools provides organizational management with sound insight into their risks and how best to manage them and reduce liability cost effectively.

MANAGING RISK IN AN INTRANET ENVIRONMENT

Ralph L. Kliem

INSIDE

Types of Risks, Risk Management Concepts, Identifying Risks, Analyzing Risks, Controlling Risks

INTRODUCTION

The rush to adopt intranet technology keeps growing daily. It is not hard to understand the enthusiastic embrace of this new technology. It is, quite frankly, quite inviting. It provides many advantages, especially when compared with the rigid, complex technology of the past. It builds on the existing client/server or distributed systems environment. It provides a convenient means to access and distribute information throughout an enterprise. Users find it easy to enter and navigate. It encourages a truly open computing environment. It enables easier distribution of applications. The advantages go on and on. It seems something akin to a perpetual motion machine. It is just too good to be true.

All these advantages can prove beguiling; many companies are finding that the intranet is too good to be true. As they embrace this technology, many companies are finding that they have more of a perpetual problem machine than one of perpetual motion. This is especially the case when they fail to prepare themselves in advance for the new technology. What is happening, of course, is that many companies are finding that they must deal with issues pertaining to organizational structuring, internal and external access to data, copyright protection, data ownership and maintenance, configuration of hardware and software, traffic management, and many others.

PAYOFF IDEA

Intranets are flexible systems that can have enterprisewide reach. Because of their scale and accessibility, intranets pose risks beyond the prominent one of security. Performance, integration, scalability, and planning are also risks that systems development managers must face when dealing with intranets. This article shows how to identify, analyze, and control risk in an intranet environment.

GROWING RISK

Many intranets are like some mystic poltergeist, lacking any structure, purpose, or boundary. Yet, the posi-

Auerbach Publications

tive and negative benefits of going the intranet route remain untested despite the history of its sister technology, the Internet.

As the intranet becomes more pervasive and complex, the opportunities for vulnerabilities increase. With these vulnerabilities comes risk. Many companies have implemented intranets, for example, without any thought about standards or policies on access, content, or use. Their oversight or deliberate neglect appears acceptable to them, reflecting a willingness to face the consequences if something goes awry.

Part of the problem is that many industries across the United States are willing to accept a certain level of risk as a tradeoff for realizing short- and long-term gains in productivity. Another contributor to the problem is that risk is often narrowly construed as being only security. In reality, a security risk — albeit important — is just one of the many types of risks facing an intranet. Many corporations find themselves facing a host of unanticipated risks related to transaction security, network capacity, configuration control, directory services, maintenance skills availability, upgrades to hardware, and backup procedures. Other intranet-related risks include performance, integration, scalability, and planning.

The risks tend to multiply as the size of, complexity of, and level of reliance on the intranet grows. Once an intranet gains momentum within an organization, it is very difficult to avoid fighting fires. The only mechanism to deal with such an environment is to perform risk management as early as possible, preferably before the intranet is up and running.

RISK MANAGEMENT CONCEPTS

Before discussing the specific types of risks facing an intranet, however, it is important to understand some general concepts about risk management. Risk is the occurrence of an event that has consequences. A vulnerability, or exposure, is a weakness that enables a risk to have an impact. The idea is to institute controls that will prevent, detect, or correct impacts from risks.

Risk management is the entire process of managing risk. It consists of three closely related actions:

- Risk identification
- Risk analysis
- Risk control

Risk identification is identifying the risks confronting a system. Risk analysis is analyzing data collected using a particular technique. Risk control is identifying and verifying the existence of measures to lessen or avoid the impact of a risk. Risk control may involve avoiding, accepting, adopting, or transferring risk. The measures in place to prevent, detect, or correct are called controls.

Risk management for an intranet offers several advantages. It identifies the most likely and most important risks facing an intranet. It enables taking a proactive approach when managing the intranet, such as identifying assets that need augmentation or improvement. It provides an opportunity to define exactly what constitutes an intranet within a company. It enables building an infrastructure to support the overall business objectives that the intranet is to help achieve. It identifies where to focus energies. Finally, it provides the material to develop contingency plans to respond appropriately to certain risks, if and when they do arise.

Of course, it makes sense to do risk assessment as early as possible. It enables identifying control weaknesses before an intranet is implemented and, therefore, institutionalizes them. It allows incorporating better controls when it is cheaper to make the appropriate changes rather than when the intranet is up and running. Finally, it gives everyone a sense of confidence early on that they are using a secure, reliable, well-managed system.

RISK IDENTIFICATION

For an intranet, the risks are innumerable, especially since the technology is new and has been adopted rapidly. Its growth has been so dramatic that a complete listing would be akin to trying to calculate the end of infinity. It impacts both functions and processes within an organization to such an extent that listing all the risks would prove futile. It is possible, however, to categorize the risks according to some arbitrary but generic criteria. Intranet risks can fall into four basic categories: personnel, operational, economic, and technological.

Personnel risks deal with the human side of an intranet. Some examples are:

- Inadequate training of users
- Lack of available skills for intranet development and maintenance
- Lack of available skills for intranet publishing and design
- Lack of available skills for systems administration
- Poor role definition for data content, usage, and maintenance
- Unclear responsibilities for dealing with traffic flow problems

Operational risks deal with business processes. A process transcends a functional entity (e.g., department) within an organization, receives input, and transforms it into output. Some examples are:

- Inadequate capability to find data
- Inadequate presentation of data
- Lack of backup and recovery procedures
- Not adequately controlling access to sensitive data
- Poor directory services

- Poor integration with legacy systems
- Poor online service support
- Poorly maintained links
- Transferring sensitive data over a network with poor security
- Uncontrolled access to unauthorized sites
- Unexpected rise in network traffic

Economic risks relate to the costs of an intranet — from development to ongoing operation. Some examples are excessive or out-of-the-ordinary costs related to:

- Internet service provider services
- Hardware upgrades
- Software upgrades
- Integration of components (e.g., desktops, server applications)
- Integration of applications with legacy systems and databases
- Labor for developing and maintaining the infrastructure (e.g., administering the site)

Technological risks deal with the hardware, software, and other media that form an intranet. Some examples are:

- Immaturity of the technology being employed
- Inadequate communications hardware and software
- Inadequate system hardware and software
- Insufficient availability of network bandwidth
- Poor availability of development and publishing tools
- Poor configuration control of clients
- Poor integration of components (e.g., local area networks, server applications)
- Poor retrieval tools and services
- Slow connection
- Unreliable server hardware and software

It would be a mistake, however, to think that these four categories are mutually exclusive.

Deciding what risks fall within each category is often a judgment call and is mainly academic. The key is to use the categories to identify the risks, determine their relative importance to one another, and recognize the controls that do or should exist.

RISK ANALYSIS

After identifying the risks, the next action is to determine their relative importance to one another and their probability of occurrence. The rank-

EXHIBIT 1 — An Ordered Listing of Intranet Risks

Risk	Probability of Occurrence	Impact
Lack of available skills for system administration	High	Major
Uncontrollable access to unauthorized sites	High	Minor
Poor integration of components (e.g., local area networks, applications)	Low	Minor
Unexpected network utilization costs	High	Major

ing of importance depends largely on the purpose management has established for the intranet. In other words, what business value is the intranet supposed to provide? In what ways is the intranet supposed to serve the interests of its users?

There are multiple approaches to analyzing risk. Basically, the approaches fall into three categories:

- Quantitative
- Qualitative
- A combination of both

Qualitative risk analysis relies on mathematical calculations to determine a risk's relative importance to another and its probability of occurrence. The Monte Carlo simulation technique falls within this category.

Qualitative risk analysis relies less on mathematical calculations and more on judgmental considerations to determine a risk's relative importance to another and probability of occurrence. Heuristics, or rules of thumb, fall within this category.

A combination of the two, of course, uses both mathematical and qualitative considerations to determine a risk's relative importance to another and its probability of occurrence. The precedence diagramming method, which uses an ordinal approach to determine priorities according to some criterion, falls within this category. Regardless of the approach, a resulting rank order listing of risks is shown in Exhibit 1.

RISK CONTROL

With the analysis complete, the next action is to identify controls that should exist to prevent, detect, or correct the impact of risks. Risk control involves a painstaking effort to understand the environment where the intranet finds itself. It means looking at a host of factors, such as:

- Applications at the client and server levels
- Architectural design of the network
- Availability of expertise
- Content and structure in databases (e.g., images, text)

- Current network capacity
- Degree of integration among system components
- Firewall protection
- Hardware components
- Importance of copyright issues
- Level of anticipated network traffic in the future
- Level of financial resources available for ongoing maintenance
- Level of security requirements
- Number of mission-critical systems depending on the intranet
- Sensitivity of data being accessed and transported
- Software components

After identifying the controls that should be in place, the next action is to verify whether they are actually in place to prevent, detect, or correct. Preventive controls mitigate or stop an event that exploits the vulnerabilities of a system. Detective controls disclose the occurrence of an event that exploited a vulnerability. Corrective controls counteract the effects of an event and preclude similar exploitation in the future.

To determine the types of controls that are in place requires painstaking "leg work," often achieved through interviews, literature reviews, and a thorough knowledge of the major components of the intranet. The result is the identification of what controls do exist and which ones are lacking or need improvement.

There are many preventive, detective, and corrective controls to apply in an intranet environment. These include:

- Adequate backup and recovery to safeguard data
- Adequate, relevant, and timely training for users and developers
- Changing passwords
- Documented and followed policies and procedures
- Metrics to ensure goals and objectives are being achieved
- Monitoring of network utilization regarding traffic flow and data content
- Monitoring system performance
- Restricting user access to specific server applications and databases
- Restricting user privileges
- Security for sensitive data and transactions
- Segregation of duties, such as reviews and approvals
- Setting up a firewall
- Tracking of hardware and software
- Tracking user access
- Upgrading hardware and software

Armed with a good idea of the type and nature of the risks confronting an intranet, the next step is to make improvements. This involves

EXHIBIT 2 — Intranet Risks and Their Controls

Risk	Control
Lack of available skills for system administration	• Cross-training • Outsourcing
Uncontrolled access to sensitive databases	• Restrictive access policies • Firewall
Poor integration of components (e.g., local area networks, server applications)	• Client and server configuration guidelines and standards
Unexpected network utilization costs	• Periodic network capacity planning • Limiting nonessential access during high peak periods

strengthening or adding controls. It means deciding whether to accept, avoid, adopt, or transfer risk. To accept a risk means letting it occur and taking no action. An example is not doing anything about external breach to the intranet. To avoid a risk means taking action to not confront a risk. An example is continuing to expand bandwidth without considering the causes (such as surfing). Adopting means living with a risk and dealing with it by working "around it." An example is waiting until a later time to access the network when usage is less. Transfer means shifting a risk over to someone else or some other organization. An example is having the user assume responsibility for accessing and displaying proprietary data. Exhibit 2 presents some examples of controls that may be taken for selected types of risks in an intranet environment.

CONCLUSION

The advantages of performing risk management for an intranet are quite obvious. Yet, the lure of the technology is so inviting that even the thought of doing any risk assessment appears more like an administrative burden. The decision to manage risk depends on the answers to two key questions: Do the advantages of not bothering to identify, analyze, and control risks exceed not doing it? Are you willing to accept the consequences if a vulnerability is taken advantage of, either deliberately or by accident? In the end, the decision to manage risk is, ironically, one of risk.

Ralph L. Kliem is president of Practical Creative Solutions, Inc., a Redmond, WA consulting and training firm. He is the co-author of *Just-in-Time Systems for Computing Environments* (published by Quorum) and *Reducing Project Risk* (published by Gower). He can be reached at 75377.2623@compuserv.com, or by phone (425) 556-9589.

68

Security Assessment

Sudhanshu Kairab, CISSP, CISA

During the past decade, businesses have become increasingly dependent on technology. IT environments have evolved from mainframes running selected applications and independent desktop computers to complex client/server networks running a multitude of operating systems with connectivity to business partners and consumers. Technology trends indicate that IT environments will continue to become more complicated and connected.

With this trend in technology, why is security important? With advances in technology, security has become a central part of strategies to deploy and maintain technology. For companies pursuing E-commerce initiatives, security is a key consideration in developing the strategy. In the business-to-consumer markets, customers cite security as the main reason for buying or not buying online. In addition, most of the critical data resides on various systems within the IT environment of most companies. Loss or corruption of data can have devastating effects on a company, ranging from regulatory penalties stemming from laws such as HIPAA (Health Insurance Portability and Accountability Act) to loss of customer confidence.

In evaluating security in a company, it is important to keep in mind that managing security is a process much like any other process in a company. Like any other business process, security has certain technologies that support it. In the same way that an ERP (enterprise resources planning) package supports various supply-chain business processes such as procurement, manufacturing, etc., technologies such as firewalls, intrusion detection systems, etc. support the security process. However, unlike some other business processes, security is something that touches virtually every part of the business, from human resources and finance to core operations. Consequently, security must be looked at as a business process and not a set of tools. The best security technology will not yield a secure environment if it is without sound processes and properly defined business requirements. One of the issues in companies today is that, as they have raced to address the numerous security concerns, security processes and technology have not always been implemented with the full understanding of the business and, as a result, have not always been aligned with the needs of the business.

When securing a company's environment, management must consider several things. In deciding what security measures are appropriate, some considerations include:

- What needs to be protected?
- How valuable is it?
- How much does downtime cost a company?
- Are there regulatory concerns (e.g., HIPAA, GLBA [Gramm–Leach–Bliley Act])?
- What is the potential damage to the company's reputation if there is a security breach?
- What is the probability that a breach can occur?

Depending on the answers to these and other questions, a company can decide which security processes make good business sense for them. The security posture must balance:

- The security needs of the business
- The operational concerns of the business
- The financial constraints of the business

The answers to the questions stated earlier can be ascertained by performing a security assessment. An independent third-party security assessment can help a company define what its security needs are and provide a framework for enhancing and developing its information security program. Like an audit, it is important for an assessment to be independent so that results are not (or do not have the appearance of being) biased in any way. An independent security assessment using an internal auditor or a third-party consultant can facilitate open and honest discussion that will provide meaningful information.

If hiring a third-party consultant to perform an assessment, it is important to properly evaluate its qualifications and set up the engagement carefully. The results of the security assessment will serve as the guidance for short- and long-term security initiatives; therefore, it is imperative to perform the appropriate due-diligence evaluation of any consulting firm considered. In evaluating a third-party consultant, some attributes that management should review include:

- *Client references.* Determine where the client has previously performed security assessments.
- *Sample deliverables.* Obtain a sense of the type of report that will be provided. Clients sometimes receive boilerplate documents or voluminous reports from security software packages that are difficult to decipher, not always accurate, and fail to adequately define the risks.
- *Qualifications of the consultants.* Determine if the consultants have technical or industry certifications (e.g., CISSP, CISA, MCSE, etc.) and what type of experience they have.
- *Methodology and tools.* Determine if the consultants have a formal methodology for performing the assessment and what tools are used to do some of the technical pieces of the assessment.

Because the security assessment will provide a roadmap for the information security program, it is critical that a quality assessment be performed. Once the selection of who is to do the security assessment is finalized, management should define or put parameters around the engagement. Some things to consider include:

- *Scope.* The scope of the assessment must be very clear, that is, network, servers, specific departments or business units, etc.
- *Timing.* One risk with assessments is that they can drag on. The people who will offer input should be identified as soon as possible, and a single point of contact should be appointed to work with the consultants or auditors performing the assessment to ensure that the work is completed on time.
- *Documentation.* The results of the assessment should be presented in a clear and concise fashion so management understands the risks and recommendations.

Standards

The actual security assessment must measure the security posture of a company against standards. Security standards range from ones that address high-level operational processes to more technical and sometimes technology-specific standards. Some examples include:

- *ISO 17799: Information Security Best Practices.* This standard was developed by a consortium of companies and describes best practices for information security in the areas listed below. This standard is very process driven and is technology independent.
  - — Security policy
  - — Organizational security
  - — Asset classification and control
  - — Personnel security
  - — Physical and environmental security
  - — Communications and operations management
  - — Access control
  - — Systems development and maintenance
  - — Business continuity management
  - — Compliance
- *Common Criteria* (http://www.commoncriteria.org). "Represents the outcome of a series of efforts to develop criteria for evaluation of IT security products that are broadly useful within the international community."[1] The Common Criteria are broken down into the three parts listed below:

  - — *Part 1:* Introduction and general model: defines general concepts and principles of IT security evaluation and presents a general model for evaluation
  - — *Part 2:* Security functional requirements
  - — *Part 3:* Security assurance requirements
- *SANS/FBI Top 20 Vulnerabilities* (http://www.sans.org/top20.htm). This is an updated list of the 20 most significant Internet security vulnerabilities broken down into three categories: general, UNIX related, and NT related.
- *Technology-specific standards.* For instance, best practices for locking down Microsoft products can be found on the Microsoft Web site.

When performing an assessment, parts or all of the standards listed above or other known standards can be used. In addition, the consultant or auditor should leverage past experience and his or her knowledge of the company.

Understanding the Business

To perform an effective security assessment, one must have a thorough understanding of the business environment. Some of the components of the business environment that should be understood include:

- What are the inherent risks for the industry in which the company operates?
- What is the long- and short-term strategy for the company?
  - — What are the current business requirements, and how will this change during the short term and the long term?
- What is the organizational structure, and how are security responsibilities handled?
- What are the critical business processes that support the core operations?
- What technology is in place?

To answer these and other questions, the appropriate individuals, including business process owners, technology owners, and executives, should be interviewed.

Inherent Risks

As part of obtaining a detailed understanding of the company, an understanding of the inherent risks in the business is required. Inherent risks are those risks that exist in the business without considering any controls. These risks are a result of the nature of the business and the environment in which it operates. Inherent risks can be related to a particular industry or to general business practices, and can range from regulatory concerns as a result of inadequate protection of data to risks associated with disgruntled employees within an information technology (IT) department. These risks can be ascertained by understanding the industry and the particular company. Executives are often a good source of this type of information.

Business Strategy

Understanding the business strategy can help identify what is important to a company. This will ultimately be a factor in the risk assessment and the associated recommendations. To determine what is important to a company, it is important to understand the long- and short-term strategies. To take this one step further, how will IT support the long- and short-term business strategies? What will change in the IT environment once the strategies are implemented? The business strategy gives an indication of where the company is heading and what is or is not important. For example, if a company is planning on consolidating business units, the security assessment might focus on integration issues related to consolidation, which would be valuable input in developing a consolidation strategy.

One example of a prevalent business strategy for companies of all sizes is facilitating employee telecommuting. In today's environment, employees are increasingly accessing corporate networks from hotels or their homes during business hours as well as off-hours. Executives as well as lower-level personnel have become dependent on the ability to access company resources at any time. From a security assessment perspective, the

key objective is to determine if the infrastructure supporting remote access is secure and reliable. Some questions that an assessment might address in evaluating a remote access strategy include:

- How will remote users access the corporate network (e.g., dial in, VPN, etc.)?
- What network resources do remote users require (e.g., e-mail, shared files, certain applications)?
  - Based on what users must access, what kind of bandwidth is required?
- What is the tolerable downtime for remote access?

Each of the questions above has technology and process implications that need to be considered as part of the security assessment.

In addition to the business strategies, it is also helpful to understand security concerns at the executive level. Executives offer the "big-picture" view of the business, which others in the business sometimes do not. This high-level view can help prioritize the findings of a security assessment according to what is important to senior management. Interfacing with executives also provides an opportunity to make them more aware of security exposures that may potentially exist.

Organizational Structure

For an information security program to be effective, the organization structure must adequately support it. Where the responsibility for information security resides in an organization is often an indication of how seriously management views information security. In many companies today, information security is the responsibility of a CISO (chief information security officer) who might to report to either the CIO (chief information officer) or the CEO (chief executive officer). The CISO position has risen in prominence since the September 11 attacks. According to a survey done in January 2002 by Booz Allen Hamilton, "firms with more than $1 billion in annual revenues … 54 percent of the 72 chief executive officers it surveyed have a chief security officer in place. Ninety percent have been in that position for more than two years."[2] In other companies, either middle- or lower-level management within an IT organization handles security.

Having a CISO can be an indication that management has a high level of awareness of information security issues. Conversely, information security responsibility at a lower level might mean a low level of awareness of information security. While this is not always true, a security assessment must ascertain management and company attitude regarding the importance of information security. Any recommendations that would be made in the context of a security assessment must consider the organizational impact and, more importantly, whether the current setup of the organization is conducive to implementing the recommendations of the security assessment in the first place.

Another aspect of where information security resides in an organization is whether roles and responsibilities are clearly defined. As stated earlier, information security is a combination of process and technology. Roles and responsibilities must be defined such that there is a process owner for the key information security-related processes. In evaluating any part of an information security program, one of the first questions to ask is: "Who is responsible for performing the process?" Oftentimes, a security assessment may reveal that, while the process is very clearly defined and adequately addresses the business risk, no one owns it. In this case, there is no assurance that the process is being done. A common example of this is the process of ensuring that terminated employees are adequately processed. When employees are terminated, some things that are typically done include:

- Payroll is stopped.
- All user access is eliminated.
- All assets (i.e., computers, ID badges, etc.) are returned.
- Common IDs and passwords that the employee was using are changed.

Each of the steps above requires coordination among various departments, depending on the size and structure of a given company. Ensuring that terminated employees are processed correctly might mean coordination among departments such as human resources, IT, finance, and others. To ensure the steps outlined above are completed, a company might have a form or checklist to help facilitate communication among the relevant departments and to have a record that the process has been completed. However, without someone in the company owning the responsibility of ensuring that the items on the checklist are completed, there is no assurance that a terminated employee is adequately processed. It might be the case that each department

thought someone else was responsible for it. Too often, in the case of terminated employees, processing is incomplete because of a lack of ownership of the process, which presents significant risk for any company.

Once there are clear roles and responsibilities for security-related processes, the next step is to determine how the company ensures compliance. Compliance with security processes can be checked using two methods.

First, management controls can be built into the processes to ensure compliance. Building on the example of terminated employees, one of the significant elements in the processing is to ensure that the relevant user IDs are removed. If the user IDs of the terminated employees are, by mistake, not removed, it can be still be caught during periodic reviews of user IDs. This periodic review is a management control to ensure that only valid user IDs are active, while also providing a measure of security compliance.

The second method of checking compliance is an audit. Many internal audit departments include information security as part of their scope as it grows in importance. The role of internal audit in an information security program is twofold. First, audits check compliance with key security processes. Internal audits focus on different processes and related controls on a rotation basis over a period of time based on risk. The auditors gain an understanding of the processes and associated risks and ensure that internal controls are in place to reasonably mitigate the risks. Essentially, internal audit is in a position to do a continuous security assessment. Second, internal audits provide a company with an independent evaluation of the business processes, associated risks, and security policies. Because of their experience with and knowledge of the business and technology, internal auditors can evaluate and advise on security processes and related internal controls.

While there are many internal audit departments that do not have an adequate level of focus on information security, its inclusion within the scope of internal audit activities is an important indication about the level of importance placed on it. Internal audit is in a unique position to raise the level of awareness of information security because of its independence and access to senior management and the audit committee of the board of directors.

Business Processes

In conjunction with understanding the organization, the core business processes must be understood when performing a security assessment. The core business processes are those that support the main operations of a company. For example, the supply-chain management process is a core process for a manufacturing company. In this case, the security related to the systems supporting supply-chain management would warrant a close examination.

A good example of where core business processes have resulted in increased security exposures is business-to-business (B2B) relationships. One common use of a B2B relationship is where business partners manage supply-chain activities using various software packages. In such a relationship, business partners might have access to each other's manufacturing and inventory information. Some controls for potential security exposures as a result of such an arrangement include ensuring that:

- Business partners have access based on a need-to-know basis
- Communication of information between business partners is secure
- B2B connection is reliable.

These security exposure controls have information security implications and should be addressed in an information security program. For example, ensuring that business partners have access on a need-to-know basis might be accomplished using the access control features of the software as well as strict user ID administration procedures. The reliability of the B2B connection might be accomplished with a combination of hardware and software measures as well as SLAs (service level agreements) establishing acceptable downtime requirements.

In addition to the core business processes listed above, security assessments must consider other business processes in place to support the operations of a company, including:

- Backup and recovery
- Information classification
- Information retention
- Physical security
- User ID administration

- Personnel security
- Business continuity and disaster recovery
- Incident handling
- Software development
- Change management
- Noncompliance

The processes listed above are the more traditional security-related processes that are common across most companies. In some cases, these processes might be discussed in conjunction with the core business processes, depending on the environment. In evaluating these processes, guidelines such as the ISO 17799 and the Common Criteria can be used as benchmarks.

It is important to remember that understanding any of the business processes means understanding the manual processes as well as the technology used to support them. Business process owners and technology owners should be interviewed to determine exactly how the process is performed. Sometimes, a walk-through is helpful in gaining this understanding.

Technology Environment

As stated in the previous section, the technology supporting business processes is an important part of the security assessment. The technology environment ranges from industry-specific applications, to network operating systems, to security software such as firewalls and intrusion detection systems. Some of the more common areas to focus on in a security assessment include:

- Critical applications
- Local area network
- Wide area network
- Server operating systems
- Firewalls
- Intrusion detection systems
- Anti-virus protection
- Patch levels

When considering the technology environment, it is important to not only identify the components but also to determine how they are used. For example, firewalls are typically installed to filter traffic going in and out of a network. In a security assessment, one must understand what the firewall is protecting and if the rule base is configured around business requirements. Understanding whether the technology environment is set up in alignment with business requirements will enable a more thoughtful security assessment.

Risk Assessment

Once there is a good understanding of the business, its critical processes, and the technology supporting the business, the actual risk assessment can be done — that is, what is the risk as a result of the security exposures? While gaining an understanding of the business and the risk assessment are listed as separate steps, it is important to note that both of these steps will tend to happen simultaneously in the context of an audit; and this process will be iterative to some extent. Due to the nature of how information is obtained and the dynamic nature of a security assessment, the approach to performing the assessment must be flexible.

The assessment of risk takes the understanding of the critical processes and technology one step further. The critical business processes and the associated security exposures must be evaluated to determine what the risk is to the company. Some questions to think about when determining risk include:

- What is the impact to the business if the business process cannot be performed?
- What is the monetary impact?
  - Cost to restore information
  - Regulatory penalties

- What is the impact to the reputation of the company?
- What is the likelihood of an incident due to the security exposure?
- Are there any mitigating controls that reduce the risk?

It is critical to involve business process and technology owners when determining risks. Depending on how the assessment is performed, some of the questions will come up or be answered as the initial information is gathered. In addition, other more detailed questions will come up that will provide the necessary information to properly assess the risk.

In addition to evaluating the business processes, the risk assessment should also be done relative to security exposures in the technology environment. Some areas on which to focus here include:

- Perimeter security (firewalls, intrusion detection, etc.)
- Servers
- Individual PCs
- Anti-virus software
- Remote access

Security issues relating to the specific technologies listed above may come up during the discussions about the critical business processes. For example, locking down servers may arise because it is likely that there are servers that support some of the critical business processes.

Once all the security risks have been determined, the consultant or auditor must identify what measures are in place to mitigate the risks. Some of the measures to look for include:

- Information security policies
- Technical controls (e.g., servers secured according to best-practice standards)
- Business process controls (e.g., review of logs and management reports)

The controls may be identified while the process is reviewed and the risk is determined. Again, a security assessment is an iterative process in which information may not be uncovered in a structured manner. It is important to differentiate and organize the information so that risk is assessed properly.

The combination of security exposures and controls (or lack thereof) to mitigate the associated risks should then be used to develop the gap analysis and recommendations. The gap analysis is essentially a detailed list of security exposures, along with controls to mitigate the associated risks. Those areas where there are inadequate controls or no controls to mitigate the security exposure are the gaps, which potentially require remediation of some kind.

The final step in the gap analysis is to develop recommendations to close the gaps. Recommendations could range from writing a security policy to changing the technical architecture to altering how the current business process is performed. It is very important that the recommendations consider the business needs of the organization. Before a recommendation is made, a cost/benefit analysis should be done to ensure that it makes business sense. It is possible that, based on the cost/benefit analysis and operational or financial constraints, the organization might find it reasonable to accept certain security risks. Because the recommendations must be sold to management, they must make sense from a business perspective.

The gap analysis should be presented in an organized format that management can use to understand the risks and implement the recommendations. An effective way to present the gap analysis is with a risk matrix with the following columns represented:

- Finding
- Risk
- Controls in place
- Recommendation

This format provides a simple and concise presentation of the security exposures, controls, and recommendations. The presentation of the gap analysis is very important because management will use it to understand the security exposures and associated risks. In addition, the gap analysis can be used to prioritize short- and long-term security initiatives.

Conclusion

For many companies, the security assessment is the first step in developing an effective information security program because many organizations do not know where they are from a security perspective. An independent security assessment and the resulting gap analysis can help determine what the security exposures are, as well as provide recommendations for additional security measures that should be in implemented. The gap analysis can also help management prioritize the tasks in the event that all the recommendations could not be immediately implemented.

The gap analysis reflects the security position at a given time, and the recommendations reflect current and future business requirements to the extent they are known. As business requirements and technologies change, security exposures will invariably change. To maintain a sound information security program, the cycle of assessments, gap analysis, and implementation of recommendations should be done on a continuous basis to effectively manage security risk.

References

1. Common Criteria Web page: http://www.commoncriteria.org/docs/origins.html.
2. Flash, Cynthia, Rise of the chief security officer, *Internet News*, March 25, 2002, http://www. internet-news.com/ent-news/article/0,7_997111,00.html.

Evaluating the Security Posture of an Information Technology Environment: The Challenges of Balancing Risk, Cost, and Frequency of Evaluating Safeguards

Brian R. Schultz, CISSP, CISA

The elements that could affect the integrity, availability, and confidentiality of the data contained within an information technology (IT) system must be assessed periodically to ensure that the proper safeguards have been implemented to adequately protect the resources of an organization. More specifically, the security that protects the data contained within the IT systems should be evaluated regularly. Without the assurance that the data contained within the system has integrity and is therefore accurate, the system is useless to serve the stakeholders who rely on the accuracy of such data.

Historically, safeguards over a system have been evaluated as a function of compliance with laws, regulations, or guidelines that are driven by an external entity. External auditors such as financial statement auditors might assess security over a system to understand the extent of security controls implemented and whether these controls are adequate to allow them to rely on the data processed by the systems. Potential partners for a merger might assess the security of an organization's systems to determine the effectiveness of security measures and to gain a better understanding of the systems' condition and value. See Exhibit 21-1 for a list of common IT evaluation methodologies.

Exhibit 21-1. Common IT evaluation types.

Type of Evaluation: Financial Statement Audit

Stakeholders: All professionals who work for the organization or who own a company that undergoes an annual financial statement audit.

Description: Financial statement auditors review the financial data of an organization to determine whether the financial data is accurately reported. As a component of performing the financial statement audit, they also review the controls (safeguards) used to protect the integrity of the data. Financial statement auditors are not concerned with the confidentiality or availability of data as long as it has no impact on the integrity of the data. This work will be conducted in accordance with American Institute of Certified Public Accountants (AICPA) standards for public organizations and in accordance with the Federal Information System Control Audit Methodology (FISCAM) for all U.S. federal agency financial statement audits.

Type of Evaluation: Due Diligence Audit before the Purchase of a Company

Stakeholders: Potential buyers of a company.

Description: Evaluation of the safeguards implemented and the condition of an IT system prior to the purchase of a company.

Type of Evaluation: SAS 70 Audit

Stakeholders: The users of a system that is being processed by a facility run by another organization.

Description: The evaluation of data centers that process (host) applications or complete systems for several organizations. The data center will frequently obtain the services of a third-party organization to perform an IT audit over the data center. The report, commonly referred to as an SAS 70 Report, provides an independent opinion of the safeguards implemented at the shared data center. The SAS 70 Report is generally shared with each of the subscribing organizations that uses the services of the data center. Because the SAS 70 audit and associated report are produced by a third-party independent organization, most subscribing organizations of the data center readily accept the results to be sufficient, eliminating the need to initiate their own audits of the data center.

Type of Evaluation: Federal Financial Institutions Examination Council (FFIEC) Information Systems Examination

Stakeholders: All professionals in the financial industry and their customers.

Description: Evaluation of the safeguards affecting the integrity, reliability, and accuracy of data and the quality of the management information systems supporting management decisions.

Type of Evaluation: Health Insurance Portability Accountability Act (HIPAA) Compliance Audit

Stakeholders: All professionals in health care and patients.

Description: Evaluation of an organization's compliance with HIPAA specifically in the area of security and privacy of healthcare data and data transmissions.

Exhibit 21-1. Common IT evaluation types (Continued).

Type of Evaluation: U.S. Federal Government Information Systems Reform Act (GISRA) Review

Stakeholders: All U.S. federal government personnel and American citizens.

Description: Evaluation of safeguards of federal IT systems with a final summary report of each agency's security posture provided to the Office of Management and Budget.

Type of Evaluation: U.S. Federal Government Risk Assessment in compliance with Office of Management and Budget Circular A-130

Stakeholders: All federal government personnel and those who use the data contained within those systems.

Description: Evaluation of U.S. government major applications and general support systems every three years to certify and accredit that the system is properly secured to operate and process data.

Evaluations of IT environments generally are not performed proactively by the IT department of an organization. This is primarily due to a performance-focused culture within the ranks of the chief information officers and other executives of organizations who have been driven to achieve performance over the necessity of security. As more organizations experience performance issues as a result of lack of effective security, there will be more proactive efforts to integrate security into the development of IT infrastructures and the applications that reside within them. In the long run, incorporating security from the beginning is significantly more effective and results in a lower cost over the life cycle of a system.

Internal risk assessments should be completed by the information security officer or an internal audit department on an annual basis and more often if the frequency of hardware and software changes so necessitates. In the case of a major launch of a new application or major platform, a pre-implementation (before placing into production) review should be performed. If an organization does not have the capacity or expertise to perform its own internal risk assessment or pre-implementation evaluation, a qualified consultant should be hired to perform the risk assessment. The use of a contractor offers many advantages:

- Independent evaluators have a fresh approach and will not rely on previously formed assumptions.
- Independent evaluators are not restricted by internal politics.
- Systems personnel are generally more forthright with an outside consultant than with internal personnel.
- Outside consultants have been exposed to an array of systems of other organizations and can offer a wider perspective on how the security posture of the system compares with systems of other organizations.

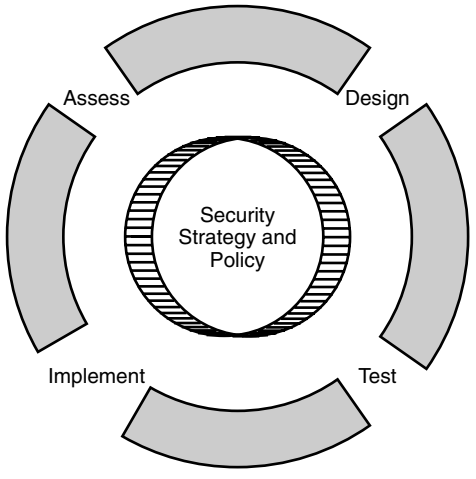


Exhibit 21-2. Security life-cycle model.

- Outside consultants might have broader technology experience based on their exposure to multiple technologies and therefore are likely to be in a position to offer recommendations for improving security.

When preparing for an evaluation of the security posture of an IT system, the security life-cycle model should be addressed to examine the organization's security strategy, policies, procedures, architecture, infrastructure design, testing methodologies, implementation plans, and prior assessment findings.

SECURITY LIFE-CYCLE MODEL

The *security life-cycle model* contains all of the elements of security for a particular component of security of an information technology as seen in Exhibit 21-2. Security elements tend to work in cycles. Ideally, the *security strategy and policy* are determined with a great deal of thought and vision followed by the sequential phases of *design*, *test*, *implement* and, finally, *assess*.

The *design phase* is when the risk analyst examines the design of safeguards and the chosen methods of implementation. In the second phase, the *test phase*, the risk assessment examines the testing procedures and processes that are used before placing safeguards into production. In the following phase, the *implementation phase*, the risk assessment analyzes the effectiveness of the technical safeguards settings contained within the operating system, multilevel security, database management system, application-level security, public key infrastructure, intrusion detection system, firewalls, and routers. These safeguards are evaluated using

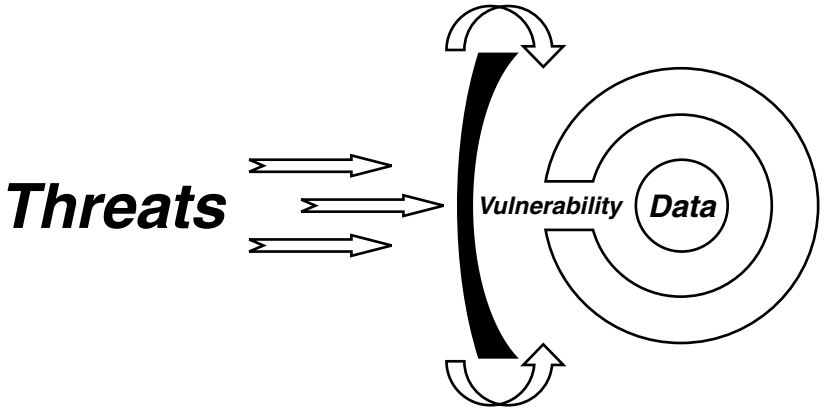


Exhibit 21-3. Elements of an organization's security posture.

technical vulnerability tools as well as a manual review of security settings provided on printed reports.

Assessing security is the last phase of the security life-cycle model, and it is in this phase that the actions taken during the previous phases of the security life-cycle model are assessed. The *assess* phase is the feedback mechanism that provides the organization with the condition of the security posture of an IT environment. The risk assessment first focuses on the security strategy and policy component of the model. The security strategy and policy component is the core of the model, and many information security professionals would argue that this is the most important element of a successful security program. The success or failure of an organization's security hinges on a well-formulated, risk-based security strategy and policy. When used in the appropriate context, the security life-cycle model is an effective tool to use as a framework in the evaluation of IT security risks.

ELEMENTS OF RISK ASSESSMENT METHODOLOGIES

A risk assessment is an active process that is used to evaluate the security of an IT environment. Contained within each security assessment methodology are the elements that permit the identification and categorization of the components of the security posture of a given IT environment. These identified elements provide the language necessary to identify, communicate, and report the results of a risk assessment. These elements are comprised of threats, vulnerabilities, safeguards, countermeasures, and residual risk analysis. As seen in Exhibit 21-3, each of these elements is dynamic and, in combination, constitutes the security posture of the IT environment.

THREATS

A threat is a force that could affect an organization or an element of an organization. Threats can be either external or internal to an organization and, by themselves, are not harmful. However, they have the potential to be harmful. Threats are also defined as either man-made — those that mankind generates — or natural — those that naturally occur. For a threat to affect an organization, it must exploit an existing vulnerability. Every organization is vulnerable to threats. The number, frequency, severity, type, and likelihood of each threat are dependent on the environment of the IT system. Threats can be ranked on a relative scale of low, medium, and high, based on the potential risk to an asset or group of assets.

- *Low* indicates a relatively low probability that this threat would have significant effect.
- *Medium* indicates a moderate probability that this threat would have significant effect if not mitigated by an appropriate safeguard.
- *High* indicates a relatively high probability that the threat could have significant effect if not mitigated by an appropriate safeguard or series of safeguards.

VULNERABILITY

Vulnerability is a weakness or condition of an organization that could permit a threat to take advantage of the weakness to affect its performance. The absence of a firewall to protect an organization's network from external attacks is an example of vulnerability in the protection of the network from potential external attacks. All organizations have and will continue to have vulnerabilities. However, each organization should identify the potential threats that could exploit vulnerabilities and properly safeguard against threats that could have a dramatic effect on performance.

SAFEGUARDS

Safeguards, also called controls, are measures that are designed to prevent, detect, protect, or sometimes react to reduce the likelihood — or to completely mitigate the possibility — of a threat to exploit an organization's vulnerabilities. Safeguards can perform several of these functions at the same time, or they may only perform one of these functions. A firewall that is installed and configured properly is an example of a safeguard to prevent external attacks to the organization's network. Ideally, a "defense-in-depth" approach should be deployed to implement multiple layers of safeguards to establish the appropriate level of protection for the given environment. The layering of protection provides several obstacles for an attacker, thereby consuming the attacker's resources of time, money, and risk in continuing the attack. For instance, a medical research firm should safeguard its product research from theft by implementing a firewall on its

network to prevent someone from obtaining unauthorized access to the network. In addition, the firm might also implement a network intrusion detection system to create an effective defense-in-depth approach to external network safeguards.

A countermeasure is a type of safeguard that is triggered by an attack and is reactive in nature. Its primary goal is to defend by launching an offensive action. Countermeasures should be deployed with caution because they could have a profound effect on numerous systems if activated by an attack.

RESIDUAL RISK ANALYSIS

As a risk assessment is completed, a list of all of the identified vulnerabilities should be documented and a residual risk analysis performed. Through this process, each individual vulnerability is examined along with the existing safeguards (if any), and the residual risk is then determined. The final step is the development of recommendations to strengthen existing safeguards or recommendations to implement new safeguards to mitigate the identified residual risk.

RISK ASSESSMENT METHODOLOGIES

Several risk assessment methodologies are available to the information security professional to evaluate the security posture of an IT environment. The selection of a methodology is based on a combination of factors, including the purpose of the risk assessment, available budget, and the required frequency.

The primary consideration in selecting a risk assessment methodology, however, is the need of the organization for performing the risk assessment. The depth of the risk assessment required is driven by the level of risk attributed to the continued and accurate performance of the organization's systems. An organization that could be put out of business by a systems outage for a few days would hold a much higher level of risk than an organization that could survive weeks or months without their system. For example, an online discount stockbroker would be out of business without the ability to execute timely stock transactions, whereas a construction company might be able to continue operations for several weeks without access to its systems without significant impact.

An organization's risk management approach should also be considered before selecting a risk assessment methodology. Some organizations are proactive in their approach to addressing risk and have a well-established risk management program. Before proceeding in the selection of a risk assessment methodology, it would be helpful to determine if the organization has such a program and the extent of its depth and breadth. In the case

of a highly developed risk assessment methodology, several layers of safeguards are deployed and require a much different risk assessment approach than if the risk management program were not developed and few safeguards had been designed and deployed. Gaining an understanding of the design of the risk management program, or lack thereof, will enable the information security professional conducting the risk assessment to quickly identify the layers of controls that should be considered when scoping the risk assessment.

The risk assessment methodologies available to the information security professional are general and not platform specific. There are several methodologies available, and the inexperienced information security professional and those not familiar with the risk assessment process will quickly become frustrated with the vast array of methodologies and opinions with regard to how to conduct an IT risk assessment. It is the author's opinion that all IT risk assessment methodologies should be based on the platform level. This is the only real way to thoroughly address the risk of a given IT environment. Some of the highest risks associated within an IT environment are technology specific; therefore, each risk assessment should include a technical-level evaluation. However, the lack of technology-specific vulnerability and safeguard information makes the task of a technically driven risk assessment a challenge to the information security professional. Hardware and software changes frequently open up new vulnerabilities with each new version. In an ideal world, a centralized depository of vulnerabilities and associated safeguards would be available to the security professional. In the meantime, the information security professional must rely on decentralized sources of information regarding technical vulnerabilities and associated safeguards. Although the task is daunting, the information security professional can be quite effective in obtaining the primary goal, which is to reduce risk to the greatest extent possible. This might be accomplished by prioritizing risk mitigation efforts on the vulnerabilities that represent the highest risk and diligently eliminating lower-risk vulnerabilities until the risk has been reduced to an acceptable level.

Several varieties of risk assessments are available to the information security professional, each one carrying unique qualities, timing, and cost. In addition, risk assessments can be scoped to fit an organization's needs to address risk and to the budget available to address risk. The lexicon and standards of risk assessments vary greatly. While this provides for a great deal of flexibility, it also adds a lot of frustration when trying to scope an evaluation and determine the associated cost. Listed below are several of the most common types of risk assessments.

QUALITATIVE RISK ASSESSMENT

A qualitative risk assessment is subjective, based on best practices and the experience of the professional performing it. Generally, the findings of a qualitative risk assessment will result in a list of vulnerabilities with a relative ranking of risk (low, medium, or high). Some standards exist for some specific industries, as listed in Exhibit 21-1; however, qualitative risk assessments tend to be open and flexible, providing the evaluator a great deal of latitude in determining the scope of the evaluation. Given that each IT environment potentially represents a unique combination of threats, vulnerabilities, and safeguards, the flexibility is helpful in obtaining quick, cost-effective, and meaningful results. Due to this flexibility, the scope and cost of the qualitative risk assessment can vary greatly. Therefore, evaluators have the ability to scope evaluations to fit an available budget.

QUANTITATIVE RISK ASSESSMENT

A quantitative risk assessment follows many of the same methodologies of a qualitative risk assessment, with the added task of determining the cost associated with the occurrence of a given vulnerability or group of vulnerabilities. These costs are calculated by determining asset value, threat frequency, threat exposure factors, safeguard effectiveness, safeguard cost, and uncertainty calculations. This is a highly effective methodology in communicating risk to an audience that appreciates interpreting risk based on cost. For example, if an information systems security officer of a large oil company wanted to increase the information security budget of the department, presentation of the proposed budget to the board of directors for approval is required. The best way for this professional to effectively communicate the need for additional funding to improve safeguards and the associated increase in the budget is to report the cost of the risk in familiar terms with which the board members are comfortable. In this particular case, the members of the board are very familiar with financial terms. Thus, the expression of risk in terms of financial cost provides a compelling case for action. For such an audience, a budget increase is much more likely to be approved if the presenter indicates that the cost of not increasing the budget has a high likelihood of resulting in a "two billion dollar loss of revenue" rather than "the risk represents a high operational cost." Although the risk represented is the same, the ability to communicate risk in financial terms is very compelling.

A quantitative risk assessment approach requires a professional or team of professionals who are exceptional in their professions to obtain meaningful and accurate results. They must be well seasoned in performing qualitative and quantitative risk assessments, as the old GI-GO (garbage-in, garbage-out) rule applies. If the persons performing the quantitative risk assessment do not properly estimate the cost of an asset and frequency of

loss expectancy, the risk assessment will yield meaningless results. In addition to requiring a more capable professional, a quantitative risk assessment approach necessitates the use of a risk assessment tool such as RiskWatch or CORA (Cost of Risk Analysis). The requirement for the advanced skills of a quantitative risk assessment professional and the use of a quantitative risk assessment tool significantly increases the cost above that of a qualitative risk assessment. For many organizations, a qualitative risk assessment would be more than adequate to identify risk for appropriate mitigation.

As a word of caution when using a quantitative approach, much like the use of statistics in politics to influence an audience's opinion, the cost information that results from a quantitative risk assessment could be manipulated to lead an audience to a variety of conclusions.

INFORMATION TECHNOLOGY AUDIT

IT audits are primarily performed by external entities and internal audit departments with the charge to determine the effectiveness of the security posture over an IT environment and, in the case of a financial statement audit, to determine the reliability (integrity) of the data contained within the system. They essentially focus on the adequacy of and compliance with existing policies, procedures, technical baseline controls, and guidelines. Therefore, the primary purpose of an IT audit is to report the condition of the system and not to improve security. However, IT auditors are usually more than willing to share their findings and recommendations with the IT department. In addition, IT auditors are required to document their work in sufficient detail as to permit another competent IT auditor to perform the exact same audit procedure (test) and come to the same conclusion. This level of documentation is time-consuming and therefore usually has an effect on the depth and breadth of the evaluation. Thus, IT audits may not be as technically deep in scope as a non-audit type of evaluation.

TECHNICAL VULNERABILITY ASSESSMENT

A technical vulnerability assessment is a type of risk assessment that is focused primarily on the technical safeguards at the platform and network levels and does not include an assessment of physical, environmental, configuration management, and management safeguards.

NETWORK TECHNICAL VULNERABILITY ASSESSMENT

The safeguards employed at the network level support all systems contained within its environment. Sometimes these collective systems are referred to as a general support system. Most networks are connected to the Internet, which requires protection from exterior threats. Accordingly, a network technical vulnerability assessment should include an evaluation of the

Exhibit 21-4. Automated technical vulnerability assessment tools.

Nessus. This is a free system security scanning software that provides the ability to remotely evaluate security within a given network and determine the vulnerabilities that an attacker might use.

ISS Internet Scanner. A security scanner that provides comprehensive network vulnerability assessment for measuring online security risks, it performs scheduled and selective probes of communication services, operating systems, applications, and routers to uncover and report systems vulnerabilities.

Shadow Security Scanner. This tool identifies known and unknown vulnerabilities, suggests fixes to identified vulnerabilities, and reports possible security holes within a network's Internet, intranet, and extranet environments. It employs a unique artificial intelligence engine that allows the product to think like a hacker or network security analyst attempting to penetrate your network.

NMAP. NMAP (Network Mapper) is an open-source utility for network exploration or security auditing. It rapidly scans large networks using raw IP packets in unique ways to determine what hosts are available on the network, what services (ports) they are offering, what operating system (and OS version) they are running, and what type of packet filters or firewalls are in use. NMAP is free software available under the terms of the GNU GPL.

Snort. This packet-sniffing utility monitors displays and logs network traffic.

L0ftCrack. This utility can crack captured password files through comparisons of passwords to dictionaries of words. If the users devised unique passwords, the utility uses brute-force guessing to reveal the passwords of the users.

safeguards implemented to protect the network and its infrastructure. This would include the routers, load balancers, firewalls, virtual private networks, public key infrastructure, single sign-on solutions, network-based operating systems (e.g., Windows 2000), and network protocols (e.g., TCP/IP). Several automated tools can be used to assist the vulnerability assessment team. See Exhibit 21-4 for a list of some of the more common tools used.

PLATFORM TECHNICAL VULNERABILITY ASSESSMENT

The safeguards employed at the platform level support the integrity, availability, and confidentiality of the data contained within the platform. A platform is defined as a combination of hardware, operating system software, communications software, security software, and the database management system and application security that support a set of data (see Exhibit 21-5 for an example of a mainframe platform diagram). The combination of these distinctly separate platform components contains a unique set of risks, necessitating that each platform be evaluated based on its unique combination. Unless the evaluator is able to examine the safeguards at the platform level, the integrity of the data cannot be properly and completely assessed and, therefore, is not reliable. Several automated tools can be used by the vulnerability assessment team.

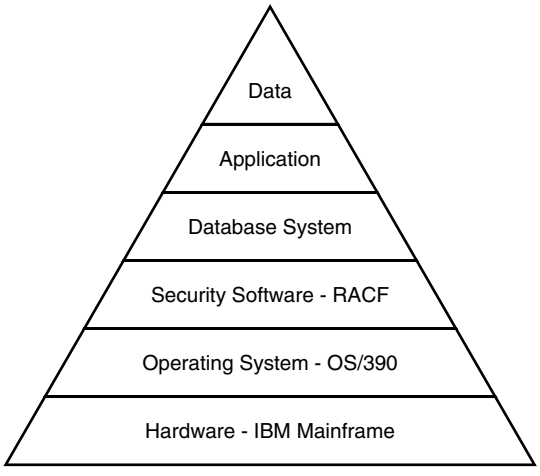


Exhibit 21-5. Mainframe platform diagram.

PENETRATION TESTING

A penetration test, also known as a pen test, is a type of risk assessment; but its purpose is quite different. A pen test is designed to test the security of a system after an organization has implemented all designed safeguards, performed a risk assessment, implemented all recommended improvements, and implemented all new recommended safeguards. It is the final test to determine if enough layered safeguards have been sufficiently implemented to prevent a successful attack against the system. This form of ethical hacking attempts to find vulnerabilities that have been overlooked in prior risk assessments. Frequently, a successful penetration is accomplished as a result of the penetration team, otherwise known as a tiger team, discovering multiple vulnerabilities that by themselves are not considered high risk but, when combined, create a backdoor permitting the penetration team to successfully exploit the low-risk vulnerabilities. There are several potential components to a pen test that, based on the organization's needs, can be selected for use:

- *External penetration testing* is performed from outside of the organization's network, usually from the Internet. The organization can either provide the pen team with the organization's range of IP addresses or ask the evaluators to perform a blind test. Blind tests are more expensive because it will take the penetration team time to discover the IP addresses of the organization. While it might seem to be a more effective test to have the team perform a blind test, it is inevitable that the team will find the IP addresses; therefore, it may be considered a waste of time and funds.

- *Internal penetration testing* is performed within the internal network of the organization. The penetration team attempts to gain access to sensitive unauthorized areas of the system. The internal penetration test is a valuable test, especially in light of the fact that an estimated 80 percent of incidents of unauthorized access are committed by employees.
- *Social engineering* can be used by the pen testers to discover vital information from the organization's personnel that might be helpful in launching an attack. For instance, a pen tester might drive up to the building of the organization, write down the name on an empty reserved parking space, and then call the help desk impersonating the absent employee to report that they had forgotten their password. The pen tester would then request that his password be reset so that he can get back into the system. Unless the help desk personnel have a way (employee number, etc.) to verify his identity, they will reset the password, giving the attacker the opportunity to make a new password for the absent employee and gain unauthorized access to the network.
- *War dialing tools* can be used to automatically dial every combination of phone numbers for a given phone number exchange in an attempt to identify a phone line that has a modem connected. Once a phone line with an active modem has been discovered, the penetration team will attempt to gain access to the system.
- *Dumpster diving* is the practice of searching through trash cans and recycling bins in an attempt to obtain information that will allow the penetration team to gain access to the system.

Penetration testing is the most exciting of all of the risk assessments because it is an all-out attempt to gain access to the system. It is the only risk assessment methodology that proves the existence of a vulnerability or series of vulnerabilities. The excitement of penetration testing is also sometimes perpetuated by those who perform them. Some pen testers, also known as ethical hackers or "white hats," are retired hackers who at one time were "black hats."

Some organizations might be tempted to skip the detailed risk assessment and risk remediation plan and go straight to a penetration test. While pen testing is an enthralling process, the results will be meaningless if the organization does not do its homework before the penetration test. In all likelihood, a good penetration team will gain access to an organization's systems if it has not gone through the rigors of the risk assessment and improvement of safeguards.

EVALUATING IDENTIFIED VULNERABILITIES

After the vulnerabilities have been identified through a risk assessment, a vulnerability analysis should be performed to rank each vulnerability according to its risk level:

- *Low.* The risk of this vulnerability is not considered significant; however, when combined with several other low-risk vulnerabilities, the aggregate might be considered either a medium or high risk. Recommended safeguards need to be reviewed to determine if they are practical or cost-effective relative to the risk of the vulnerability.
- *Medium.* This risk is potentially significant. If the vulnerability could be exploited more readily in combination with another vulnerability, then this risk could be ranked higher. Corrective action of a medium risk level should be taken within a short period of time after careful consideration of the cost-effectiveness of implementing the recommended safeguard.
- *High.* The risk of this vulnerability is significant and, if exploited, could have profound effects on the viability of the organization. Immediate corrective action should be taken to mitigate the risk.

ANALYZING PAIRED VULNERABILITIES

In addition to ranking individual vulnerabilities, an analysis of all of the vulnerabilities should be performed to determine if any of the combinations of vulnerabilities, when considered together, represent a higher level of risk. These potentially higher-risk combinations should be documented and action taken to mitigate the risk. This is particularly important when considering the low-risk items because the combination of these lower-risk items could create the backdoor that permits an attacker to gain access to the system. To determine the relative nominal risk level of the identified vulnerabilities, the information security professional should identify potential layers of safeguards that mitigate a risk and then determine the residual risk. A residual risk mitigation plan should then be developed to reduce the residual risk to an acceptable level.

CONCLUSION

Unfortunately, security assessments are usually the last action that the IT department initiates as part of its security program. Other priorities such as application development, infrastructure building, or computer operations typically take precedence. Many organizations typically do not take security past the initial implementation because of a rush-to-build functionality of the systems — until an IT auditor or a hacker forces them to take security seriously. The "pressures to process" sometimes force organizations to ignore prudent security design and security assessment, leaving security as an afterthought. In these circumstances, security is not considered a critical element in serving the users; thus, many times security is left behind. The reality is that information contained within a system cannot be relied upon as having integrity unless security has been assessed and adequate protection of the data has been provided for the entire time the data has resided on the system.

Evaluating the security posture of an IT environment is a challenge that involves balancing the risk, frequency of evaluation, and cost. Security that is designed, tested, and implemented based on a strong security strategy and policy will be highly effective and in the long run cost-effective. Unfortunately, there are no clear-cut answers regarding how often a given IT environment should be evaluated. The answer may be found by defining how long the organization may viably operate without the systems. Such an answer will define the level of risk the organization is willing, or is not willing, to accept. A security posture that is built with the knowledge of this threshold of risk can lead to a system of safeguards that is both risk-based and cost-effective.

ABOUT THE AUTHOR

Brian Schultz, CISSP, CISA, is chairman of the board of INTEGRITY, a nonprofit organization dedicated to assisting the federal government with implementation of information security solutions. An expert in the field of information security assessment, Mr. Schultz has, throughout his career, assessed the security of numerous private and public organizations. He is a founding member of the Northern Virginia chapter of the Information Systems Security Association (ISSA).

69

Cyber-Risk Management: Technical and Insurance Controls for Enterprise-Level Security

Carol A. Siegel, Ty R. Sagalow, and Paul Serritella

Traditional approaches to security architecture and design have attempted to achieve the goal of the elimination of risk factors — the complete prevention of system compromise through technical and procedural means. Insurance-based solutions to risk long ago admitted that a complete elimination of risk is impossible and, instead, have focused more on reducing the impact of harm through financial avenues — providing policies that indemnify the policyholder in the event of harm.

It is becoming increasingly clear that early models of computer security, which focused exclusively on the risk-elimination model, are not sufficient in the increasingly complex world of the Internet. There is simply no magic bullet for computer security; no amount of time or money can create a perfectly hardened system. However, insurance cannot stand alone as a risk mitigation tool — the front line of defense must always be a complete information security program and the implementation of security tools and products. It is only through leveraging both approaches in a complementary fashion that an organization can reach the greatest degree of risk reduction and control. Thus, today, the optimal model requires a program of understanding, mitigating, and transferring risk through the use of integrating technology, processes, and insurance — that is, a risk management approach.

The risk management approach starts with a complete understanding of the risk factors facing an organization. Risk assessments allow for security teams to design appropriate control systems and leverage the necessary technical tools; they also are required for insurance companies to properly draft and price policies for the remediation of harm. Complete risk assessments must take into account not only the known risks to a system but also the possible exploits that might develop in the future. The completeness of cyber risk management and assessment is the backbone of any secure computing environment.

After a risk assessment and mitigation effort has been completed, insurance needs to be procured from a specialized insurance carrier of top financial strength and global reach. The purpose of the insurance is threefold: (1) assistance in the evaluation of the risk through products and services available from the insurer, (2) transfer of the financial costs of a successful computer attack or threat to the carrier, and (3) the provision of important post-incident support funds to reduce the potential reputation damage after an attack.

The Risk Management Approach

As depicted in Exhibit 69.1, risk management requires a continuous cycle of assessment, mitigation, insurance, detection, and remediation.

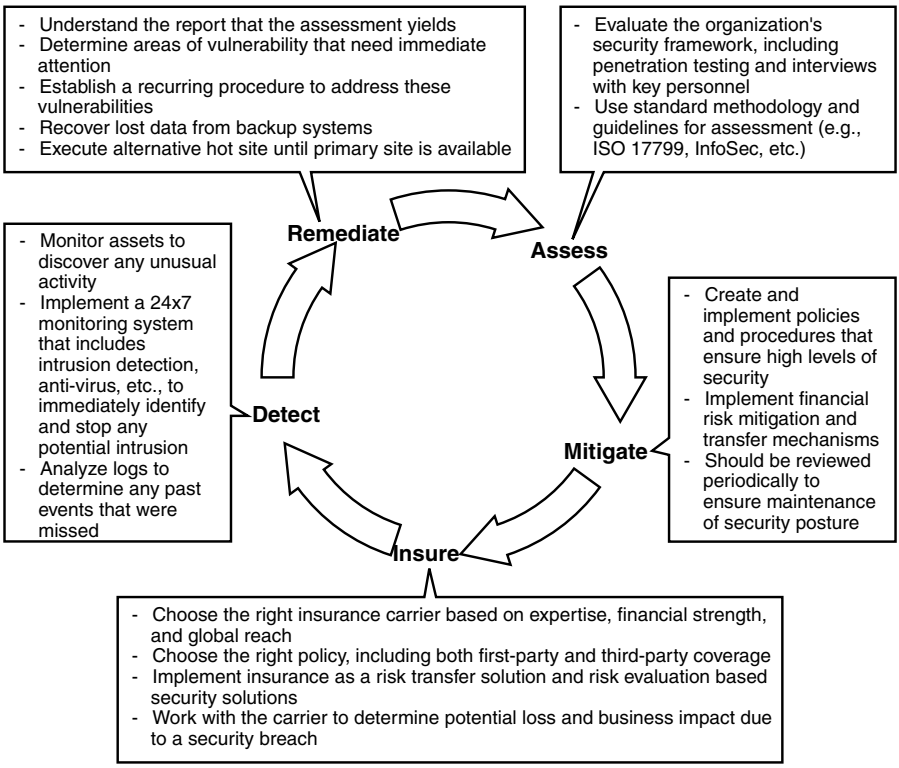


EXHIBIT 69.1 Risk management cycle.

Assess

An assessment means conducting a comprehensive evaluation of the security in an organization. It usually covers diverse aspects, ranging from physical security to network vulnerabilities. Assessments should include penetration testing of key enterprise systems and interviews with security and IT management staff. Because there are many different assessment formats, an enterprise should use a method that conforms to a recognized standard (e.g., ISO 17799, InfoSec — Exhibit 69.2). Regardless of the model used, however, the assessment should evaluate people, processes, technology, and financial management. The completed assessment should then be used to determine what technology and processes should be employed to mitigate the risks exposed by the assessment.

An assessment should be done periodically to determine new vulnerabilities and to develop a baseline for future analysis to create consistency and objectivity.

Mitigate

Mitigation is the series of actions taken to reduce risk, minimize chances of an incident occurring, or limit the impact of any breach that does occur. Mitigation includes creating and implementing policies that ensure high levels of security. Security policies, once created, require procedures that ensure compliance. Mitigation also includes determining and using the right set of technologies to address the threats that the organization faces and implementing financial risk mitigation and transfer mechanisms.

Insure

Insurance is a key risk transfer mechanism that allows organizations to be protected financially in the event of loss or damage. A quality insurance program can also provide superior loss prevention and analysis recommendations, often providing premium discounts for the purchase of certain security products and services from

EXHIBIT 69.2 The 11 Domains of Risk Assessment

Security Policy: During the assessment, the existence and quality of the organization's security policy are evaluated. Security policies should establish guidelines, standards, and procedures to be followed by the entire organization. These need to be updated frequently.

Organizational Security: One of the key areas that any assessment looks at is the organizational aspect of security. This means ensuring that adequate staff has been assigned to security functions, that there are hierarchies in place for security-related issues, and that people with the right skill sets and job responsibilities are in place.

Asset Classification and Control: Any business will be impacted if the software and hardware assets it has are compromised. In evaluating the security of the organization, the existence of an inventory management system and risk classification system have to be verified.

Personnel Security: The hiring process of the organization needs to be evaluated to ensure that adequate background checks and legal safeguards are in place. Also, employee awareness of security and usage policies should be determined.

Physical and Environmental Security: Ease of access to the physical premises needs to be tested, making sure that adequate controls are in place to allow access only to authorized personnel. Also, the availability of redundant power supplies and other essential services has to be ensured.

Communication and Operations Management: Operational procedures need to be verified to ensure that information processing occurs in a safe and protected manner. These should cover standard operating procedures for routine tasks as well as procedures for change control for software, hardware, and communication assets.

Access Control: This domain demands that access to systems and data be determined by a set of criteria based on business requirement, job responsibility, and time period. Access control needs to be constantly verified to ensure that it is available only on a need-to-know basis with strong justification.

Systems Development and Maintenance: If a company is involved in development activity, assess whether security is a key consideration at all stages of the development life cycle.

Business Continuity Management: Determining the existence of a business continuity plan that minimizes or eliminates the impact of business interruption is a part of the assessment.

Compliance: The assessment has to determine if the organization is in compliance with all regulatory, contractual, and legal requirements.

Financial Considerations: The assessment should include a review to determine if adequate safeguards have to be implemented to ensure that any security breach results in minimal financial impact. This is implemented through risk transfer mechanisms — primarily insurance that covers the specific needs of the organization.

companies known to the insurer that dovetail into a company's own risk assessment program. Initially, determining potential loss and business impact due to a security breach allows organizations to choose the right policy for their specific needs. The insurance component then complements the technical solutions, policies, and procedures. A vital step is choosing the right insurance carrier by seeking companies with specific underwriting and claims units with expertise in the area of information security, top financial ratings, and global reach. The right carrier should offer a suite of policies from which companies can choose to provide adequate coverage.

Detect

Detection implies constant monitoring of assets to discover any unusual activity. Usually this is done by implementing a 24/7 monitoring system that includes intrusion detection to immediately identify and stop any potential intrusion. Additionally, anti-virus solutions allow companies to detect new viruses or worms as they appear. Detection also includes analyzing logs to determine any past events that were missed and specification of actions to prevent future misses. Part of detection is the appointment of a team in charge of incident response.

Remediate

Remediation is the tactical response to vulnerabilities that assessments discover. This involves understanding the report that the assessment yields and prioritizing the areas of vulnerability that need immediate attention.

The right tactic and solution for the most efficient closing of these holes must be chosen and implemented. Remediation should follow an established recurring procedure to address these vulnerabilities periodically.

In the cycle above, most of the phases focus on the assessment and implementation of technical controls. However, no amount of time or money spent on technology will eliminate risk. Therefore, insurance plays a key role in any risk management strategy. When properly placed, the insurance policy will transfer the financial risk of unavoidable security exposures from the balance sheet of the company to that of the insurer. As part of this basic control, companies need to have methods of detection (such as intrusion detection systems, or IDS) in place to catch the cyber-attack when it takes place. Post incident, the insurer will then remediate any damage done, including finance and reputation impacts. The remediation function includes recovery of data, insurance recoveries, and potential claims against third parties. Finally, the whole process starts again with an assessment of the company's vulnerabilities, including an understanding of a previously unknown threat.

Types of Security Risks

The CSI 2001 Computer Crime and Security Survey[2] confirms that the threat from computer crime and other information security breaches continues unabated and that the financial toll is mounting. According to the survey, 85 percent of respondents had detected computer security breaches within the past 12 months; and the total amount of financial loss reported by those who could quantify the loss amounted to $377,828,700 — that is, over $2 million per event.

One logical method for categorizing financial loss is to separate loss into three general areas of risk:

1. *First-party financial risk:* direct financial loss not arising from a third- party claim (called first-party security risks).
2. *Third-party financial risk:* a company's legal liabilities to others (called third-party security risks).
3. *Reputation risk:* the less quantifiable damages such as those arising from a loss of reputation and brand identity. These risks, in turn, arise from the particular cyber-activities. Cyber-activities can include a Web site presence, e-mail, Internet professional services such as Web design or hosting, network data storage, and E-commerce (i.e., purchase or sale of goods and services over the Internet).

First-party security risks include financial loss arising from damage, destruction, or corruption of a company's information assets — that is, data. Information assets — whether in the form of customer lists and privacy information, business strategies, competitor information, product formulas, or other trade secrets vital to the success of a business — are the real assets of the 21st century. Their proper protection and quantification are key to a successful company. Malicious code transmissions and computer viruses — whether launched by a disgruntled employee, overzealous competitor, cyber-criminal, or prankster — can result in enormous costs of recollection and recovery.

A second type of first-party security risk is the risk of revenue loss arising from a successful denial-of-service (DoS) attack. According to the Yankee Group, in February 2000 a distributed DoS attack was launched against some of the most sophisticated Web sites, including Yahoo, Buy.com, CNN, and others, resulting in $1.2 billion in lost revenue and related damages. Finally, first-party security risk can arise from the theft of trade secrets.

Third-party security risk can manifest itself in a number of different types of legal liability claims against a company, its directors, officers, or employees. Examples of these risks can arise from the company's presence on the Web, its rendering of professional services, the transmission of malicious code or a DoS attack (whether or not intentional), and theft of the company's customer information.

The very content of a company's Web site can result in allegations of copyright and trademark infringement, libel, or invasion of privacy claims. The claims need not even arise from the visual part of a Web page but can, and often do, arise out of the content of a site's metatags — the invisible part of a Web page used by search engines.

If a company renders Internet-related professional services to others, this too can be a source of liability. Customers or others who allege that such services, such as Web design or hosting, were rendered in a negligent manner or in violation of a contractual agreement may find relief in the court system.

Third-party claims can directly arise from a failure of security. A company that negligently or through the actions of a disgruntled employee transmits a computer virus to its customers or other e-mail recipients may be open to allegations of negligent security practices. The accidental transmission of a DoS attack can pose similar legal liabilities. In addition, if a company has made itself legally obligated to keep its Web site open on a 24/7 basis to its customers, a DoS attack shutting down the Web site could result in claims by its customers.

EXHIBIT 69.3 First- and Third-Party Risks

Activity	First-Party Risk	Third-Party Risk
Web site presence	Damage or theft of data (assumes database is connected to network) via hacking	Allegations of trademark, copyright, libel, invasion of privacy, and other Web content liabilities
E-mail	Damage or theft of data (assumes database is connected to network) via computer virus; shutdown of network via DoS attack	Transmission of malicious code (e.g., NIMDA) or DoS due to negligent network security; DoS customer claims if site is shut down due to DoS attack
E-commerce	Loss of revenue due to successful DoS attack	Customer suits
Internet professional services		Customer suits alleging negligent performance of professional services
Any		Claims against directors and officers for mismanagement

A wise legal department will make sure that the company's customer agreements specifically permit the company to shut down its Web site for any reason at any time without incurring legal liability.

Other potential third-party claims can arise from the theft of customer information such as credit card information, financial information, health information, or other personal data. For example, theft of credit card information could result in a variety of potential lawsuits, whether from the card-issuing companies that then must undergo the expense of reissuing, the cardholders themselves, or even the Web merchants who later become the victims of the fraudulent use of the stolen credit cards. As discussed later, certain industries such as financial institutions and healthcare companies have specific regulatory obligations to guard their customer data.

Directors and officers (D&Os) face unique, and potentially personal, liabilities arising out of their fiduciary duties. In addition to case law or common-law obligations, D&Os can have obligations under various statutory laws such as the Securities Act of 1933 and the Securities & Exchange Act of 1934. Certain industries may also have specific statutory obligations such as those imposed on financial institutions under the Gramm–Leach–Bliley Act (GLBA), discussed in detail later.

Perhaps the most difficult and yet one of the most important risks to understand is the intangible risk of damage to the company's reputation. Will customers give a company their credit card numbers once they read in the paper that a company's database of credit card numbers was violated by hackers? Will top employees remain at a company so damaged? And what will be the reaction of the company's shareholders? Again, the best way to analyze reputation risk is to attempt to quantify it. What is the expected loss of future business revenue? What is the expected loss of market capitalization? Can shareholder class or derivative actions be foreseen? And, if so, what can the expected financial cost of those actions be in terms of legal fees and potential settlement amounts?

The risks just discussed are summarized in Exhibit 69.3.

Threats

The risks defined above do not exist in a vacuum. They are the product of specific threats, operating in an environment featuring specific vulnerabilities that allow those threats to proceed uninhibited. Threats may be any person or object, from a disgruntled employee to an act of nature, that may lead to damage or value loss for an enterprise. While insurance may be used to minimize the costs of a destructive event, it is not a substitute for controls on the threats themselves.

Threats may arise from external or internal entities and may be the product of intentional or unintentional action. External entities comprise the well-known sources — hackers, virus writers — as well as less obvious ones such as government regulators or law enforcement entities. Attackers may attempt to penetrate IT systems through various means, including exploits at the system, server, or application layers. Whether the intent is to interrupt business operations, or to directly acquire confidential data or access to trusted systems, the cost in system downtime, lost revenue, and system repair and redesign can be crippling to any enterprise. The collapse

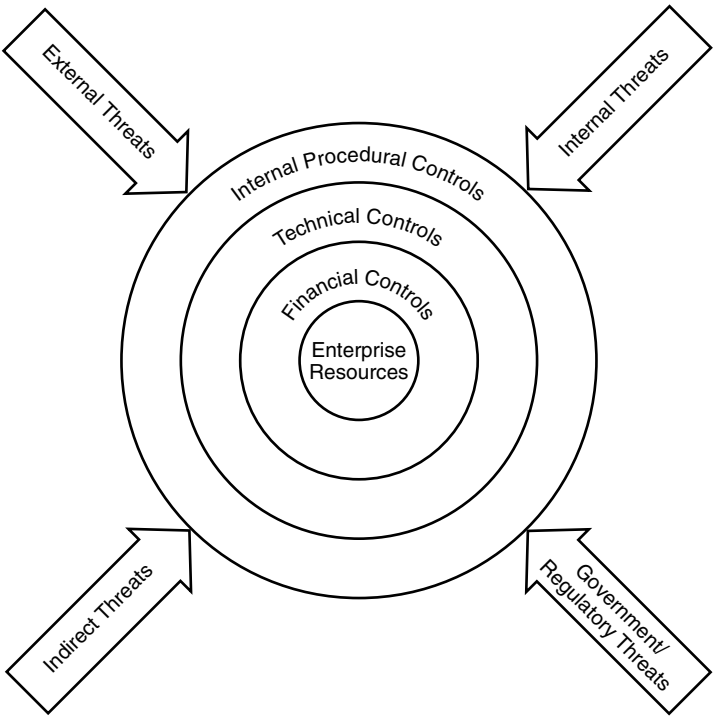


EXHIBIT 69.4 Enterprise resource threats.

of the British Internet service provider (ISP) Cloud-Nine in January 2002, due to irreparable damage caused by distributed DoS attacks launched against its infrastructure, is only a recent example of the enterprise costs of cyber-attacks.[3]

Viruses and other malicious code frequently use the same exploits as human attackers to gain access to systems. However, as viruses can replicate and spread themselves without human intervention, they have the potential to cause widespread damage across an internal network or the Internet as a whole.

Risks may arise from non-human factors as well. For example, system outages through failures at the ISP level, power outages, or natural disasters may create the same loss of service and revenue as attackers conducting DoS attacks. Therefore, technical controls should be put in place to minimize those risks. These risks are diagrammed in Exhibit 69.4.

Threats that originate from within an organization can be particularly difficult to track. This may entail threats from disgruntled employees (or ex-employees), or mistakes made by well-meaning employees as well. Many standard technical controls — firewalls, anti-virus software, or intrusion detection — assume that the internal users are working actively to support the security infrastructure. However, such controls are hardly sufficient against insiders working actively to subvert a system. Other types of risks — for example, first-party risks of intellectual property violations — may be created by internal entities without their knowledge. Exhibit 69.5 describes various threats by type.

As noted, threats are comprised of motive, access, and opportunity — outsiders must have a desire to cause damage as well as a means of affecting the target system. While an organization's exposure to risk can never be completely eliminated, all steps should be taken to minimize exposure and limit the scope of damage. Such vulnerabilities may take a number of forms.

Technical vulnerabilities include exploits against systems at the operating system, network, or application level. Given the complexity and scope of many commercial applications, vulnerabilities within code become increasingly difficult to detect and eradicate during the testing and quality assurance (QA) processes. Examples range from the original Internet Worm to recently documented vulnerabilities in commercial instant messaging clients and Web servers. Such weaknesses are an increasing risk in today's highly interconnected environments.

EXHIBIT 69.5 Threat Matrix

	Threat	Description	Security Risk	Controls
External	System penetration (external source)	Attempts by external parties to penetrate corporate resources to modify or delete data or application systems	Moderate	Strong authentication; strong access control; ongoing system support and tracking
	Regulatory action	Regulatory action or investigation based on corporate noncompliance with privacy and security guidelines	Low to moderate	Data protection; risk assessment and management programs; user training; contractual controls
	Virus penetration	Malicious code designed to self-replicate	Moderate	Technological: anti-virus controls
	Power loss or connectivity loss	Loss of Internet connectivity, power, cooling system; may result in large-scale system outages	Low	Redundant power and connectivity; contractual controls with ISP/hosting facilities
Internal	Intellectual property violation	Illicit use of third-party intellectual property (images, text, code) without appropriate license arrangements	Low to moderate	Procedural and personnel controls; financial controls mitigating risk
	System penetration (internal source)	Malicious insiders attempting to access restricted data	Moderate	Strong authentication; strong access control; use of internal firewalls to segregate critical systems

Weaknesses within operating procedures may expose an enterprise to risk not controlled by technology. Proper change management processes, security administration processes, and human resources controls and oversight, for example, are necessary. They may also prove disruptive in highly regulated environments, such as financial services or healthcare, in which regulatory agencies require complete sets of documentation as part of periodic auditing requirements.

GLBA/HIPAA

Title V of the Gramm–Leach–Bliley Act (GLBA) has imposed new requirements on the ways in which financial services companies handle consumer data. The primary focus of Title V, and the area that has received the most attention, is the sharing of personal data among organizations and their unaffiliated business partners and agencies. Consumers must be given notice of the ways in which their data is used and must be given notice of their right to opt out of any data-sharing plan.

However, Title V also requires financial services organizations to provide adequate security for systems that handle customer data. Security guidelines require the creation and documentation of detailed data security programs addressing both physical and logical access to data, risk assessment, and mitigation programs, and employee training in the new security controls. Third-party contractors of financial services firms are also bound to comply with the GLBA regulations.

On February 1, 2001, the Department of the Treasury, Federal Reserve System, and Federal Deposit Insurance Corporation issued interagency regulations, in part requiring financial institutions to:

- Develop and execute an information security program.
- Conduct regular tests of key controls of the information security program. These tests should be conducted by an independent third party or staff independent of those who develop or maintain the program.
- Protect against destruction, loss, or damage to customer information, including encrypting customer information while in transit or storage on networks.
- Involve the board of directors, or appropriate committee of the board, to oversee and execute all of the above.

Because the responsibility for developing specific guidelines for compliance was delegated to the various federal and state agencies that oversee commercial and financial services (and some are still in the process of being issued), it is possible that different guidelines for GLBA compliance will develop between different states and different financial services industries (banking, investments, insurance, etc.).

The Health Insurance Portability and Accountability Act (HIPAA) will force similar controls on data privacy and security within the healthcare industry. As part of HIPAA regulations, healthcare providers, health plans, and clearinghouses are responsible for protecting the security of client health information. As with GLBA, customer medical data is subject to controls on distribution and usage, and controls must be established to protect the privacy of customer data. Data must also be classified according to a standard classification system to allow greater portability of health data between providers and health plans. Specific guidelines on security controls for medical information have not been issued yet. HIPAA regulations are enforced through the Department of Health and Human Services.

As GLBA and HIPAA regulations are finalized and enforced, regulators will be auditing those organizations that handle medical or financial data to confirm compliance with their security programs. Failure to comply can be classified as an unfair trade practice and may result in fines or criminal action. Furthermore, firms that do not comply with privacy regulations may leave themselves vulnerable to class-action lawsuits from clients or third-party partners. These regulations represent an entirely new type of exposure for certain types of organizations as they increase the scope of their IT operations.

Cyber-Terrorism

The potential for cyber-terrorism deserves special mention. After the attacks of 9/11/01, it is clear that no area of the world is protected from a potential terrorist act. The Internet plays a critical role in the economic stability of our national infrastructure. Financial transactions, running of utilities and manufacturing plants, and much more are dependent upon a working Internet. Fortunately, companies are coming together in newly formed entities such as ISACs (Information Sharing and Analysis Centers) to determine their interdependency vul-

nerabilities and plan for the worst. It is also fortunate that the weapons used by a cyber-terrorist do not differ much from those of a cyber-criminal or other hacker. Thus, the same risk management formula discussed above should be implemented for the risk of cyber-terrorism.

Insurance for Cyber-Risks

Insurance, when properly placed, can serve two important purposes. First, it can provide positive reinforcement for good behavior by adjusting the availability and affordability of insurance depending upon the quality of an insured's Internet security program. It can also condition the continuation of such insurance on the maintenance of that quality. Second, insurance will transfer the financial risk of a covered event from a company's balance sheet to that of the insurer.

The logical first step in evaluating potential insurance solutions is to review the company's traditional insurance program, including its property (including business interruption) insurance, comprehensive general liability (CGL), directors and officers insurance, professional liability insurance, and crime policies. These policies should be examined in connection with a company's particular risks (see above) to determine whether any gap exists. Given that these policies were written for a world that no longer exists, it is not surprising that traditional insurance policies are almost always found to be inadequate to address today's cyber-needs. This is not due to any *defect* in these time-honored policies but simply due to the fact that, with the advent of the new economy risks, there comes a need for specialized insurance to meet those new risks.

One of the main reasons why traditional policies such as property and CGL do not provide much coverage for cyber-risks is their approach that *property* means *tangible property and not data*. Property policies also focus on *physical* perils such as fire and windstorm. Business interruption insurance is sold as part of a property policy and covers, for example, lost revenue when your business burns down in a fire. It will not, however, cover E-revenue loss due to a DoS attack. Even computer crime policies usually do not cover loss other than for money, securities, and other *tangible* property. This is not to say that traditional insurance can *never* be helpful with respect to cyber-risks. A mismanagement claim against a company's directors and officers arising from cyber-events will generally be covered under the company's directors' and officers' insurance policy to the same extent as a non-cyber claim. For companies that render professional services to others for a fee, such as financial institutions, those that fail to reasonably render those services due to a cyber-risk may find customer claims to be covered under their professional liability policy. (Internet professional companies should still seek to purchase a specific Internet professional liability insurance policy.)

Specific Cyber-Liability and Property Loss Policies

The inquiry detailed above illustrates the extreme dangers associated with relying upon traditional insurance policies to provide broad coverage for 21st-century cyber-risks. Regrettably, at present there are only a few specific policies providing expressed coverage for all the risks of cyberspace listed at the beginning of this chapter. One should be counseled against buying an insurance product simply because it has the name *Internet* or *cyber* in it. So-called Internet insurance policies vary widely, with some providing relatively little *real* coverage. A properly crafted Internet risk program should contain multiple products within a *suite concept* permitting a company to choose which risks to cover, depending upon where it is in its Internet maturity curve.[4] A suite should provide at least six areas of coverage, as shown in Exhibit 69.6.

These areas of coverage may be summarized as follows:

- *Web content liability* provides coverage for claims arising out of the content of your Web site (including the invisible metatags content), such as libel, slander, copyright, and trademark infringement.
- *Internet professional liability* provides coverage for claims arising out of the performance of professional services. Coverage usually includes both Web publishing activities as well as pure Internet services such as being an ISP, host, or Web designer. Any professional service conducted over the Internet can usually be added to the policy.
- *Network security coverage* comes in two basic types:
  - *Third-party coverage* provides liability coverage arising from a failure of the insured's security to prevent unauthorized use of or access to its network. This important coverage would apply, subject to the policy's full terms, to claims arising from the transmission of a computer virus (such as the Love Bug or Nimda virus), theft of a customer's information (most notably including credit card

EXHIBIT 69.6 First- and Third-Party Coverage

	First-Party Coverage	Third-Party Coverage
Media		Web content liability
E&O		Professional liability
Network security	Cyber-attack caused damage, destruction and corruption of data, theft of trade secrets or E-revenue business interruption	Transmission of a computer virus or DoS liability; theft of customer information liability; DoS customer liability
Cyber-extortion	Payment of cyber-investigator	Payment of extortion amount where appropriate
Reputation	Payment of public relations fees up to $50,000	
Criminal reward	Payment of criminal reward fund up to $50,000	

information), and so-called denial-of-service liability. In the past year alone, countless cases of this type of misconduct have been reported.

— *First-party coverage* provides, upon a covered event, reimbursement for loss arising out of the altering, copying, misappropriating, corrupting, destroying, disrupting, deleting, damaging, or theft of information assets, whether or not criminal. Typically the policy will cover the cost of replacing, reproducing, recreating, restoring, or recollecting. In case of theft of a trade secret (a broadly defined term), the policy will either pay or be capped at the endorsed negotiated amount. First-party coverage also provides reimbursement for lost E-revenue as a result of a covered event. Here, the policy will provide coverage for the period of recovery plus an extended business interruption period. Some policies also provide coverage for dependent business interruption, meaning loss of E-revenue as a result of a computer attack on a third-party business (such as a supplier) upon which the insured's business depends.

- *Cyber-extortion coverage* provides reimbursement of investigation costs, and sometimes the extortion demand itself, in the event of a covered cyber-extortion threat. These threats, which usually take the form of a demand for "consulting fees" to prevent the release of hacked information or to prevent the extortion from carrying out a threat to shut down the victims' Web sites, are all too common.
- *Public relations or crisis communication coverage* provides reimbursement up to $50,000 for use of public relation firms to rebuild an enterprise's reputation with customers, employees, and shareholders following a computer attack.
- *Criminal reward funds coverage* provides reimbursement up to $50,000 for information leading to the arrest and conviction of a cyber-criminal. Given that many cyber-criminals hack into sites for "bragging rights," this unique insurance provision may create a most welcome chilling effect.

Loss Prevention Services

Another important feature of a quality cyber-risk insurance program is its loss prevention services. Typically these services could include anything from free online self-assessment programs and free educational CDs to a full-fledged, on-site security assessment, usually based on ISO 17799. Some insurers may also add other services such as an internal or external network scan. The good news is that these services are valuable, costing up to $50,000. The bad news is that the insurance applicant usually has to pay for the services, sometimes regardless of whether or not it ends up buying the policy. Beginning in 2001, one carrier has arranged to pay for these services as part of the application process. This is welcome news. It can only be hoped that more insurers will follow this lead.

Finding the Right Insurer

As important as finding the right insurance product is finding the right insurer. Financial strength, experience, and claims philosophy are all important. In evaluating insurers, buyers should take into consideration the factors listed in Exhibit 69.7.

EXHIBIT 69.7 Finding the Right Insurer

Quality	Preferred or Minimum Threshold
Financial strength	Triple-A from Standard & Poor's
Experience	At least two years in dedicated, specialized unit composed of underwriters, claims, technologists, and legal professionals
Capacity	Defined as amount of limits single carrier can offer; minimum acceptable: $25,000,000
Territory	Global presence with employees and law firm contacts throughout the United States, Europe, Asia, Middle East, South America
Underwriting	Flexible, knowledgeable
Claims philosophy	Customer focused; willing to meet with client both before and after claim
Policy form	Suite permitting insured to choose right coverage including eight coverages described above
Loss prevention	Array of services, most importantly including FREE on-site security assessments conducted by well-established third-party (worldwide) security assessment firms

In summary, traditional insurance is not up to the task of dealing with today's cyber-risks. To yield the full benefits, insurance programs should provide and implement a purchase combination of traditional and specific cyber-risk insurance.

Technical Controls

Beyond insurance, standard technical controls must be put in place to manage risks. First of all, the basic physical infrastructure of the IT data center should be secured against service disruptions caused by environmental threats. Organizations that plan to build and manage their own data centers should implement fully redundant and modular systems for power, Internet access, and cooling. For example, data centers should consider backup generators in case of area-wide power failures, and Internet connectivity from multiple ISPs in case of service outages from one provider.

In cases where the customer does not wish to directly manage its data center, the above controls should be verified before contracting with an ASP or ISP. These controls should be guaranteed contractually, as should failover controls and minimum uptime requirements.

Physical Access Control

Access control is an additional necessity for a complete data center infrastructure. Physical access control is more than simply securing entrances and exits with conventional locks and security guards. Secure data centers should rely on alarm systems and approved locks for access to the most secure areas, with motion detectors throughout. More complex security systems, such as biometric[5] or dual-factor authentication (authentication requiring more than one proof of identity; e.g., card and biometric), should be considered for highly secure areas. Employee auditing and tracking for entrances and exits should be put in place wherever possible, and visitor and guest access should be limited. A summary of potential controls is provided in Exhibit 69.8.

If it is feasible to do so, outside expertise in physical security, like logical security, should be leveraged wherever possible. Independent security audits may provide insight regarding areas of physical security that are not covered by existing controls. Furthermore, security reports may be required by auditors, regulators, and other third parties. Audit reports and other security documentation should be kept current and retained in a secure fashion.

Again, if an organization uses outsourced facilities for application hosting and management, it should look for multilevel physical access control. Third-party audit reports should be made available as part of the vendor search process; security controls should be made part of the evaluation criteria. As with environmental controls, access controls should also be addressed within the final service agreement such that major modifications to the existing access control infrastructure require advance knowledge and approval. Organizations should insist on periodic audits or third-party reviews to ensure compliance.

EXHIBIT 69.8 Physical Controls

Physical Control	Description	Role
Access control	Grants access to physical resources through possession of keys, cards, biometric indicators, or key combinations; multi-factor authentication may be used to increase authentication strength; access control system that requires multiple-party authentication provide higher levels of access control	Securing data center access in general, as well as access to core resources such as server rooms; media — disks, CD-ROMs, tapes — should be secured using appropriate means as well; organizations should model their access control requirements on the overall sensitivity of their data and applications
Intrusion detection	Detection of attempted intrusion through motion sensors, contact sensors, and sensors at standard access points (doors, windows, etc.)	At all perimeter access points to the data center, as well as in critical areas
24/7 Monitoring	Any data center infrastructure should rely on round-the-clock monitoring, through on-premises personnel and off-site monitoring	Validation to existing alarm and access control systems

Network Security Controls

A secure network is the first layer of defense against risk within an E-business system. Network-level controls are instrumental in preventing unauthorized access from within and without, and tracking sessions internally will detect and alert administrators in case of system penetration. Exhibit 69.9 conceptually depicts the overall architecture of an E-business data center.

Common network security controls include the following features.

Firewalls

Firewalls are critical components of any Internet-facing system. Firewalls filter network traffic based on protocol, destination port, or packet content. As firewall systems have become more advanced, the range of different attack types that can be recognized by the firewall has continued to grow. Firewalls may also be upgraded to filter questionable content or scan incoming traffic for attack signatures or illicit content.

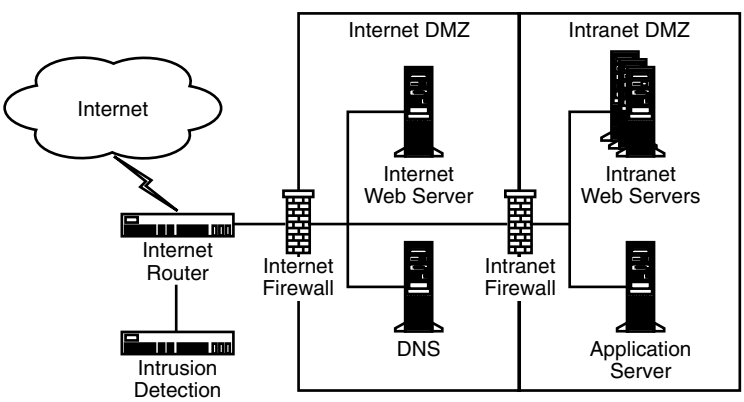


EXHIBIT 69.9 Demilitarized zone architecture.

EXHIBIT 69.10 Secure Network Design Principles

Redundancy. Firewall systems, routers, and critical components such as directory servers should be fully redundant to reduce the impact of a single failure.

Currency. Critical network tools must be kept up-to-date with respect to patch-level and core system operations. Vulnerabilities are discovered frequently, even within network security devices such as firewalls or routers.

Scalability. An enterprise's network security infrastructure should be able to grow as business needs require. Service outages caused by insufficient bandwidth provided by an ISP, or server outages due to system maintenance, can be fatal for growing applications. The financial restitution provided by cyber-risk coverage might cover business lost during the service outage but cannot address the greater issues of loss of business, consumer goodwill, or reputation.

Simplicity. Complexity of systems, rules, and components can create unexpected vulnerabilities in commercial systems. Where possible, Internet-facing infrastructures should be modularized and simplified such that each component is not called upon to perform multiple services. For example, an organization with a complex E-business infrastructure should separate that network environment from its own internal testing and development networks, with only limited points of access between the two environments. A more audited and restricted set of rules may be enforced in the former without affecting the productivity of the latter.

For any infrastructure that requires access to business data, a multiple-firewall configuration should be used. An Internet demilitarized zone (DMZ) should be created for all Web-accessible systems — Web servers or DNS servers — while an intranet DMZ, separated from the Internet, contains application and database servers. This architecture prevents external entities from directly accessing application logic or business data.

Network Intrusion Detection Systems

Networked IDSs track internal sessions at major network nodes and look for attack signatures — a sequence of instructions corresponding to a known attack. These systems generally are also tied into monitoring systems that can alert system administrators in the case of detected penetration. More advanced IDSs look for only "correct" sequences of packets and use real-time monitoring capabilities to identify suspicious but unknown sequences.

Anti-Virus Software

Anti-virus gateway products can provide a powerful second level of defense against worms, viruses, and other malicious code. Anti-virus gateway products, provided by vendors such as Network Associates, Trend Micro, and Symantec, can scan incoming HTTP, SMTP, and FTP traffic for known virus signatures and block the virus before it infects critical systems.

As described in Exhibit 69.10, specific design principles should be observed in building a stable and secure network. Exhibit 69.11 provides a summary of the controls in question.

Increasingly, organizations are moving toward managed network services rather than supporting the systems internally. Such a solution saves the organization from having to build staff for managing security devices, or to maintain a 24/7 administration center for monitoring critical systems. Such a buy (or, in this case, hire) versus build decision should be seriously considered in planning your overall risk management framework. Organizations looking to outsource security functions can certainly save money, resources, and time; however, organizations should look closely at the financial as well as technical soundness of any such vendors.

Application Security Controls

A successful network security strategy is only useful as a backbone to support the development of secure applications. These controls entail security at the operating system level for enterprise systems, as well as trust management, encryption, data security, and audit controls at the application level.

Operating systems should be treated as one of the most vulnerable components of any application framework. Too often, application developers create strong security controls within an application, but have no control over

EXHIBIT 69.11 Network Security Controls

Network Control	Description	Role
Firewall	Blocks connections to internal resources by protocol, port, and address; also provides stateful packet inspection	Behind Internet routers; also within corporate networks to segregate systems into DMZs
IDS	Detects signature of known attacks at the network level	At high-throughput nodes within networks, and at perimeter of network (at firewall level)
Anti-virus	Detects malicious code at network nodes	At Internet HTTP and SMTP gateways

the lower level exploits. Furthermore, system maintenance and administration over time is frequently overlooked as a necessary component of security. Therefore, the following controls should be observed:

- Most major OS suppliers — Microsoft, Sun, Hewlett-Packard, etc. — provide guidelines for operating system hardening. Implement those guidelines on all production systems.
- Any nonessential software should be removed from production systems.
- Administer critical servers from the system console wherever possible. Remote administration should be disabled; if this is not possible, secure log-in shells should be used in place of less secure protocols such as Telnet.
- Host-based intrusion detection software should be installed on all critical systems. A host-based IDS is similar to the network-based variety, except it only scans traffic intended for the target server. Known attack signatures may be detected and blocked before reaching the target application, such as a Web or application server.

Application-level security is based on maintaining the integrity and confidentiality of the system as well as the data managed by the system. A Web server that provides promotional content and brochures to the public, for example, has little need to provide controls on confidentiality. However, a compromise of that system resulting in vandalism or server downtime could prove costly; therefore, system and data integrity should be closely controlled. These controls are partially provided by security and the operating system and network levels as noted above; additional controls, however, should be provided within the application itself.

Authentication and authorization are necessary components of application-level security. Known users must be identified and allowed access to the system, and system functions must be categorized such that users are only presented with access to data and procedures that correspond to their defined privilege level.

The technical controls around authentication and authorization are only as useful as the procedural controls around user management. The enrollment of new users, management of personal user information and usage profiles, password management, and the removal of defunct users from the system are required for an authentication engine to provide real risk mitigation.

Exhibit 69.12 provides a summary of these technologies and procedures.

Data Backup and Archival

In addition to technologies to prevent or detect unauthorized system penetration, controls should be put in place to restore data in the event of loss. System backups — onto tape or permanent media — should be in place for any business-critical application.

Backups should be made regularly — as often as daily, depending on the requirements of the business — and should be stored off-site to prevent loss or damage. Test restores should also be performed regularly to ensure the continued viability of the backup copies. Backup retention should extend to at least a month, with one backup per week retained for a year and monthly backups retained for several years. Backup data should always be created and stored in a highly secure fashion.

Finally, to ensure system availability, enterprise applications should plan on at least one tier of redundancy for all critical systems and components. Redundant systems can increase the load-bearing capacity of a system as well as provide increased stability. The use of enterprise-class multi-processor machines is one solution; multiple systems can also be consolidated into server farms. Network devices such as firewalls and routers can

EXHIBIT 69.12 Application Security Controls

Application Control	Description	Role
System hardening	Processes, procedures, and products to harden operating system against exploitation of network services	Should be performed for all critical servers and internal systems
Host-based intrusion detection	Monitors connections to servers and detects malicious code or attack signatures	On all critical servers and internal systems
Authentication	Allows for identification and management of system users through identities and passwords	For any critical systems; authentication systems may be leveraged across multiple applications to provide single sign-on for enterprise
Access control	Maps users, by identity or by role, to system resources and functions	For any critical application
Encryption	Critical business data or non-public client information should be encrypted (i.e., obscured) while in transit over public networks	For all Internet-based transactional connectivity; encryption should also be considered for securing highly sensitive data in storage

also be made redundant through load balancers. Businesses may also wish to consider maintaining standby systems in the event of critical data center failure. Standby systems, like backups, should be housed in a separate storage facility and should be tested periodically to ensure stability. These backup systems should be able to be brought online within 48 hours of a disaster and should be restored with the most recently available system backups as well.

Conclusion

The optimal model to address the risks of Internet security must combine technology, process, and insurance. This risk management approach permits companies to successfully address a range of different risk exposures, from direct attacks on system resources to unintentional acts of copyright infringement. In some cases, technical controls have been devised that help address these threats; in others, procedural and audit controls must be implemented. Because these threats cannot be completely removed, however, cyber-risk insurance coverage represents an essential tool in providing such nontechnical controls and a major innovation in the conception of risk management in general. A comprehensive policy backed by a specialized insurer with top financial marks and global reach allows organizations to lessen the damage caused by a successful exploit and better manage costs related to loss of business and reputation. It is only through merging the two types of controls that an organization can best minimize its security threats and mitigate its IT risks.

Notes

1. The views and policy interpretations expressed in this work by the authors are their own and do not necessarily represent those of American International Group, Inc., or any of its subsidiaries, business units, or affiliates.
2. See http://www.gocsi.com for additional information.
3. Coverage provided in *ISPreview,* ZDNet.
4. One carrier's example of this concept can be found at www.aignetadvantage.com.
5. Biometrics authentication comprises many different measures, including fingerprint scans, retinal or iris scans, handwriting dynamics, and facial recognition.

70

A Progress Report on the CVE Initiative

Robert Martin, Steven Christey, and David Baker

Common Vulnerabilities and Exposures (CVE) is an international, community-based effort, including industry, government, and academia, that is working to create an organizing mechanism to make identifying, finding, and fixing software product vulnerabilities more rapid and efficient. A few years ago, each of us was faced with a cacophony of naming methods for defining individual security problems in software. This made it difficult to assess, manage, and fix vulnerabilities and exposures when using the various vulnerability services, tools, and databases along with the software suppliers' update announcements and alerts. For example, Exhibit 70.1 shows how in 1998 each of a dozen leading organizations used different names to refer to the same well-known vulnerability in the phf phonebook CGI program. Such confusion made it difficult to understand which vulnerabilities an organization faced and which ones each tool was looking for (or not looking for). Then, to get the fix to the identified vulnerability, users still had to figure out what name the vulnerability or exposure was assigned by their software supplier.

Driven by a desire to develop an integrated picture of what was happening on its corporate networks, and while trying to properly research options for selecting some new network security tools, the MITRE Corporation[1] (http://www.mitre.org) began designing a method to sort through this vulnerability naming confusion. The approach involved the creation of a unified reference list of vulnerability and exposure names that were mapped to the equivalent items in each tool and database. In January 1999, MITRE presented a paper at the 2nd Workshop on Research with Security Vulnerability Databases at Purdue University[2] that outlined the concept and approach for what today is known as the Common Vulnerabilities and Exposures Initiative (http://cve.mitre.org). The primary product of this Initiative is the CVE List, a reference list of standard names for vulnerabilities and exposures.

The CVE List was envisioned as a simple mechanism for linking vulnerability-related databases, tools, and concepts. It was believed to be critical for the information security community to concur with the CVE approach and begin incorporating the common names into their various products and services. Therefore, CVE's role was limited to that of a logical bridge to avoid competing with existing and future commercial efforts.

Although the CVE name itself was simple in concept, there would be nothing simple about implementing the CVE Initiative. To be successful, all existing vulnerability information would have to be examined and compared to determine which parts of this overall set of information referred to the same problem. Then, unique and consistent descriptions for each problem would have to be created, and the technical leaders of the information security community would have to be brought together to agree on the descriptions. The CVE List would have to be broadly distributed for commercial vendors and researchers to adopt it. A CVE compatibility evaluation process would have to be designed to verify vendor claims of support for the CVE names in products and services, and policies would have to be created to encourage the use of CVE-compatible products. The CVE Initiative would also have to be an ongoing effort because new vulnerabilities are always being discovered, and at an increasing rate. Finally, the CVE Initiative had to include international participation in both those helping with the development of the CVE List, and by the vendor community and other organizations using the common names in their products and services.

EXHIBIT 70.1 Vulnerability Tower of Babel, 1998

Organization	Name Referring to Vulnerability
AXENT (now Symantec)	phf CGI allows remote command execution
BindView	#107 — cgi-phf
Bugtraq	PHF Attacks — fun and games for the whole family
CERIAS	http_escshellcmd
CERT	CA-96.06.cgi_example_code
Cisco Systems	HTTP — cgi-phf
CyberSafe	Network: HTTP "phf" attack
DARPA	0x00000025 = HTTP PHF attack
IBM ERS	ERS-SVA-E01-1996:002.1
ISS	http — cgi-phf
Symantec	#180 HTTP server CGI example code compromises http server
SecurityFocus	#629 — phf Remote Command Execution Vulnerability

To guide the various aspects of the CVE Initiative to enable the adoption of the CVE List as a common mechanism for referring to vulnerabilities and exposures, CVE has targeted five specific areas of activity, to include:

1. Uniquely naming every publicly known information security vulnerability and exposure
2. Injecting CVE names into security and vendor advisories
3. Establishing CVE usage in information security products as common practice
4. Having CVE usage permeate policy guidelines about methodologies and purchasing, included as requirements for new capabilities, and introducing CVE into training, education, and best practices suggestions
5. Convincing commercial software developers to use CVE names in their fix-it sites and update mechanisms

The remainder of this chapter describes the various challenges, solutions, and approaches that the CVE Initiative has undertaken (or faced) in the development of the various elements of the CVE Initiative.

Implementing the CVE Initiative

After a positive response from the Purdue CERIAS Workshop, MITRE formed the CVE Editorial Board in May 1999 with 12 commercial vendor and research organizations, which worked to come to agreement on the initial CVE List with MITRE as moderator. During this same time, a MITRE team worked to develop a public Web site to host the CVE List, archive discussions of the Editorial Board, and host declarations of vendor intent to make products CVE-compatible. The CVE Initiative was publicly unveiled in September 1999. The unveiling included an initial list of 321 entries, a press release teleconference, and a CVE booth that was staffed with the Editorial Board members at the SANS 1999 technical conference. It was a very powerful message to attendees to see the CVE booth staffed by competing commercial vendors working together to solve an industry problem. There was a large audience of system administrators and security specialists in attendance, who had been dealing with the same problem that motivated the creation of the CVE Initiative.

As the volume of incoming vulnerability information increased for both new and legacy issues, MITRE established a content team to help with the job of generating CVE content. The roles and responsibilities of the Editorial Board were formalized. MITRE worked with vendors to put CVE names in security advisories as vulnerabilities were announced, and worked with the CVE Senior Advisory Council to develop policy recommending the use of CVE-compatible products and services and to find ways of funding the CVE Initiative for the long term. Since the beginning, MITRE has promoted the CVE Initiative in and at various venues, including hosting booths at industry tradeshows, interviewing with the media, publishing CVE-focused articles in national and international journals,[3] and presenting CVE-focused talks in public forums and conferences.

The CVE List

The CVE Initiative has had to address many different perspectives, desires, and needs as it developed the CVE List. The common names in the CVE List are the result of open and collaborative discussions of the CVE Editorial Board (a deeper discussion of the Board can be found later in this chapter), along with various supporting and facilitating activities by MITRE and others. With MITRE's support, the Board identifies which vulnerabilities or exposures to include on the CVE List and agrees on the common name, description, and references for each entry. MITRE maintains the CVE List and Web site, moderates Editorial Board discussions, analyzes submitted items, and provides guidance throughout the process to ensure that CVE remains objective and continues to serve the public interest.

CVE Candidates versus CVE Entries

CVE candidates are those vulnerabilities or exposures under consideration for acceptance into the official CVE List. Candidates are assigned special numbers that distinguish them from CVE entries. Each candidate has three primary items associated with it: (1) number (also referred to as a name), (2) description, and (3) references. The number is an encoding of the year that the candidate number was assigned and a unique number N for the Nth candidate assigned that year (e.g., CAN-1999-0067). If the candidate is accepted into CVE, these numbers become CVE entries. For example, the previous candidate number would have an eventual CVE number of CVE-1999-0067, where the "CAN" prefix is replaced with the "CVE" prefix. The assignment of a candidate number is not a guarantee that it will become an official CVE entry.

Data Sources and Expansion of the CVE List

Throughout the life of the CVE List, MITRE has relied on other data sources to identify vulnerabilities. As a result, MITRE can concentrate on devising the standard names, instead of "reinventing the wheel" and conducting the research required to find the initial vulnerability reports. Before CVE was publicly released in September 1999, a "draft CVE" was created and submitted to the Editorial Board for feedback. ISS, L-3 Security (later acquired by Symantec), SANS, and Netect (later acquired by BindView) provided information that was used to help create the draft CVE. Data was also drawn from other sources, including Bugtraq and NTBugtraq posts, CERT advisories, and security tools such as NAI's CyberCop Scanner, Cisco's NetSonar, and AXENT's NetRecon.

In November 1999, two months after the first version of the CVE List was made available, MITRE asked Editorial Board members to provide a "top 100" list of vulnerabilities that should be in the CVE List, which produced more than 800 submissions. Contributing organizations were Purdue CERIAS, ISS, Harris, BindView, Hiverworld (later nCircle), Cisco, L-3 Security (later acquired by Symantec), and AXENT (later acquired by Symantec). At this time, MITRE also began processing newly discovered vulnerabilities, using the periodic vulnerability summaries published by SecurityFocus, Network Computing/SANS, ISS, and the National Infrastructure Protection Center (NIPC).

To manage the volume of vulnerabilities that were submitted, MITRE began developing the submission matching and refinement process described later in this chapter.

In the summer of 2000, MITRE again sought to expand the CVE List to include older "legacy" problems that were not in the original draft CVE, this time receiving copies of the vulnerability databases from ten organizations — a total of approximately 8400 submissions. The contributors were AXENT (now Symantec), BindView, Harris Corporation, Cisco Systems, Purdue University's Center for Education and Research in Information and Security (CERIAS), Hiverworld (now nCircle), SecurityFocus, Internet Security Systems (ISS), Network Associates, L3 (now Symantec), and the Nessus Project. These contributions were made while newly discovered issues were also being processed in parallel. In the following year, MITRE expanded its support staff and improved its processes and utilities for dealing with the increasing volume of information.

Of the 8400 legacy submissions received in the summer of 2000, MITRE has thus far eliminated 2500 submissions that duplicated existing candidates or entries, or did not meet the CVE definition of a vulnerability or exposure. An additional 3900 submissions require additional information from the source that provided them (generally due to lack of references or vague descriptions), and 1100 have been set aside for more detailed examination and study. Many of these 1100 vulnerability submissions describe insecure configurations and

require further study. Configuration problems are difficult to identify with CVE because configuration is system dependent, such problems are not as well studied as software implementation errors, and they could be described at multiple levels of abstraction. MITRE's research and analysis is currently focusing on the Windows-based portion of these configuration problems.

The remaining 900 legacy submissions formed the basis of 563 CVE candidates that were proposed to the Board in September 2001. A small number of submissions from November 1999 still remain, mostly due to the lack of sufficient information to create a candidate.

While MITRE processes the remaining legacy submissions and conducts the necessary background research, it continues to receive between 400 and 600 new submissions per month from ISS, SecurityFocus, Neohapsis, and the National Infrastructure Protection Center. Each month, an additional 20 to 70 specific candidates are reserved before a new vulnerability or exposure is publicly known, with the candidate number then included in vendor and security community member alerts and advisories.

Because there was an increased emphasis on creating legacy candidates during the summer of 2001, a backlog of submissions for recent issues developed. Candidates for those issues were to be created by early 2002, and additional processes are being implemented to avoid such backlogs in the future. One avenue that is being pursued to address this problem is the active engagement of vendors and researchers to include CVE candidate names in their initial advisories and alerts. To date, a variety of individuals and organizations have reserved more than 870 candidate numbers for use in public announcements of vulnerabilities, including ISS, IBM, Rain Forest Puppy, @stake, Microsoft, BindView, NAI, CERT/CC, SGI, eEye, COMPAQ, Ernst & Young, CISCO, Rapid 7, NSFOCUS, Sanctum, Alcatel, EnGarde Secure Linux, Caldera, Red Hat, SecurityFocus, VIGILANTe.com, Cert-IST, Mandrake Linux, Debian, Foundstone, Apple, iDEFENSE, HP, Symantec, DHS/NIPC, KDE e. V., Beyond Security Ltd., Digital Defense Inc., Core-ST, The OpenPKG Project, Corsaire, The FreeBSD Project, and Gentoo Linux.

Growth of the CVE List since Inception

As previously mentioned, the first version of the CVE List was released in September 1999; it contained 321 CVE entries that MITRE had researched and reviewed with the initial Editorial Board members. As Exhibit 70.2 shows, the number of entries in the CVE List stands at 2573 entries as of mid-May 2003, while candidates number 3463. Notable increases occurred in November 1999, September 2001, and February/March 2002 in conjunction with the growth of the list as described in the previous section. The CVE Web site now tracks

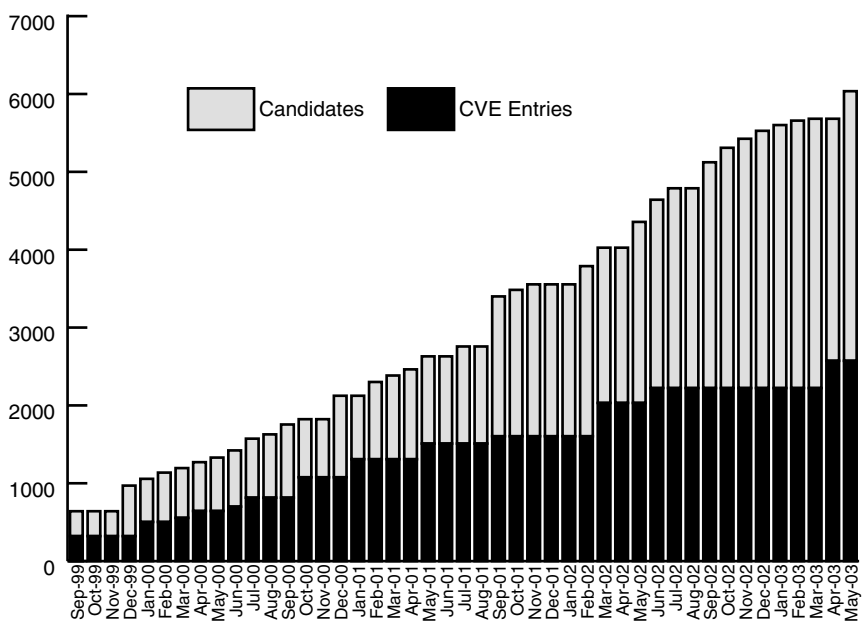


EXHIBIT 70.2 CVE growth over time.

some 6036 uniquely named vulnerabilities and exposures, which include the current CVE List, recently added legacy candidates, and the ongoing generation of new candidates from recent discoveries.

The Process of Building the CVE List: The Submission Stage*:* Stage 1

The CVE review process is divided into three stages: (1) the initial submission stage, (2) the candidate stage, and (3) the entry stage. MITRE is solely responsible for the submission stage but is dependent on its data sources for the submissions. The Editorial Board shares the responsibility for the candidate and entry stages, although the entry stage is primarily managed by MITRE as part of normal CVE maintenance.

Content Team

For the CVE project, MITRE has a content team whose primary task is to analyze, research, and process incoming vulnerability submissions from CVE's data sources, transforming the submissions into candidates. The CVE Editor, who is ultimately responsible for all CVE content, leads the team.

Conversion Phase

During the submission stage, MITRE's CVE Content Team, which consists of MITRE security analysts and researchers, collects raw information from various sources, for example, the various Board members who have provided MITRE with their databases, or publishers of weekly vulnerability summaries. Each separate item in the data source (typically a record of a single vulnerability) is then converted to a "submission," which is represented in a standardized format that facilitates processing by automated programs. Each submission includes the unique identifier that is used by the original data source.

Matching Phase

After this conversion phase, each target submission is automatically matched against all other submissions, candidates, and entries using information retrieval techniques. The matching is based primarily on keywords that are extracted from a submission's description, references, and short title. The keywords are weighted according to how frequently they appear, which generally gives preference to infrequently seen terms such as product and vendor names and specific vulnerability details. Keyword matching is not completely accurate, as there may be variations in spelling of important terms such as product names, or an anomalous term may be given a larger weight than a human would use. The closest matches for the target submission (typically ten) are then presented to a content team member, who identifies which submissions are describing the same issue (or the same set of closely related issues) as the target submission.

Once matching is complete, all related submissions are combined into submission groups, which may include any candidates or entries that were found during matching. Each group identifies a single vulnerability or a group of closely related vulnerabilities. These groups are then processed in the next phase, called "refinement."

Refinement Phase

Typically, a content team member is assigned a batch of 20 or more submission groups, which usually includes both duplicate submissions and new issues.

During refinement, the team member analyzes a submission group and determines whether one or more of the submissions identify an existing CVE item. If so, then the analyst notes any additional references that are in the new submission, but not the existing CVE item, so that the existing CVE item's references can be extended.

If there are submissions from the group that do not describe an existing CVE item, then a team member makes the following assessment:

- Apply CVE content decisions to decide whether any candidates should be created.
- Apply CVE content decisions to decide how many candidates must be created.

(Content decisions are covered in a later section.)

For each candidate to be created, the analyst does the following:

- List the associated references using CVE's canonical reference format.
- Create a description.
- Determine if there is vendor acknowledgment.
- Identify any related content decisions.

- Identify other supporting information such as the date the problem was announced, high-level operating system (OS), whether the issue is remotely or locally exploitable, and a few other attributes. This information is used to group sets of candidates later in the process, or to provide tailored voting ballots to individual Editorial Board members.
- Identify any keywords that could help in later submission matching (as well as the CVE Web site's search engine). Typically, the keywords include alternate spellings or terms that were not explicitly necessary for the description.
- Identify the rationales for acknowledgment and content decisions in the "analysis" section.

In some cases, an analyst may choose to delay analysis of a submission group (or a portion of the group) when an issue is unusually complex or if other individuals need to be consulted.

Submission refinement is a bottleneck because deep analysis is sometimes required to understand the reported problem, apply the content decisions, find vendor acknowledgment, research the references, and write the descriptions. Refinement is especially difficult for new analysts because there is a large amount of detail and background knowledge required before the analyst becomes comfortable and productive in doing refinement.

For each action that the content team member undertakes — whether identifying a duplicate, rejecting a submission, or suggesting the creation of a new candidate — a "refinement group" is produced. One or more refinement groups are produced from the original submission group, depending on how many separate issues were in the original submission group.

Editing Phase

After refinement, the CVE Editor reviews the work of the analysts, occasionally making modifications to follow CVE style, ensuring that CVE content decisions are being followed, or performing advanced research. Occasionally, submissions may be added or removed from the refinement groups. The Editor provides feedback to the analyst for the purposes of training or to raise certain issues. Because the submission matching may not always find all related submissions, typically due to spelling inconsistencies across submissions, the Editor can merge multiple refinement groups that were produced by different analysts.

The Editor then processes the resulting refinement groups. New candidate numbers are assigned to the groups that identify new issues (the "assignment" phase in the candidate stage).

After candidate assignment, each data source is provided with a backmap from their submission to the associated CVE items (whether newly created candidates, or existing candidates or entries). The backmap can reduce the amount of effort the data source needs to perform to maintain a mapping to CVE. After the backmaps for the candidates are generated, the associated submissions are removed from the submission pool. In addition to backmaps, a new type of map referred to as a "gapmap" is also provided to the information source. The gapmap identifies the newly created candidates that were not found in the data source's original set of submissions, which may make the source aware of additional security problems that they had not seen previously.

In some cases, the submission stage may be entirely bypassed. This usually happens when an individual or organization reserves a candidate number in order to include it in the initial public announcement of a vulnerability, as described in further detail in a later section.

The Process of Building the CVE List: The Candidate Stage: Stage 2

Assignment Phase

Candidates are normally created in one of three ways: they are refined by the content team using submissions from CVE's data sources; they are reserved by an organization or individual who uses it when first announcing a new issue; or they are created "out-of-band" by the CVE Editor, typically to quickly create a candidate for a new, critical issue that is being widely reported.

Proposal Phase

The CVE Editor organizes candidates into clusters of 20 to 50 candidates. For new issues, the clusters are typically grouped by the initial public announcement dates of the candidates. For legacy issues, the clusters can be created according to other criteria that make the clusters more manageable for Editorial Board members

to work with, such as UNIX vendor advisories. The candidate clusters are then proposed to the Board for review and voting.

Voting Phase

Editorial Board members review proposed candidates, registering their feedback with a vote and optional commentary. Votes include ACCEPT, MODIFY (signifying the need for a minor change), REJECT, RECAST (signifying the need for a major change), REVIEWING (member is actively reviewing the candidate but does not have a vote ready), and NOOP (no opinion). A Board member may ACCEPT or MODIFY a candidate if (1) it has been acknowledged by the vendor, (2) the issue has been replicated by the voting Board member, (3) the issue has been reported or replicated by someone whom the member trusts, or (4) there is independent confirmation from another party. MITRE is considering whether (4) is sufficient to establish the veracity of a candidate. One issue that has not yet been resolved is how to deal with "permanent" candidates that may be real but never receive enough positive votes to be accepted as official entries.

Modification Phase

The candidate can be modified based on feedback from Board members. (More information on this appears in the "Modification" section below.)

Interim Decision Phase

The CVE Editor decides when the review of a candidate is complete or has come to a standstill. The Editor casts a single ACCEPT or REJECT vote, then gives Board members a "last call" opportunity to post any final comments or objections (at least four business days). If there are extensive comments or objections that require additional voting, the candidate may be returned to the modification phase.

Final Decision Phase

If the CVE Editor determines that no sufficient grounds exist for changing the vote made in the Interim Decision, then the decision becomes final. If the candidate is ACCEPTed, the Editor announces to all Board members that the candidate will be placed into CVE, and identifies the CVE name that will be assigned to the new entry. If the candidate is REJECTed, the Editor notes the reason for rejection.

The Process of Building the CVE List: The Entry Stage: Stage 3 of 3

Publication Phase

If the candidate has been ACCEPTed, the candidate is converted into an entry by changing the name from CAN-YYYY-NNNN to CVE-YYYY-NNNN and removing the voting record. The new entry is then added to the next version of the CVE List.

Modification Phase

The entry may need to be modified in simple ways, for example, to clarify the description or add more references. (More information on this appears in the following section.)

Modifications and Deletions in the CVE List and Candidates List

Modification

Most candidates and entries are modified by adding more references (such as additional vendor advisories), or through small changes to descriptions (such as fixing typos and clarifying the issue). Candidate modifications are normally not explicitly presented to the Editorial Board or the public, due to the number and frequency of changes that take place. For entries, the Editorial Board is notified of basic modifications at least four business days before the new CVE version is targeted for creation.

For CVE users who want to track modification in the CVE List, MITRE provides "version difference reports" that detail which entries have changed, and how they have changed, between two versions. For various reasons, this capability was not made available for candidates, but the Cassandra project being led by Purdue CERIAS now offers a change monitoring report that includes changes to candidates (https://cassandra.cerias.purdue.edu/CVE_changes/).

Some modifications may be substantial. For example, a candidate may need to be split into multiple items, or multiple candidates may need to be merged into a single item (*recast*). This will happen if a content decision was not applied properly when the candidate was first created, or if new information forces such a change. In some cases, a recast may be required for entries. The procedure for recasting candidates and entries has not been completely defined, because most of these changes are due to content decisions that have not been finalized yet. However, at a minimum, the procedure of recasting a candidate or entry will include the incorporation of some type of forward pointers that will go from any recast item to the "corrected" items.

In other cases, a description and/or the set of references may be vague enough that the item could appear to describe more than one different vulnerability. This happened more frequently in the early days of CVE when the utility of references in deconflicting similar issues, and the importance of having necessary details in the descriptions, was not fully understood. Vague descriptions and missing references can lead to mapping errors in CVE-compatible products and services. Vendor security advisories with vague information present a special challenge: the issue is likely to be real (otherwise the vendor would not have reported it), but the issue could already be identified in a different CVE item. Consultation with the vendor may clear up any ambiguity, but it is not always possible or feasible.

Deletions

There may be several reasons why a candidate or entry should be "Deleted" from its associated list, including:

- If it is a duplicate of another CVE item
- If further analysis shows that the vulnerability does not exist (e.g., the original report was incorrect)
- If the item needs to be recast

Because any number of CVE-compatible products and services could be using older CVE identifiers, it is important to keep a record of what happens to each item that must be "deleted." A *candidate* is deleted by rejecting it. An *entry* is deleted by deprecating it. The process is the same for candidates and entries, and includes the following:

1. An announcement is made to the Editorial Board that the item will be rejected or deleted.
2. At least four business days are allowed for Board members to raise any questions (for candidates, this takes place in the Interim Decision phase)
3. A Final Decision is made to Reject or Deprecate the item.
4. All references for the item are deleted.
5. The description is removed and replaced with a statement that says that the item has been Rejected (for candidates) or Deprecated (for entries).
6. A short reason for the action is included in the description.
7. If the item is a duplicate, a reference is made to the correct CVE item(s).
8. The change is noted in the next CVE difference report.
9. The item remains in its associated list so that there is always a record of what happened to it.

The references and descriptions are removed so that it is clear to everyone that the item is no longer identifying the original vulnerability, and that the item is not returned as a result of keyword searches.

Candidate Reservation and Candidate Numbering Authorities

Candidate reservation allows responsible researchers, vendors, and incident response teams to include candidate numbers in the initial public announcement of a vulnerability. It ensures that a candidate number is instantly available to all CVE users and makes it easier to track vulnerabilities over time.

The basic process is:

1. There is a request for one or more candidate number(s).
2. MITRE reserves the candidate number(s) and provides the number(s) to the requester, and creates "blank," content-free candidate(s) on the CVE Web site.
3. The requester shares the candidate number(s) with all parties involved in the disclosure.
4. The requester includes the candidate number(s) in the vulnerability advisory.
5. The requester makes the candidate(s) public and notifies MITRE.

6. MITRE updates the candidate(s) on the CVE Web site to provide the details.
7. MITRE proposes the candidate(s) to the Editorial Board.
8. If a candidate is accepted as an official CVE entry, then the requester updates the number in the advisory.

If a candidate was reserved and the issue was never made public, the candidate will be deleted. This is referred to as "releasing" the candidate. Because the candidate was never public — and in some cases, the candidate was never assigned to a specific vulnerability — it is deleted entirely.

Candidate Numbering Authorities

Candidate Numbering Authorities (CNAs) are organizations that distribute CVE candidate numbers to researchers and information technology vendors for inclusion in first-time public announcements of new vulnerabilities, without directly involving MITRE in the details of the specific vulnerabilities. On an as-needed basis, MITRE provides a CNA with a pool of candidate numbers for the CNA to assign.

CNAs can help the CVE Initiative in several ways. When they function as intermediaries between a vulnerability researcher and the affected vendor, they can provide a candidate number without notifying MITRE of the vulnerability, which reduces the risk of accidental disclosure of vulnerability information. They increase the scope and visibility of CVE candidates by providing additional access points for researchers and vendors to obtain candidate numbers. They can utilize existing working relationships with researchers and vendors, which the affected parties may not have formed with MITRE. If they are already an integral part of the normal process by which vulnerabilities are disclosed, their participation prevents the addition of another party (i.e., MITRE) from interfering with that process or causing further delays. Finally, their participation relieves MITRE of some potentially labor-intensive tasks, allowing it to dedicate resources to other aspects of CVE that need attention.

Requirements to be a CNA

A CNA must be a major software vendor with a significant user base and an established security advisory capability, or an established third party that typically acts as a neutral interface between researchers and vendors. MITRE also functions as a CNA in a limited capacity.

The CNA must be an established distribution point for first-time vulnerability announcements. It must have a member of the Editorial Board who performs technical tasks. In keeping with the CVE requirement to identify public issues, the CNA must only assign candidates to security issues that will be made public. Finally, it must follow responsible disclosure practices that are accepted by a significant portion of the security community. Responsible disclosure is important for CVE because, otherwise, it is more likely that duplicate or inaccurate information will be introduced into CVE.

CNA Tasks

CNAs must consistently apply documented CVE content decisions (with exceptions made for technical subtleties or incomplete documentation). They must also coordinate the exchange of candidate numbers across all involved parties (vendor, researcher, response team, etc.) in order to reduce the risk of producing duplicate candidate numbers. CNAs must notify MITRE when candidates have been publicly announced. Because disclosure practices directly impact the accuracy of CVE, CNAs must recommend best practices in vulnerability disclosure to both researcher and vendor. A CNA must verify that the reported vulnerability has not already been assigned a CVE or candidate number.

MITRE is working to increase the number of CNAs. Some of the greatest challenges include educating CNAs about content decisions and determining the process for exchanging candidate numbers across multiple parties, especially if more than one party reserves candidates from MITRE.

Communications from CNAs to MITRE

The following series of communications occur between CNAs and MITRE:

1. The CNA requests a pool of candidate numbers.

2. The CNA announces the publication of a new candidate, which allows MITRE to update the candidate information on the CVE Web site.
3. The CNA may need to consult with MITRE regarding CVE content decisions.
4. The CNA notifies MITRE of suspected abuses of the CVE process by researchers.
5. The CNA notifies MITRE and other parties when duplicate candidates are detected.

The primary method of communication is through e-mail, although telephone discussions are sometimes necessary when a problem is particularly complex with respect to CVE content decisions or the nature of the vulnerability.

A third-party CNA must also maintain awareness of all vendors and CNAs who utilize candidate numbers. Because a third party might gain a competitive advantage by initially providing candidate numbers to a limited audience (outside of the researcher and vendor), the CNA should not publish CVE candidate numbers in any manner that might provide it with an economic, technical, or political advantage over its competitors.

Vendor CNAs must clearly advertise their security point of contact. They must provide the candidate to other affected parties (e.g., other vendors, researchers, or response teams). They must include candidate numbers in their own advisories. They can only use their pool of candidates for vulnerabilities in their own products. They must apply CVE content decisions to determine the proper number of candidates to assign, even if the content decisions are contrary to the vendor's own criteria. If the issue does not meet the vendor's minimum risk level for releasing an advisory, the CNAs should still provide candidates for that issue. Finally, when an issue has already been published and assigned a candidate, the vendor must use the existing candidate number.

Vendor Liaisons

As can be seen by the requirements for a CNA, it can be resource intensive and technically difficult to act as a CNA. Many vendors may want to participate properly in the CVE Initiative but not have the capability or desire to act as a CNA. A vendor liaison can work with another CNA to obtain or verify CVE candidates in the liaison's own products, and include candidate numbers in its advisories.

Researcher Responsibilities

The researcher must reserve candidates for a particular vulnerability from only one CNA. If the affected software vendor is a CNA, then the researcher must obtain the candidate from the vendor. The researcher needs to provide the CNA with enough details for the CNA to apply CVE content decisions. The researcher must coordinate the exchange of the candidate number across all involved parties. Finally, the researcher must include the candidate number in an advisory and publish the information through known reliable channels (vendor or response team), or known public channels with peer review (such as Bugtraq or NTBugtraq).

Researchers could adversely affect the reservation process in several different ways that could impact the overall quality of CVE. For example, the researcher's disclosure process may frequently result in duplicate candidates (e.g., by refusing to work with a vendor). The researcher may frequently publish issues that are discovered to be false or so error-prone as to cause his associated candidates to be rejected by the Editorial Board. It is the responsibility of MITRE and the CNAs to identify and resolve such abuses.

Content Decisions

CVE content decisions are the guidelines used to ensure that CVE items are created in a consistent fashion, independent of who is doing the creation. There are two major types of content decisions: inclusion and abstraction. Inclusion content decisions specify whether a vulnerability or exposure should go into CVE. Abstraction content decisions specify the level of abstraction (level of detail) at which a vulnerability should be described (e.g., whether a particular security issue should be given one candidate or five candidates).

There are differences between many vulnerability databases or products in the type of content they include, as well as the level of abstraction. These differences occur within the same database or product. Because of this variety and the flat structure of the CVE name, CVE cannot be sufficiently flexible to account for these differences. It is important for vulnerability analysts to be aware of these differences. As such, CVE content decisions not only document the guidelines for creating content, they often indicate areas in which there is inconsistency across vulnerability information sources. Quantitative analyses of vulnerabilities that use CVE-

EXHIBIT 70.3 The SF-LOC and SF-EXEC Content Decisions

CD:SF-LOC: multiple security flaws in the same executable, but possibly in different lines of code	CD:SF-EXEC: multiple executables exhibiting the same problem
CD:SF-LOC only applies when there may be multiple bugs that appear in the same executable (modulo the codebase, i.e., all "ps" executables in UNIX are treated the same).	CD:SF-EXEC only applies when there are multiple executables in the same package that demonstrate the same problem.
SPLIT (create separate CANs) between problems of different types (e.g., buffer overflow versus symlink problem).	"The same package" basically means "bundled executables that perform related functions that are not distributed separately." Microsoft Word and PowerPoint are not the same package (they can be installed separately). The set of executable programs that support the lpd capability in UNIX are the same package.
SPLIT between problems of the same type if problem X appears in some version that problem Y does not.	SPLIT when the problems are of different types.
MERGE problems of the same type within the same version. Explicitly list the different problems in the description.	SPLIT when the problems are in different versions (for some definition of "version" that effectively describes the package).
	MERGE when the problems are of the same type. Explicitly identify the separate affected "components" or executables in the package.

normalized data can be more easily replicated, and the CVE content decisions help to ensure that the data is normalized in a predictable fashion.

Two of the most commonly used content decisions (CDs) are shown in Exhibit 70.3. They also highlight some of the most common discrepancies across vulnerability information sources. These CDs were revised many times over a period of a year and a half, but they were stabilized in early 2001 when they were modified to make them less sensitive to the amount of information that is available for a vulnerability. From an academic perspective, this approach is not optimal but it is proving to be repeatable and less likely to cause candidates to become split or merged when new information becomes available after the initial analysis has been performed.

CD:SF-LOC is less sensitive to the lack of detailed information such as source code, exploits, or attack traces. However, it is still sensitive to changes in version information. Problems that occur in libraries pose special challenges for this content decision because they could be exhibited at several points within the same executable, or in many different executables. Ultimately, while this CD is intended to minimize the amount of information required to produce results, it is still dependent on critical information sources such as the vendor of the vulnerable product.

CD:SF-EXEC is also susceptible to error if the problem occurs in a library or other common codebase.

There are approximately 15 other content decisions currently defined for CVE, some of which are identified in the "Scope of the CVE List" section.

CVE Editorial Board

The CVE Editorial Board includes prominent information security specialists from numerous information-security-related organizations around the world, including commercial security-tool vendors, academic and research institutions, and government agencies. MITRE invites other information security experts to participate on an as-needed basis, based on recommendations from other Board members or MITRE's own identification of gaps within the current representation. Archives of Board meetings and discussions are publicly available on the CVE Web site.

Members of the Editorial Board have different roles and tasks in support of the CVE Initiative. There are four roles: Technical members, liaisons, advocates, and emeritus members. Each Board member has one primary role but can take on other roles. Technical members participate in the creation, design, review, maintenance, and applications of the CVE List. Liaisons represent a significant constituency, related to or affected by CVE, in an area that does not necessarily have technical representation on the Board. In some cases, a liaison may represent an individual organization, which may include software vendors. Advocates actively support or promote CVE in a highly visible fashion. This role is reserved for respected leaders in the security community who help bring credibility to the CVE Initiative and give CVE a wider reach outside the security community. Emeritus members were formerly active and influential in the CVE Initiative and are recognized for their significant contributions.

Board members must meet the minimum levels of effort for the tasks they undertake, which varies across tasks. If a Board member participates in multiple tasks, then the minimum expectations for each individual task may be lowered accordingly.

All members perform *consultation* and *awareness* tasks. Consultation includes participating in Board meetings, or discussion of ad hoc issues related to CVE content or Editorial Board processes such as content decisions, Board membership, or CVE compatibility. Awareness includes participating in Board meetings or reading meeting summaries, and regularly reading posts on the Editorial Board mailing lists.

Many members also perform outreach by actively promoting CVE and educating the public about it, or introducing various contacts to the CVE Initiative. Occasionally, some Board members participate in activities that are undertaken under the Board context, but not directly related to CVE.

Technical members regularly perform one or more of the following tasks:

- *Voting.* The primary task for most technical members is to review, comment on, and vote on CVE candidates proposed to the Editorial Board. Some members vote regularly. Others vote on an ad hoc basis (e.g., when there is an effort to reach a specific content goal).
- *Content provider.* Some Board members provide their vulnerability databases to MITRE for conversion into candidates, which ensures that CVE content is as complete as possible. Others are actively involved in candidate reservation. Others may be CNAs, which are authorized to assign CVE candidate numbers to security issues before they are publicized.
- *CIEL.* Members participate in the review and development of the Common Intrusion Event List (CIEL), a "CVE-for-IDS" that is currently being drafted by MITRE and is discussed later.

Liaisons perform one or more of the following tasks:

- *Community education.* The liaison must educate the liaison's own community about CVE, where appropriate.
- *Board education.* The liaison must educate the Editorial Board about the needs and interests for CVE of the liaison's community, where appropriate.
- *Voting.* If the member is a software vendor liaison, the member must vote on candidates related to that vendor's products.

Liaisons may undertake other technical tasks.

A liaison that represents a constituency beyond an individual organization must be visible and active in the liaison's constituency community. A liaison who represents an individual organization must be able to effectively communicate with the relevant parts of that organization. Software vendor liaisons must be able to effectively communicate with the vendor's security and product development teams.

Advocates' tasks include endorsing CVE to constituencies that will benefit from it, fostering better communication between constituencies, participating in Editorial Board activities (especially in decisions related to Board structure and strategic activities), and consulting when needed.

Guiding the Direction of CVE: The CVE Senior Advisory Council

The CVE Senior Advisory Council was established to ensure that the CVE Initiative receives the sponsorship — including funding and guidance — required to maximize the effectiveness of CVE in supporting government efforts to improve the nation's ability to identify and respond to vulnerabilities and information assurance attacks or issues. The CVE Senior Advisory Council is composed of senior executives in U.S. Government agencies, many of which provide (or provided) funding for CVE.

The Council provides business planning oversight and prioritization of new CVE and related services, discusses CVE and related security policy implications for the federal government, and identifies materials and resources that might be useful for government CIOs and senior managers.

The Council promotes the adoption of CVE at the strategic level; works to assure funding for core CVE activities over the long term, including outreach to government organizations and agencies; and acts as a catalyst for CVE and related activities. The Council also brings to CVE its insights on community needs and possible areas for new CVE-related services.

Council membership is extended to the senior executives of those government organizations that provide funding for core CVE activities, as well as other executives who have the background and ability to help the Council achieve the stated objectives.

EXHIBIT 70.4 Cross-linking through the CVE List.

One of the Council's main roles is to provide strategic guidance for the direction of CVE. For example, the Council has encouraged MITRE to involve the various Information Sharing and Analysis Centers (ISACs) more closely in CVE, conduct outreach to large organizations outside the security industry, define qualitative goals, and concentrate more on addressing the needs of the IDS segment of the security-tools industry with respect to CVE.

CVE Compatibility

The basic premise of the CVE List is that there be one name for a vulnerability or exposure. A CVE-compatible product or service must understand the CVE names for vulnerabilities and allow the user to interact with the product or service in terms of those CVE names. This does not mean that the product or service only uses CVE names for vulnerabilities, but rather that in addition to its own native label for a vulnerability, it is aware of the CVE name for that vulnerability. This support for CVE names is central to the concept of CVE compatibility. The CVE-compatible tool, Web site, database, or service must use CVE names in a way that allows users to correlate its information with other repositories, tools, and services that also use CVE names, as shown in Exhibit 70.4.

Uses of CVE Compatibility

Integrating vulnerability services, databases, Web sites, and tools that incorporate CVE names will provide an organization with more complete and efficient coverage of security issues. For example, a report from a vulnerability scanning tool that uses CVE names will enable the organization to quickly and accurately locate fix information in one or more of the CVE-compatible databases and Web sites.

It is also possible to determine exactly which vulnerabilities and exposures are covered by each CVE-compatible tool or service because the CVE List provides a baseline for comparison. After determining which of the CVE entries apply to its platforms, operating systems, and commercial software packages, an organization can compare this subset of the CVE List to any particular tool's or service's coverage.

Network and security trade journals are already referring to CVE name support as a desirable feature in product reviews and comparisons of scanners and IDS devices.[4,5] Similarly, the National Institute for Science and Technology (NIST) has published a recommendation to all federal government agencies and services for the use of CVE-compatible products and services whenever possible;[6] and in February 2003, the Department of Defense issued Directive 8500.2, Information Assurance (IA) Implementation Instruction, which states that,

"For improved interoperability, preference is given to tools that express vulnerabilities in the Common Vulnerabilities and Exposures (CVE) naming convention."

Just as other types of information security products tend to focus on a particular core function or capability, platform, or types of issues, the various products, services, and repositories that strive to meet the CVE compatibility requirements will focus on different portions of the CVE List. For example, some deal with UNIX while others focus on Windows NT; and some focus on network-based or host-based vulnerabilities. Users must evaluate CVE-compatible items against their organization's specific needs in terms of platform and software product coverage.

The CVE Compatibility Requirements

At its core, CVE compatibility involves four basic requirements:

1. Customers are able to use CVE names to inquire about scope, content, or coverage, and then receive any related information.
2. Customers are able to obtain output that includes all related CVE names.
3. The owner of the item makes a good-faith effort to ensure that the item's mapping from its own elements to CVE names remains as accurate as the CVE List and that the compatible items are updated over time.
4. Standard documentation includes a description of CVE compatibility and the details of how customers can use the CVE-related functionality of their tool, database, Web site, or service.

In general, vendors are given flexibility to implement the requirements in a variety of ways. Users can then determine which features or implementations are best suited to their needs.

The CVE Compatibility Evaluation Process

MITRE's current approach for establishing the compatibility of a product or service involves two phases. The first phase requires the completion of a short informational "CVE Compatibility Declaration Form," which is used to register an organization's declaration of intent with respect to CVE compatibility. The organization is asked to review the compatibility requirements and then make a statement regarding whether the organization believes that its product or service currently fulfills the requirements, or if the organization is working toward fulfilling the requirements. This phase of the CVE compatibility process does not result in an official evaluation by MITRE; rather, MITRE only reviews the declaration. As long as the products or services are commercially available, the declaration and an endorsement quote from the vendor (if desired) are posted on the CVE Web site. This phase of the compatibility process has been in effect since October 1999, when the CVE Initiative started, and can be performed very quickly. It makes the vendor aware of high-level expectations for CVE compatibility and establishes the proper communication channels between MITRE and the organization.

When the organization believes that its product or service has obtained full compliance with the CVE compatibility requirements, it can then request a formal review and evaluation, which begins the second phase. In development for the past year, this formal process has a "branding program" and logo to indicate successful completion of the compatibility evaluation. A major component of this phase requires specific details about how an organization has satisfied each of the mandatory CVE compatibility requirements. The organization must complete an extended "CVE Compatibility Requirements Evaluation Form," which requires the signature of an authorized representative of the submitting organization. Additionally, the organization provides MITRE with the CVE-related user documentation for the product or service.

The organization's statements and documents are evaluated, and MITRE arranges to verify the accuracy of the mapping between CVE names and the organization's underlying data repository. Upon completion of this review, the organization's detailed evaluation form and supporting statements will be posted on the CVE Web site for public review and use, along with MITRE's concurrence with the organization's statement. MITRE then provides the organization with the special CVE-compatible logo and formally gives the organization permission to use the CVE-compatible logo and the term "CVE-compatible."

Although the second phase takes more effort than the first phase for both the submitting organization and MITRE, it has been designed to minimize the expense to both. This approach avoids an evaluation process that would make it too expensive for freeware or smaller software vendors to obtain compatibility. Using the questionnaire and statement of compatibility, the level of effort is kept reasonable, while making a good effort to verify that the submitting organization properly understands and correctly implements the CVE compatibility requirements. The publication of the organization's statement on the CVE Web site allows end users to

compare how different products satisfy the requirements and then the market can then decide which specific implementations are best.

MITRE started internal testing for the second phase of the CVE compatibility assessment process in February 2002. A "beta test" was then conducted with a small number of organizations in the April to September timeframe, followed by a public roll-out on May 7, 2003.

Growth of CVE-Compatible Products and Services

The list of organizations declaring CVE-compatible products and services is continuously expanding and is international in scope. As of mid-May 2003, 84 organizations had made declarations of compatibility. For a current list, visit the CVE Web site (http://cve.mitre.org/compatible/).

The number of products and services that are working toward CVE compatibility has grown significantly over time. In October 1999, 15 products intended to be CVE-compatible; six months later, the number had doubled to 30, and exceeded 50 by July 2001. After an increase in activity in recent months, there are 126 products or services on the way to CVE compatibility as of mid-May 2003; 56 other organizations are working on declarations for 121 additional products or services.

Challenges and Opportunities

As CVE moves forward, it faces a variety of challenges and opportunities. The challenges include renumbering the CVE List, identifying the proper scope for the CVE List, and addressing the impact of vulnerability disclosure practices on CVE accuracy (including vendor acknowledgment and replication of the vulnerability). At the same time, the opportunities include analyzing vulnerability causes, improving vulnerability testing methods and veracity, facilitating large-scale quantitative comparisons of security tools and databases, filling in some gaps in research (such as analysis of configuration problems and developing a low-level taxonomy of vulnerabilities), and delivering real improvements in the way organizations manage the risks from vulnerabilities and exposures.

Challenges in the Current Naming Scheme

The current naming scheme allows humans to easily distinguish between CVE candidates and entries (CAN-YYYY-NNNN versus CVE-YYYY-NNNN). This distinction was chosen early in the CVE Initiative, partially based on how names are handled in other fields. However, CVE names are normally not considered atomic in data processing operations, and as such they may not be found easily by most search mechanisms. Also, the differing candidate and CVE numbering schemes cause maintenance and search problems.

Search engines may separate the name into three different terms (CAN, YYYY, NNNN) because the hyphen is sometimes considered a word separator, which can make it difficult to easily find information on the Internet using CVE names. In other cases, a search engine may need to be modified to quote the hyphen parts of the CVE name. Finally, the encoding of the year in the name may cause some problems with misuse, as it does not necessarily reflect the year in which the vulnerability was first publicized. In addition, the sequence number within the name may represent a small information leak if a candidate number is reserved for an issue long before the issue is made publicly known.

The differences between the candidate name and the CVE entry name can be difficult to manage. For example, when a candidate becomes an official entry, all CVE-compatible vendors need to update their databases to convert the candidate number to a CVE number, which can be labor intensive. In addition, users might still search for the candidate number instead of the CVE number; and some CVE-compatible products or services may not find the associated CVE entry if the user uses the candidate number in the search. To avoid this problem, each CVE-compatible product or service would need to implement a specialized function. Some omit the CAN- and CVE- prefixes outright, but this prevents a user from knowing whether the item is a candidate or an entry. The CVE Web site handles these discrepancies flexibly, but it requires specialized code. Many CVE-compatible tools are not as flexible, and such a capability is not required because CVE compatibility does not require the use of candidates.

A solution would be to construct the CVE names in a way that minimizes these types of implementation problems. Using just a number would not be suitable because numbers are so commonly used in so many databases and search engines that it could be difficult to properly distinguish a CVE number from other

numbers. A scheme in which a symbol (CVE) is prepended to a number (e.g., CVE12345) could work better. If such a scheme is adopted, then the status of a CVE item — whether candidate or entry — could be noted as a separate field in CVE.

While a change to the naming scheme may provide substantial benefits, the utility of CVE would be lost if the names change too often. CVE-compatible vendors will incur high maintenance costs if and when CVE moves to a new naming scheme. Educating the public will be an additional challenge. Therefore, MITRE and the CVE Editorial Board must give strong and thoughtful consideration to any new scheme. The naming scheme should only be changed once, and there should be a period of time in which the original scheme is still supported.

Scope of the CVE List

The scope of the CVE List has been discussed and debated many times during the evolution of CVE. The discussion has generally focused on two questions:

1. What types of security issues are included on the list?
2. What type of information is included with each issue, and what is the format of CVE information?

Not only do people define "vulnerability" differently, which will impact what would or would not be included on the CVE List in and of itself, but they also have different ideas regarding which types of issues should be included on the CVE List. Some of these issues are formalized in content decisions (prefaced by "CD:").

- *Vulnerabilities and exposures in beta software (CD:EX-BETA).* These types of vulnerabilities are reported fairly often, but it is sometimes argued that beta software is expected to be buggy. In general, such vulnerabilities are excluded from CVE, with the following exceptions: (1) if the software is in wide distribution, or (2) if the software is consistently released in beta versions instead of final versions (e.g., the ICQ program).
- Vulnerabilities and exposures in online services such as free Web-based mail services, online banking, and E-commerce, etc. (CD:EX-ONLINE-SVC). Such problems are normally addressed with a single fix on the server, by the service provider, and do not require any action by its customers.
- Problems in a network client that cause a denial-of-service whose scope is limited to the client, which can be addressed by restarting the client, and which can only be exploited by a passive attack (CD:EX-CLIENT-DOS). For example, if a Web browser cannot handle a certain sequence of characters, but the problem can only be triggered by enticing a user to visit a particular Web site and it only causes the client to crash, then that issue would not be added to CVE.
- Malicious code such as viruses, worms, and Trojan horses (this category excludes backdoors that were deliberately inserted by the developer). Technically, the presence of such malicious software satisfies CVE's definition of a vulnerability. However, attempting to identify and catalog all malicious code would expand the size of CVE significantly, making it unusable for too many people. In addition, it is believed that defining standard names for malware is best left to the anti-virus community.
- Vague reports of vulnerabilities, even in vendor advisories (CD:VAGUE).
- Issues related to security policy violations. Policies such as minimum password length and password aging, approved services, and conformance to specific software versions vary across each enterprise, so it is difficult to create CVE items that try to capture such policies. Insecure configurations often fall into this category.
- *Issues not necessarily proven to be "exploitable."* For example, many Linux vendors release an advisory for an issue that may have security implications, even if an exploit is not known to exist. This often happens with buffer overflows, format string vulnerabilities, and signedness errors.
- Issues related to intrusion detection "events" that are not easily described in terms of vulnerabilities or exposures (e.g., port scanning).

One difficulty with regard to these decisions is that some vulnerability scanners, intrusion detection systems, databases, and services may identify some of the security issues that fall into one or more of the above categories of items. Some end users may also wish to see some of these types of problems addressed by CVE. For example,

one of the most frequently asked questions is whether CVE is devising a standard name for viruses. (Many end users have had difficulty dealing with viruses that have multiple names from different vendors.)

In most of these "exception cases," it has not yet been decided whether these types of security problems will be included or excluded from the CVE List. These content decisions (which are further described later in this chapter) are periodically discussed by the Editorial Board. MITRE typically creates candidates for beta software, client-side DoS, and vague vulnerability reports. However, these candidates are "labeled" with the associated draft content decisions, and they will not be accepted as official entries until sufficient discussion has taken place by the Editorial Board and the content decisions have been sufficiently evaluated for completeness and repeatability. For intrusion detection events, MITRE is creating the Common Intrusion Event List (CIEL), which is described elsewhere in this chapter.

The second area of debate about the scope of the CVE List focuses on the type of information that should be included with each issue, and also the format of CVE information. CVE entries currently have three fields: name, description, and references. Candidates are included with additional information such as votes from Editorial Board members and the phase, which identifies how far the candidate has progressed through the review process. End users of CVE sometimes ask for additional fields beyond what is currently provided, including risk level, operating system, product vendor, fix information, and greater levels of detail in the descriptions. Such information is not required for the purpose of naming vulnerabilities, but the request for this additional information does indicate that some consumers wish to use CVE as a vulnerability database, or they want an easier way to identify the set of CVE names they care about. There are two main concerns with respect to making this information available: (1) it increases the workload on MITRE and the Editorial Board, and (2) it would expand CVE's scope more directly in competition with commercial security vulnerability database vendors.

While MITRE has decided not to adopt these previous types of suggested additions to the information in the CVE List, in other cases, MITRE is considering making available additional information that is specifically related to how CVE content is managed. For example, candidates include an "analysis" field that describes how content decisions were applied (e.g., why a particular level of abstraction was chosen), how vendor acknowledgment was determined, and other information that may indicate why a candidate was created in the way it was. Other information that is included is a reference to the particular content decisions that affect the candidate, the date that the vulnerability was publicly announced, what specific modifications were made to the candidate, whether the vendor has acknowledged the problem, and the dates of each phase that the candidate has reached (e.g., proposal, modification, and interim decision).

Other users of CVE would like to obtain more precise change logs for each candidate or entry. Some of this information is made available to Editorial Board members for voting purposes. Because voting ballots appear in the Editorial Board mailing list archives, some of the information is publicly viewable, but it cannot be extracted easily. However, this information would be useful to a certain portion of CVE users, such as those who may want to know why a candidate that has sufficient ACCEPT votes has not been promoted to an official entry. There are plans to make some of these fields more easily accessible in the future.

Labeling each candidate or entry with a "confidence level" that represents a level of certainty that the report was correct was also considered. Some candidates identify vulnerabilities in uncommon software, which are reported by researchers who are unknown to the voting Board members. Subsequently, there may not always be a strong level of confidence that the issue is real or accurately described. Confidence is now "encoded" within the recommended voting guidelines for when Board members can ACCEPT a candidate, but it was decided that an overt and separate field would not be created.

Another request that is received fairly often is to provide the CVE List as an XML document. Work in that direction has started but is not complete as of this writing.

CVE has also been approached about translating the CVE List and CVE Web site into other languages. While interested in supporting the use of CVE by groups that do not have English as their native language, CVE's resources will not allow such efforts by CVE. However, by the time this chapter is published, a Chinese translation of the CVE Web site, including a translated version of the CVE List and candidates, will be available on a site hosted by another organization. A licensing mechanism and coordination process was devised so that other organizations interested in hosting similar sites in other languages can be accommodated. CVE's main focus in these translation arrangements is to ensure the quality and integrity of the CVE Initiative while broadening its international reach.

Addressing the Needs of IDS Tools with CIEL

Many events that are detected by IDSs do not have a clear association with vulnerabilities or exposures, including port mapping, protocol decodes, failed binary integrity checks, and generic attack signatures. For cases in which an event overlaps CVE (e.g., an attempt to exploit a specific vulnerability), the CVE descriptions focus on the nature of the security problem as opposed to how it might be exploited. A number of Editorial Board members and others involved in IDS work have expressed the desire to have CVE encompass all IDS events.

MITRE is currently building a draft list for IDS events, referred to as CIEL (Common Intrusion Event List, pronounced "seal"), that is sometimes informally described as "a CVE for intrusion detection." It is intended to provide a naming scheme for all network- or host-based events that may be useful in detecting intruder activities, but are not directly associated with CVE items. MITRE is monitoring the efforts of the IETF Intrusion Detection Working Group (IDWG) to identify areas of overlap with CIEL. The IDWG is addressing the larger needs for information exchange across IDSes, but CIEL could be used to satisfy the IDWG's requirement for a common attack name.

Several assumptions will be guiding the development of CIEL: there is a wider variety of IDS events than vulnerabilities, there is more variety across IDSs in the level of abstraction (or level of detail) than there is in vulnerability scanners and databases, and there is not much well-defined and commonly accepted terminology in the IDS arena.

In early 2002, MITRE created a CIEL working group under the CVE Editorial Board. Discussions are held on a separate mailing list. As of this writing, MITRE is still expanding the Editorial Board to include other members of the CIEL working group.

Managing Risk with CVE Compatibility

The increase in CVE-compatible products and services can change the way organizations use security tools and data sources to address their operational security posture. For example, an organization can use CVE-compatible products and services to improve its response to security advisories. CVE-compatible advisories include CVE entries of vulnerabilities that scanners can check for, and an IDS can be examined for appropriate attack signatures for the vulnerability described in the advisory. The incorporation of CVE names and CVE-compatible products and services provides a more structured and predictable process for handling advisories than most organizations currently possess.

Along similar lines, when a group of concerned security professionals last year composed a list of the ten most-common critical Internet security threats, they included CVE names for them.[7] Orchestrated by the SANS Institute, the effort represented the consensus of a wide variety of security experts. To help ensure specificity and make the recommendations actionable, each suggestion included the appropriate CVE names, totaling 68, and detailed the specific issue areas for a variety of platforms and products. The next update to the SANS list,[8] which is now co-sponsored by the FBI, grew to a list of the 20 most-common critical Internet security threats and included 125 CVE names. The most recent top-20 list now includes 242 CVE names.

Additionally, as shown in Exhibit 70.5, compatible products and services can be used by an organization to check over an ongoing attack with its CVE-compatible IDS system (A). In a CVE-compatible IDS, specific vulnerabilities that are susceptible to the detected attack are provided as part of the attack report. This information can be compared against the latest vulnerability scan by a CVE-compatible scanner (B) to determine whether the enterprise has one of the vulnerabilities or exposures that can be exploited by the attack. If it does, a CVE-compatible fix database at the vendor of the software product or a vulnerability Web site (C) can identify the location of the fix for a CVE entry (D), if one exists.

In addition, for systems that an organization builds or maintains for customers, CVE-compatible advisories and announcements can help directly identify any need for software fixes from the commercial vendors of those systems. For security issues in software distributed by multiple vendors, CVE names can help users determine when different advisories are referring to the same vulnerability.[9]

Summary of Progress

Here is one way of looking at progress against the CVE strategy:

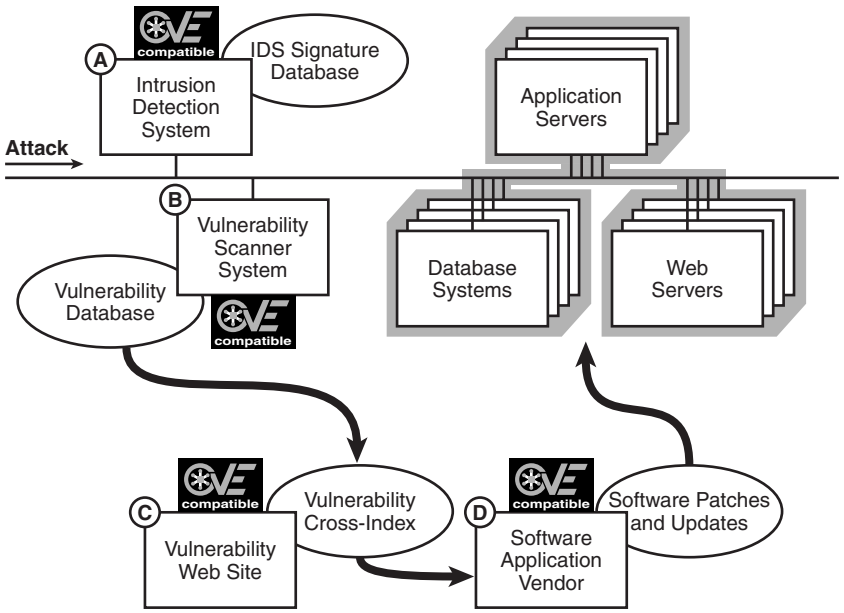


EXHIBIT 70.5 A CVE-enabled process.

- CVE is gradually approaching the goal of uniquely naming every publicly known security-relevant software mistake. More than half of all known software mistakes are now either included on the CVE List or are under review.
- CVE names are now regularly included in advisories by a fairly large group of organizations, including ISS, IBM, Rain Forest Puppy, @stake, Microsoft, BindView, NAI, CERT/CC, SGI, eEye, COMPAQ, Ernst & Young, CISCO, Rapid 7, NSFOCUS, Sanctum, Alcatel, EnGarde Secure Linux, Caldera, Red Hat, SecurityFocus, VIGILANTe.com, Cert-IST, Mandrake Linux, Debian, Foundstone, Apple, iDEFENSE, HP, Symantec, DHS/NIPC, KDE e. V., Beyond Security Ltd., Digital Defense Inc., Core-ST, The OpenPKG Project, Corsaire, The FreeBSD Project, and Gentoo Linux.
- CVE usage in information security products and services now stands at more than 240 that are either available or in development, with more being announced regularly.
- CVE usage has been included in a recent recommendation from the U.S. Department of Defense, which follows the earlier recommendation from the National Institute of Science and Technology (NIST).
- Various trade journals have started using support for CVE names as a review item in articles.
- For three years in a row, the SANS Top Ten/Top Twenty guidance (now co-issued by the Federal Bureau of Investigation [FBI]) has included CVE names.
- The CVE E-newsletters are subscribed to by more than 4000 different organizations from more than 90 countries worldwide, and the CVE Web site is being visited from individuals in more than 125 countries on a regular basis.
- Of the dozen companies this has been discussed with, several are considering adding CVE name support to their fix-it sites and update mechanisms.

Acknowledgment

The summary work contained in this chapter was funded by the MITRE Corporation. It is based on the composite effort of all those working on the Common Vulnerabilities and Exposures Initiative, including but not limited to the CVE Editorial Board, the CVE Advisory Council, and CVE-compatible vendors.

Notes

1. MITRE is a not-for-profit company that works in the public interest to provide systems engineering, research and development, and information technology support to the U. S. Government.
2. D.E. Mann and S.M. Christey, "Towards a Common Enumeration of Vulnerabilities," *2nd Workshop Research with Security Vulnerability Databases*, Purdue University, West Lafayette, IN, 1999; http://cve.mitre.org/docs/cerias.html (current as of May 2003).
3. R.A. Martin, "Managing Vulnerabilities in Networked Systems," *IEEE Computer Society's Computer Magazine,* 34(11), November 2001; http://www.computer.org/computer/co2001/ry032abs.htm (current as of May 2003).
4. J. Forristal and G. Shipley, "Vulnerability Assessment Scanners," *Network Computing,* 8 Jan. 2001; http://www.networkcomputing.com/1201/1201f1b2.html (current May 2003).
5. P. Mueller and G. Shipley, "To Catch a Thief," *Network Computing,* August 20, 2001; http://www.networkcomputing.com/1217/1217f1.html (current as of May 2003).
6. A. Saita, "CVE-Use Recommendations Open for Comment," *Security Wire Digest, 4(9), February 4, 2002;* http://www.INFOSECURITYMAG.COM/digest/2002/02-04-02.shtml#1b (current as of May 2003).
7. W. Jackson, "Top 10 System Security Threats Are Familiar Foes," *Government Computer News,* August 2000; http://www.gcn.com/state/vol6_no8/news/812-1.html (current as of May 2003).
8. S. Bonisteel, "'Top 10' List of Net Security Holes Grows to 20," *Newsbytes.com,* October 2, 2001.
9. Mark J. Cox, "'Chinese Whisper' Security Advisories," *LinuxWorld.com,* January 21, 2002; http://www.linuxworld.com/site-stories/2002/0121.whisper.html (current as of May 2003).

71

Roles and Responsibilities of the Information Systems Security Officer

Carl Burney, CISSP

Information is a major asset of an organization. As with any major asset, its loss can have a negative impact on the organization's competitive advantage in the marketplace, a loss of market share, and become a potential liability to shareholders or business partners. Protecting information is as critical as protecting other organizational assets, such as plant assets (i.e., equipment and physical structures) and intangible assets (i.e., copyrights or intellectual property). It is the information systems security officer (ISSO) who establishes a program of information security to help ensure the protection of the organization's information.

The information systems security officer is the main focal point for all matters involving information security. Accordingly, the ISSO will:

- Establish an information security program including:
  - Information security plans, policies, standards, guidelines, and training
- Advise management on all information security issues
- Provide advice and assistance on all matters involving information security

The Role of The Information Systems Security Officer

There can be many different security roles in an organization in addition to the information system security officer, such as:

- Network security specialist
- Database security specialist
- Internet security specialist
- E-business security specialist
- Public key infrastructure specialist
- Forensic specialist
- Risk manager

Each of these roles is in a unique, specialized area of the information security arena and has specific but limited responsibilities. However, it is the role of the ISSO to be responsible for the entire information security effort in the organization. As such, the ISSO has many broad responsibilities, crossing all organizational lines, to ensure the protection of the organization's information.

EXHIBIT 71.1 An Information Security Program Will Cover a Broad Spectrum

Policies, Standards, Guidelines, and Rules	Reports
Access controls	Risk management
Audits and reviews	Security software/hardware
Configuration management	Testing
Contingency planning	Training
Copyright	Systems acquisition
Incident response	Systems development
Personnel security	Certification/accreditation
Physical security	Exceptions

Responsibilities of The Information Systems Security Officer

As the individual with the primary responsibility for information security in the organization, the ISSO will interact with other members of the organization in all matters involving information security, to include:

- Develop, implement, and manage an information security program.
- Ensure that there are adequate resources to implement and maintain a cost-effective information security program.
- Work closely with different departments on information security issues, such as:
  - The physical security department on physical access, security incidents, security violations, etc.
  - The personnel department on background checks, terminations due to security violations, etc.
  - The audit department on audit reports involving information security and any resulting corrective actions
- Provide advice and assistance concerning the security of sensitive information and the processing of that information.
- Provide advice and assistance to the business groups to ensure that information security is addressed early in all projects and programs.
- Establish an information security coordinating committee to address organization-wide issues involving information security matters and concerns.
- Serve as a member of technical advisory committees.
- Consult with and advise senior management on all major information security-related incidents or violations.
- Provide senior management with an annual state of information security report.

Developing, implementing, and managing an information security program is the ISSO's primary responsibility. The Information Security Program will cross all organizational lines and encompass many different areas to ensure the protection of the organization's information. Exhibit 71.1 contains a noninclusive list of the different areas covered by an information security program.

Policies, Standards, Guidelines, and Rules

- Develop and issue security policies, standards, guidelines, and rules.
- Ensure that the security policies, standards, guidelines, and rules appropriately protect all information that is collected, processed, transmitted, stored, or disseminated.
- Review (and revise if necessary) the security policies, standards, guidelines, and rules on a periodic basis.
- Specify the consequences for violations of established policies, standards, guidelines, and rules.
- Ensure that all contracts with vendors, contractors, etc. include a clause that the vendor or contractor must adhere to the organization's security policies, standards, guidelines, and rules, and be liable for any loss due to violation of these policies, standards, guidelines, and rules.

Access Controls

- Ensure that access to all information systems is controlled.

- Ensure that the access controls for each information system are commensurate with the level of risk, determined by a risk assessment.
- Ensure that access controls cover access by workers at home, dial-in access, connection from the Internet, and public access.
- Ensure that additional access controls are added for information systems that permit public access.

Audits and Reviews

- Establish a program for conducting periodic reviews and evaluations of the security controls in each system, both periodically and when systems undergo significant modifications.
- Ensure audit logs are reviewed periodically and all audit records are archived for future reference.
- Work closely with the audit teams in required audits involving information systems.
- Ensure the extent of audits and reviews involving information systems is commensurate with the level of risk, determined by a risk assessment.

Configuration Management

- Ensure that configuration management controls monitor all changes to information systems software, firmware, hardware, and documentation.
- Monitor the configuration management records to ensure that implemented changes do not compromise or degrade security and do not violate existing security policies.

Contingency Planning

- Ensure that contingency plans are developed, maintained in an up-to-date status, and tested at least annually.
- Ensure that contingency plans provide for enough service to meet the minimal needs of users of the system and provide for adequate continuity of operations.
- Ensure that information is backed up and stored off-site.

Copyright

- Establish a policy against the illegal duplication of copyrighted software.
- Ensure inventories are maintained for each information system's authorized/legal software.
- Ensure that all systems are periodically audited for illegal software.

Incident Response

- Establish a central point of contact for all information security-related incidents or violations.
- Disseminate information concerning common vulnerabilities and threats.
- Establish and disseminate a point of contact for reporting information security-related incidents or violations.
- Respond to and investigate all information security-related incidents or violations, maintain records, and prepare reports.
- Report all major information security-related incidents or violations to senior management.
- Notify and work closely with the legal department when incidents are suspected of involving criminal or fraudulent activities.
- Ensure guidelines are provided for those incidents that are suspected of involving criminal or fraudulent activities, to include:
  - — Collection and identification of evidence
  - — Chain of custody of evidence
  - — Storage of evidence

Personnel Security

- Implement personnel security policies covering all individuals with access to information systems or having access to data from such systems. Clearly delineate responsibilities and expectations for all individuals.
- Ensure all information systems personnel and users have the proper security clearances, authorizations, and need-to-know, if required.
- Ensure each information system has an individual, knowledgeable about information security, assigned the responsibility for the security of that system.
- Ensure all critical processes employ separation of duties to ensure one person cannot subvert a critical process.
- Implement periodic job rotation for selected positions to ensure that present job holders have not subverted the system.
- Ensure users are given only those access rights necessary to perform their assigned duties (i.e., least privilege).

Physical Security

- Ensure adequate physical security is provided for all information systems and all components.
- Ensure all computer rooms and network/communications equipment rooms are kept physically secure, with access by authorized personnel only.

Reports

- Implement a reporting system, to include:
  - — Informing senior management of all major information security related incidents or violations
  - — An annual State of Information Security Report
  - — Other reports as required (i.e., for federal organizations: OMB CIRCULAR NO. A-130, Management of Federal Information Resources)

Risk Management

- Establish a risk management program to identify and quantify all risks, threats, and vulnerabilities to the organization's information systems and data.
- Ensure that risk assessments are conducted to establish the appropriate levels of protection for all information systems.
- Conduct periodic risk analyses to maintain proper protection of information.
- Ensure that all security safeguards are cost-effective and commensurate with the identifiable risk and the resulting damage if the information was lost, improperly accessed, or improperly modified.

Security Software/Hardware

- Ensure security software and hardware (i.e., anti-virus software, intrusion detection software, firewalls, etc.) are operated by trained personnel, properly maintained, and kept updated.

Testing

- Ensure that all security features, functions, and controls are periodically tested, and the test results are documented and maintained.
- Ensure new information systems (hardware and software) are tested to verify that the systems meet the documented security specifications and do not violate existing security policies.

Training

- Ensure that all personnel receive mandatory, periodic training in information security awareness and accepted information security practices.
- Ensure that all new employees receive an information security briefing as part of the new employee indoctrination process.
- Ensure that all information systems personnel are provided appropriate information security training for the systems with which they work.
- Ensure that all information security training is tailored to what users need to know about the specific information systems with which they work.
- Ensure that information security training stays current by periodically evaluating and updating the training.

Systems Acquisition

- Ensure that appropriate security requirements are included in specifications for the acquisition of information systems.
- Ensure that all security features, functions, and controls of a newly acquired information system are tested to verify that the system meets the documented security specifications and does not violate existing security policies, prior to system implementation.
- Ensure that all default passwords are changed when installing new systems.

Systems Development

- Ensure information security is part of the design phase.
- Ensure that a design review of all security features is conducted.
- Ensure that all information systems security specifications are defined and approved prior to programming.
- Ensure that all security features, functions, and controls are tested to verify that the system meets the documented security specifications and does not violate existing security policies, prior to system implementation.

Certification/Accreditation

- Ensure that all information systems are certified/accredited, as required.
- Act as the central point of contact for all information systems that are being certified/accredited.
- Ensure that all certification requirements have been met prior to accreditation.
- Ensure that all accreditation documentation is properly prepared before submission for final approval.

Exceptions

- If an information system is not in compliance with established security policies or procedures, and cannot or will not be corrected:
  - — Document:
- The violation of the policy or procedure
- The resulting vulnerability
- Any necessary corrective action that would correct the violation
- A risk assessment of the vulnerability.
  - — Have the manager of the information system that is not in compliance document and sign the reasons for noncompliance.
  - — Send these documents to the CIO for signature.

The Nontechnical Role of the Information Systems Security Officer

As mentioned, the ISSO is the main focal point for all matters involving information security in the organization, and the ISSO will:

- Establish an information security program.
- Advise management on all information security issues.
- Provide advice and assistance on all matters involving information security.

Although information security may be considered technical in nature, a successful ISSO is much more than a "techie." The ISSO must be a businessman, a communicator, a salesman, and a politician.

The ISSO (the businessman) needs to understand the organization's business, its mission, its goals, and its objectives. With this understanding, the ISSO can demonstrate to the rest of the management team how information security supports the business of the organization. The ISSO must be able to balance the needs of the business with the needs of information security.

At those times when there is a conflict between the needs of the business and the needs of information security, the ISSO (the businessman, the politician, and the communicator) will be able to translate the technical side of information security into terms that business managers will be better able to understand and appreciate, thus building consensus and support. Without this management support, the ISSO will not be able to implement an effective information security program.

Unfortunately, information security is sometimes viewed as unnecessary, as something that gets in the way of "real work," and as an obstacle most workers try to circumvent. Perhaps the biggest challenge is to implement information security into the working culture of an organization. Anybody can stand up in front of a group of employees and talk about information security, but the ISSO (the communicator and the salesman) must "reach" the employees and instill in them the value and importance of information security. Otherwise, the information security program will be ineffective.

Conclusion

It is readily understood that information is a major asset of an organization. Protection of this asset is the daily responsibility of all members of the organization, from top-level management to the most junior workers. However, it is the ISSO who carries out the long list of responsibilities, implementing good information security practices, providing the proper guidance and direction to the organization, and establishing a successful information security program that leads to the successful protection of the organization's information.

72

Information Protection: Organization, Roles, and Separation of Duties

Rebecca Herold, CISSP, CISA, FLMI

Successful information protection and security requires the participation, compliance, and support of all personnel within your organization, regardless of their positions, locations, or relationships with the company. This includes any person who has been granted access to your organization's extended enterprise information, and any employee, contractor, vendor, or business associate of the company who uses information systems resources as part of the job. A brief overview of the information protection and security responsibilities for various groups within your organization follows.

All Personnel within the Organization

All personnel have an obligation to use the information according to the specific protection requirements established by your organization's information owner or information security delegate. A few of the basic obligations include, but are not limited to, the following:

- Maintaining confidentiality of log-on passwords
- Ensuring the security of information entrusted to their care
- Using the organization's business assets and information resources for approved purposes only
- Adhering to all information security policies, procedures, standards, and guidelines
- Promptly reporting security incidents to the appropriate management area

Information Security Oversight Committee

An information protection and/or security oversight committee comprised of representatives from various areas of your organization should exist or be created if not already in existence. The members should include high-level representatives from each of your revenue business units, as well as a representative from your organization's legal, corporate auditing, human resources, physical and facilities management, and finance and accounting areas. The oversight committee should be responsible for ensuring and supporting the establishment, implementation, and maintenance of information protection awareness and training programs to assist management in the security of corporate information assets. Additionally, the committee should be kept informed of all information security-related issues, new technologies, and provide input for information security, protection costs, and budget approvals.

Corporate Auditing

The corporate auditing department should be responsible for ensuring compliance with the information protection and security policies, standards, procedures, and guidelines. They should ensure that the organizational business units are operating in a manner consistent with policies and standards, and ensure any audit plan includes a compliance review of applicable information protection policies and standards that are related to the audit topic. Additionally, a high-level management member of the corporate auditing department should take an active role in your organization's information security oversight committee.

Human Resources

Your human resources department should be responsible for providing timely information to your centrally managed information protection department, as well as the enterprise and division systems managers and application administrators, about corporate personnel terminations or transfers. They should also enforce the stated consequences of noncompliance with the corporate policies, and a high-level member of the human resources department should take an active role in your organization's information security oversight committee.

Law

Your law department should have someone assigned responsibility for reviewing your enterprise security policies and standards for legal and regulatory compliance and enforceability. Your law department should also be advised of and responsible for addressing legal issues arising from security incidents. Additionally, a high-level member of the law department should take an active role in your organization's information security oversight committee. This person should be savvy with computer and information technology and related issues; otherwise, the person will not make a positive contribution to the oversight committee, and could, in fact, create unnecessary roadblocks or stop necessary progress based upon lack of knowledge of the issues.

Managers

Your organization's line management should retain primary responsibility for identifying and protecting information and computer assets within their assigned areas of management control. When talking about a manager, we are referring to any person who has been specifically given responsibility for directing the actions of others and overseeing their work — basically, the immediate manager or supervisor of an employee. Managers have ultimate responsibility for all user IDs and information owned by company employees in the areas of their control. In the case of non-employee individuals such as contractors, consultants, etc., managers are responsible for the activity and for the company assets used by these individuals. This is usually the manager responsible for hiring the outside party. Managers have additional information protection and security responsibilities including, but not limited to, the following:

- Continually monitor the practices of employees and consultants under their control and take necessary corrective actions to ensure compliance with the organization's policies and standards.
- Inform the appropriate security administration department of the termination of any employee so that the user ID owned by that individual can be revoked, suspended, or made inaccessible in a timely manner.
- Inform the appropriate security administration department of the transfer of any employee if the transfer involves the change of access rights or privileges.
- Report any security incident or suspected incident to the centralized information protection department.
- Ensure the currency of user ID information (e.g., employee identification number and account information of the user ID owner).
- Educate the employees in their area of your organization's security policies, procedures, and standards for which they are accountable.

IT Administrators (Information Delegates)

A person, organization, or process that implements or administers security controls for the information owners are referred to as information delegates. Such information delegates typically (but not always) are part of the information technology departments with primary responsibilities for dealing with backup and recovery of the business information, applying and updating information access controls, installing and maintaining information security technology and systems, etc.

An information delegate is also any company employee who owns a user ID that has been assigned attributes or privileges associated with access control systems such as Top Secret, RACF, ACF2, etc. This user ID allows them to set system-wide security controls or administrator user IDs and information resource access rights. These security and systems administrators may report to either a business division or the central information protection department.

Information delegates are also responsible for implementing and administering security controls for corporate extended enterprise information as instructed by the information owner or delegate. Some of the responsibilities of information delegates include, but are not limited to, the following:

- Perform backups according to the backup requirements established by the information owner.
- Document backup schedule, backup intervals, storage locations, and number of backup generation copies.
- Regularly test backups to ensure they can be used successfully to restore data.
- When necessary, restore lost or corrupted information from backup media to return the application to production status.
- Perform related tape and DASD management functions as required to ensure availability of the information to the business.
- Ensure record retention requirements are met based on the information owner's analysis.
- Implement and administer security controls for corporate extended enterprise information as instructed by the information owner or delegate.
- Electronically store information in locations based on classification.
- Specifically identify the privileges associated with each system, and categorize the staff allocated to these privileges.
- Produce security log reports that will report applications and system violations and incidents to the central information protection department.
- Understand the different data environments and the impact of granting access to them.
- Ensure access requests are consistent with the information directions and security guidelines.
- Administer access rights according to criteria established by the information owners.
- Create and remove user IDs as directed by the appropriate managers.
- Administer the system within the scope of the job description and functional responsibilities.
- Distribute and follow up on security violation reports.
- Report suspected security breaches to your central information protection department.
- Give passwords of newly created user IDs to the user ID owner only.
- Maintain responsibility for day-to-day security of information.

Information Asset and Systems Owners

The information asset owner for a specific data item is a management position within the business area facing the greatest negative impact from disclosure or loss of that information. The information asset owner is ultimately responsible for ensuring that appropriate protection requirements for the information assets are defined and implemented. The information owner responsibilities include, but are not limited to, the following:

- Assign initial information classification and periodically review the classification to ensure it still meets the business needs.
- Ensure security controls are in place commensurate with the information classification.
- Review and ensure currency of the access rights associated with information assets they own.

- Determine security requirements, access criteria, and backup requirements for the information assets they own.
- Report suspected security breaches to corporate security.
- Perform, or delegate if desired, the following:
  - — Approval authority for access requests from other business units or assign a delegate in the same business unit as the executive or manager owner
  - — Backup and recovery duties or assign to the information custodian
  - — Approval of the disclosure of information
  - — Act on notifications received concerning security violations against their information assets
  - — Determine information availability requirements
  - — Assess information risks

Systems owners must consider three fundamental security areas: management controls, operational controls, and technical controls. They must follow the direction and requests of the information owners when establishing access controls in these three areas.

Information Protection

An area should exist that is responsible for determining your organization's information protection and security directions (strategies, procedures, guidelines), as approved or suggested by the information protection oversight committee, to ensure information is controlled and secured based on its value, risk of loss or compromise, and ease of recoverability. As a very high overview, some of the responsibilities of an information protection department include, but are not limited to, the following:

- Provide information security guidelines to the information management process.
- Develop a basic understanding of your organization's information to ensure proper controls are implemented.
- Provide information security design input, consulting, and review.
- Ensure appropriate security controls are built into new applications.
- Provide information security expertise and support for electronic interchange.
- Create information protection audit standards and baselines.
- Help reduce your organization's liability by demonstrating a standard of due care or diligence by following general standards or practices of professional care.
- Help ensure awareness of information protection and security issues throughout your entire organization and act as internal information security consultants to project members.
- Promote and evaluate information and computer security in IT products and services.
- Advise others within your organization of information security needs and requirements.

The remainder of this chapter includes a full discussion of the roles and related issues of the information protection department.

What Is the Role of Information Protection?

Secure information and network systems are essential to providing high-quality services to customers, avoiding fraud and disclosure of sensitive information, promoting efficient business operations, and complying with laws and regulations. Your organization must make information protection a visible, integral component of all your business operations. The best way to accomplish this is to establish a department dedicated to ensuring the protection of all your organization's information assets throughout every department and process. Information protection, or if you prefer, information security, is a very broad discipline.

Your information protection department should fulfill five basic roles:

1. Support information risk management processes.
2. Create corporate information protection policies and procedures.
3. Ensure information protection awareness and training.

4. Ensure the integration of information protection into all management practices.
5. Support your organization's business objectives.

Risk Management

Risk management is a necessary element of a comprehensive information protection and security program. What is risk management? The General Accounting Office (GAO) has a good, high-level definition: risk management is the process of assessing risk, taking steps to reduce risk to an acceptable level, and maintaining that level of risk. There are four basic principles of effective risk management.

Assess Risk and Determine Needs

Your organization must recognize that information is an essential asset that must be protected. When high-level executives understand and demonstrate that managing risks is important and necessary, it will help to ensure that security is taken seriously at lower levels in your organization and that security programs have adequate resources.

Your organization must develop practical risk assessment procedures that clearly link security to business needs. However, do not spend too much time trying to quantify the risks precisely — the difficulty in identifying such data makes the task inefficient and overly time consuming.

Your organization must hold program and business managers accountable for ensuring compliance with information protection policies, procedures, and standards. The accountability factor will help ensure that managers understand the importance of information protection and not dismiss it, considering it a hindrance.

You must manage risk on a continuing basis. As new technologies evolve, you must stay abreast of the associated risks to your information assets. And, as new information protection tools become available, you must know how such tools can help you mitigate risks within your organization.

Establish a Central Information Protection and Risk Management Focus

This is your information protection department. You must carry out key information protection risk management activities. Your information protection department will serve as a catalyst for ensuring that information security risks are considered in planned and ongoing operations. You need to provide advice and expertise to all organizational levels and keep managers informed about security issues. Information protection should research potential threats, vulnerabilities, and control techniques, and test controls, assess risks, and identify needed policies.

The information protection department must have ready and independent access to senior executives. Security concerns can often be at odds with the desires of business managers and system developers when they are developing new computer applications — they want to do so quickly and want to avoid controls that they view as impeding efficiency and convenience. By elevating security concerns to higher management levels, it helps ensure that the risks are understood by those with the most to lose from information security incidents and that information security is taken into account when decisions are made.

The information protection department must have dedicated funding and staff. Information protection budgets need to cover central staff salaries, training and awareness costs, and security software and hardware.

The central information protection department must strive to enhance its staff professionalism and technical skills. It is important in fulfilling your role as a trusted information security advisor to keep current on new information security vulnerabilities as well as new information security tools and practices.

Information and Systems Security Must Be Cost Effective

The costs and benefits of security must be carefully examined in both monetary and nonmonetary terms to ensure that the cost of controls does not exceed expected benefits. Security benefits have direct and indirect costs. Direct costs include purchasing, installing, and administering security measures, such as access control software or fire-suppression systems. Indirect costs include system performance, employee morale, and retraining requirements.

Information and Systems Security Must Be Periodically Reassessed

Security is never perfect when a system is implemented. Systems users and operators discover new vulnerabilities or ways to intentionally or accidentally circumvent security. Changes in the system or the environment

can also create new vulnerabilities. Procedures become outdated over time. All these issues make it necessary to periodically reassess the security of your organization's security.

Information Protection Policies, Procedures, Standards, and Guidelines

The information protection department must create corporate information protection policies with business unit input and support. Additionally, they must provide guidance and training to help the individual business units create their own procedures, standards, and guidelines that support the corporate information protection policies.

The Information Protection Department Must Create and Implement Appropriate Policies and Related Controls

You need to link the information protection policies you create to the business risks of your organization. The information protection policies must be adjusted on a continuing basis to respond to newly identified risks. Be sure to pay particular attention to addressing user behavior within the information protection policies.

Distinguish between information protection policies and guidelines or standards. Policies generally outline fundamental requirements that managers consider mandatory. Guidelines and standards contain more detailed rules for how to implement the policies.

It is vital to the success of the information protection policies for the oversight group and executive management to visibly support the organization's information protection policies.

Information and Systems Security Is Often Constrained by Societal Factors.

The ability of your information protection department to support the mission of your organization may be limited by various social factors depending upon the country in which your offices are located, or the laws and regulations that exist within certain locations where you do business. Know your operating environments and ensure your policies are in sync with these environments.

Awareness and Training

The information protection department must make your organization aware of information protection policies, related issues, and news on an ongoing basis. Additionally, it must provide adequate training — not only to help ensure personnel know how to address information security risks and threats, but also to keep the information protection department personnel up-to-date on the most appropriate methods of ensuring information security.

An Information Protection Department Must Promote Awareness of Information Protection Issues and Concerns throughout Your Entire Organization

The information protection department must continually educate users and others on risks and related policies. Merely sending out a memo to management once every year or two is not sufficient. Use attention-getting and user-friendly techniques to promote awareness of information protection issues. Awareness techniques do not need to be dry or boring — they should not be, or your personnel will not take notice of the message you are trying to send.

An Information Protection Department Must Monitor and Evaluate Policy and Control Effectiveness of the Policies

The information protection department needs to monitor factors that affect risk and indicate security effectiveness. One key to your success is to keep summary records of actual security incidents within your organization to measure the types of violations and the damage suffered from the incidents. These records will be valuable input for risk assessments and budget decisions. Use the results of your monitoring and record keeping to help determine future information protection efforts and to hold managers accountable for the activities and incidents that occur. Stay aware of new information protection and security monitoring tools and techniques to address the issues you find during the monitoring.

An Information Protection Department Must Extend Security Responsibilities to Those Outside Your Organization

Your organization and the systems owners have security responsibilities outside your own organization. You have a responsibility to share appropriate knowledge about the existence and extent of security measures with your external users (e.g., customers, business partners, etc.) so they can be confident that your systems are adequately secured, and so they can help to address any risks you communicate to them.

An Information Protection Department Must Make Security Responsibilities Explicit

Information and systems security responsibilities and accountability must be clearly and explicitly documented and communicated. The information security responsibilities of all groups and audiences within your organization must be communicated to them, using effective methods and on an ongoing basis.

Information Protection Must Be Integrated into Your Organization's Management Practices

Information and systems security must be an integral element of sound management practices. Ultimately, managers of the areas owning the information must decide what level of risk they are willing to accept, taking into account the cost of security controls as well as the potential financial impact of not having the security controls. The information protection department must help management understand the risks and associated costs. Information and systems security requires a comprehensive approach that is integrated within your organization's management practices. Your information protection department also needs to work with traditional security disciplines, such as physical and personnel security. To help integrate information protection within your management practices, use the following:

- Establish a process to coordinate implementation of information security measures. The process should coordinate specific information security roles and responsibilities organization-wide, and it should aid agreement about specific information security methods and processes such as risk assessment and a security classification system. Additionally, the process should facilitate coordination of organization-wide security initiatives and promote integration of security into the organizational information planning process. The process should call for implementation of specific security measures for new systems or services and include guidelines for reviewing information security incidents. Also, the process should promote visible business support for information security throughout your organization.
- Establish a management approval process to centrally authorize new IT facilities from both a business and technical standpoint.
- Make managers responsible for maintaining the local information system security environment and supporting the corporate information protection policies when they approve new facilities, systems, and applications.
- Establish procedures to check hardware and software to ensure compatibility with other system components before implementing them into the corporate systems environment.
- Create a centralized process for authorizing the use of personal information processing systems and facilities for use in processing business information. Include processes to ensure necessary controls are implemented. In conjunction with this, ensure the vulnerabilities inherent in using personal information processing systems and facilities for business purposes have been assessed.
- Ensure management uses the information protection department for specialized information security advice and guidance.
- Create a liaison between your information protection department and external information security organizations, including industry and government security specialists, law enforcement authorities, IT service providers, and telecommunications authorities, to stay current with new information security threats and technologies and to learn from the experiences of others.
- Establish management procedures to ensure that the exchange of security information with outside entities is restricted so that confidential organizational information is not divulged to unauthorized persons.
- Ensure your information protection policies and practices throughout your organization are independently reviewed to ensure feasibility, effectiveness, and compliance with written policies.

Information Protection Must Support the Business Needs, Objectives, and Mission Statement of Your Organization

Information and systems security practices must support the mission of your organization. Through the selection and application of appropriate safeguards, the information protection department will help your organization's mission by protecting its physical and electronic information and financial resources, reputation, legal position, employees, and other tangible and intangible assets. Well-chosen information security policies and procedures do not exist for their own sake — they are put in place to protect your organization's assets and support the organizational mission. Information security is a means to an end, and not an end in itself. In a private-sector business, having good security is usually secondary to the need to make a profit. With this in mind, security ought to be seen as a way to increase the firm's ability to make a profit. In a public-sector agency, security is usually secondary to the agency's provision of services to citizens. Security, in this case then, ought to be considered as a way to help improve the service provided to the public.

So, what is a good mission statement for your information protection department? It really depends upon your business, environment, company size, industry, and several other factors. To determine your information protection department's mission statement, ask yourself these questions:

- What do your personnel, systems users, and customers expect with regard to information and systems security controls and procedures?
- Will you lose valued staff or customers if information and systems security is not taken seriously enough, or if it is implemented in such a manner that functionality is noticeably impaired?
- Has any downtime or monetary loss occurred within your organization as a result of security incidents?
- Are you concerned about insider threats? Do you trust your users? Are most of your systems users local or remote?
- Does your organization keep non-public information online? What is the loss to your organization if this information is compromised or stolen?
- What would be the impact of negative publicity if your organization suffered an information security incident?
- Are there security guidelines, regulations, or laws your organization is required to meet?
- How important are confidentiality, integrity, and availability to the overall operation of your organization?
- Have the information and network security decisions that have been made been consistent with the business needs and economic stance of your organization?

To help get you started with creating your own information protection department mission statement, here is an example for you to use in conjunction with considering the previous questions:

> The mission of the information protection department is to ensure the confidentiality, integrity, and availability of the organization's information; provide information protection guidance to the organization's personnel; and help ensure compliance with information security laws and regulations while promoting the organization's mission statement, business initiatives, and objectives.

Information Protection Budgeting

How much should your organization budget for information protection? You will not like the answer; however, there is no benchmark for what information protection and security could or should cost within organizations. The variables from organization to organization are too great for such a number. Plus, it really depends upon how information protection and security costs are spread throughout your organization and where your information protection department is located within your organization.

Most information and network security spending recommendations are in extremes. The Gartner Group research in 2000 showed that government agencies spent 3.3 percent of their IT budgets on security — a significantly higher average percentage than all organizations as a whole spent on security (2.6 percent). Both numbers represent a very low amount to spend to protect an organization's information assets. Then there is the opinion of a former chief security officer at an online trading firm who believes the information security

budget should be 4 to 10 percent of *total company revenues* and not part of the IT budget at all. An October 2001 *Computerworld*/J.P. Morgan Security poll showed that companies with annual revenues of more than $500 million are expected to spend the most on security in 2002, when security-related investments will account for 11.2 percent of total IT budgets on average, compared with an average of 10.3 percent for all the users which responded to the poll. However, there are other polls, such as a 2001 survey from Metricnet, that shows that only 33 percent of companies polled after September 11, 2001, will spend more than 5 percent of their IT budgets on security. What is probably the most realistic target for information security spending is the one given by eSecurityOnline.com, which indicates information protection should be 3 to 5 percent of the company's total revenue.

Unfortunately, it has been documented in more than one news report that some CIOs do not consider information security a normal or prudent business expense. Some CFOs and CEOs have been quoted as saying information security expenses were "nuisance protection." Some decision makers need hard evidence of a security threat to their companies before they will respond. But doing nothing is not a viable option. It only takes one significant security incident to bring down a company.

When budgeting for information protection, keep in mind the facts and experiences of others. As the San Francisco-based Computer Security Institute found in its 2001 annual Computer Crime and Security Survey, 85 percent of the respondents admitted they had detected computer security breaches during the year. While only 35 percent of the respondents admitted to being able to quantify the losses, the total financial impact from these incidents was a staggering $378 million in losses.

The CIO of the Department of Energy's (DoE) Lawrence Livermore National Laboratory in Livermore, California, indicated in 2001 that security incidents had risen steadily by about 20 percent a year. Security of information is not a declining issue; it is an increasingly significant issue to address. Basically, security is a matter of existence or nonexistence for data.

So, to help you establish your information protection budget:

- *Establish need before cost.* If you know money is going to be a stumbling block, then do not lead with a budget request. Instead, break down your company's functions by business process and illustrate how these processes are tied to the company's information and network. Ask executive management, "What do you want to protect?" and then show them, "This is what it will cost to do it."
- *Show them numbers.* It is not enough to talk about information security threats in broad terms. Make your point with numbers. Track the number of attempted intrusions, security incidents, and viruses within your organization. Document them in reports and plot them on graphs. Present them monthly to your executive management. This will provide evidence of the growing information security threat.
- *Use others' losses to your advantage.* Show them what has happened to other companies. Use the annual CSI/FBI computer crime and security statistics. Give your executive managers copies of *Tangled Web* by Richard Power to show them narratives of exactly what has happened to other companies.
- *Put it in legal terms.* Corporate officers are not only accountable for protecting their businesses' financial assets, but are also responsible for maintaining critical information. Remind executive management that it has a fiduciary responsibility to detect and protect areas where information assets might be exposed.
- *Keep it simple.* Divide your budget into categories and indicate needed budgets within each. Suggested categories include:
  - — Personnel
  - — Software systems
  - — Hardware systems
  - — Awareness and training
  - — Law and regulation compliance
  - — Emerging technology research
  - — Business continuity
- *Show them where it hurts.* Simply state the impact of not implementing or funding security.

Executive Management Must Sponsor and Support Information Protection

Executive management must clearly and unequivocally support information protection and security initiatives. It must provide a role model for the rest of your organization that adhering to information protection policies and practices is the right thing to do. It must ensure information protection is built into the management framework. The management framework should be established to initiate and control the implementation of information security within your organization. Ideally, the structure of a security program should result from the implementation of a planned and integrated management philosophy. Managing computer security at multiple levels brings many benefits. The higher levels (such as the headquarters or unit levels) must understand the organization as a whole, exercise more authority, set policy, and enforce compliance with applicable policies and procedures. On the other hand, the systems levels (such as the computer facility and applications levels) know the technical and procedural requirements and problems. The information protection department addresses the overall management of security within the organization as well as corporate activities such as policy development and oversight. The system-level security program can then focus on the management of security for a particular information processing system. A central information protection department can disseminate security-related information throughout the organization in an efficient and cost-effective manner. A central information protection department has an increased ability to influence external and internal policy decisions. A central information protection department can help ensure spending its scarce security dollars more efficiently. Another advantage of a centralized program is its ability to negotiate discounts based on volume purchasing of security hardware and software.

Where Does the Information Security Role Best Fit within the Organization?

Information security should be separated from operations. When the security program is embedded in IT operations, the security program often lacks independence, exercises minimal authority, receives little management attention, and lacks resources. In fact, the GAO identified this type of organizational mode (information security as part of IT operations) as a principal basic weakness in federal agency IT security programs.

The location of the information protection department needs to be based on your organization's goals, structure, and culture. To be effective, a central information protection department must be an established part of organization management.

Should Information Protection Be a Separate Business Unit Reporting to the CEO?

This is the ideal situation. Korn/Ferry's Jim Bock, a recruiter who specializes in IT and information security placements, has noticed that more chief security officers are starting to report directly to the CEO, on a peer level to the CIO. This provides information protection with a direct line to executive management and demonstrates the importance of information security to the rest of the organization.

Should Information Protection Be a Separate Business Unit Reporting to the CIO?

This is becoming more commonplace. This could be an effective area for the information protection group. However, there exists conflict of interest in this position. Additionally, security budgets may get cut to increase spending in the other IT areas for which the CIO has responsibility. Based upon recent history and published reports, CIOs tend to focus more on technology and security; they may not understand the diverse information protection needs that extend beyond the IT arena.

Should Information Protection Be a Separate Business Unit Reporting to the CFO?

This could possibly work if the CFO also understands the information security finance issues. However, it is not likely because it is difficult (if not impossible) to show a return on investment for information security costs; so this may not be a good location for the information protection department.

Should Information Protection Exist as a Department within IT Reporting to the IT VP?

This is generally not a good idea. Not only does this create a true conflict of interest, but it also demonstrates to the rest of the organization an attitude of decreased importance of information security within the organi-

zation. It creates a competition of security dollars with other IT dollars. Additionally, it sends the message that information protection is only a technical matter and does not extend to all areas of business processes (such as hard-copy protection, voice, fax, mail, etc.).

Should Information Protection Exist as a Group within Corporate Auditing Reporting to the Corporate Auditor?

This has been attempted within several large organizations, and none that I have known of have had success with this arrangement. Not only does this create a huge conflict of interest — auditors cannot objectively audit and evaluate the same security practices the people within their same area created — but it also sends the message to the rest of the organization that information security professionals fill the same role as auditors.

Should Information Protection Exist as a Group within Human Resources Reporting to the HR VP?

This could work. One advantage of this arrangement is that the area creating the information protection policies would be within the same area as the people who enforce the policies from a disciplinary aspect. However, this could also create a conflict of interest. Also, by placing information protection within the HR area, you could send the message to the rest of the organization that information protection is a type of police unit; and it could also place it too far from executive management.

Should Information Protection Exist within Facilities Management Reporting to the Risk Management Director?

This does place all types of risk functions together, making it easier to link physical and personnel security with information security. However, this could be too far removed from executive management to be effective.

Should Information Protection Exist as a Group within IT Reporting to Middle Management?

This is probably the worst place to put the information protection group. Not only is this too far removed from executive management, but this also creates a conflict of interest with the IT processes to which information security practices apply. It also sends a message to the rest of the organization that information protection is not of significant importance to the entire organization and that it only applies to the organization's computer systems.

What Security Positions Should Exist, and What Are the Roles, Requirements, and Job Descriptions for Each?

Responsibilities for accomplishing information security requirements must be clearly defined. The information security policy should provide general guidance on the allocation of security roles and responsibilities within the organization. General information security roles and responsibilities must be supplemented with a more detailed local interpretation for specific sites, systems, and services. The security of an information system must be made the responsibility of the owner of that system. To avoid any misunderstanding about individual responsibilities, assets and security processes associated with each individual must be clearly defined. To avoid misunderstanding individual responsibilities, the manager responsible for each asset or security process must be assigned and documented. To avoid misunderstanding individual responsibilities, authorization levels must be defined and documented. Multiple levels of dedicated information security positions must exist to ensure full and successful integration of information protection into all aspects of your organization's business processes. So what positions are going to accomplish all these tasks? A few example job descriptions can be found in Exhibit 72.1. The following are some suggestions of positions for you to consider establishing within your organization:

EXHIBIT 72.1 Example Job Descriptions

The following job descriptions should provide a reference to help you create your own unique job descriptions for information security-related positions based upon your own organization's needs.

Compliance Officer

Job Description

A regulatory/compliance attorney to monitor, interpret, and communicate laws and legislation impacting regulation. Such laws and legislation include HIPAA regulations. The compliance officer will be responsible for compliance and quality control covering all areas within the information technology and operations areas. Responsibilities include:

- Quality assurance
- Approval and release of all personal health information
- HIPAA compliance oversight and implementation
- Ensuring all records and activities are maintained acceptably in accordance with health and regulatory authorities

Qualifications

- J.D. with outstanding academics and a minimum of ten years of experience
- Three to five years' current experience with healthcare compliance and regulatory issues
- In-depth familiarity with federal and state regulatory matters (Medicare, Medicaid, fraud, privacy, abuse, etc.)

Chief Security Officer

Job Description

The role of the information security department is primarily to safeguard the confidential information, assets, and intellectual property that belongs to or is processed by the organization. The scope of this position primarily involves computer security but also covers physical security as it relates to the safeguarding of information and assets. The CSO is responsible for enforcing the information security policy, creating new procedures, and reviewing existing procedures to ensure that information is handled in an appropriate manner and meets all legislative requirements, such as those set by the HIPAA security and privacy standards. The security officer must also be very familiar with anti-virus software, IP firewalls, VPN devices, cryptographic ciphers, and other aspects of computer security.

Requirements

- Experience with systems and networking security
- Experience with implementing and auditing security measures in a multi-processor environment
- Experience with data center security
- Experience with business resumption planning
- Experience with firewalls, VPNs, and other security devices
- Good communication skills, both verbal and written
- Good understanding of security- and privacy-related legislation as it applies to MMIS
- Basic knowledge of cryptography as it relates to computer security
- CISSP certification

Duties and Responsibilities

The information security department has the following responsibilities:

- Create and implement information security policies and procedures.
- Ensure that procedures adhere to the security policies.

EXHIBIT 72.1 Example Job Descriptions (continued)

- Ensure that network security devices exist and are functioning correctly where they are required (such as firewalls and software tools such as anti-virus software, intrusion detection software, log analysis software, etc.).
- Keep up-to-date on known computer security issues and ensure that all security devices and software are continuously updated as problems are found.
- Assist the operations team in establishing procedures and documentation pertaining to network security.
- Assist the engineering team to ensure that infrastructure design does not contain security weaknesses.
- Assist the facilities department to ensure that physical security is adequate to protect critical information and assets.
- Assist the customer systems administration and the professional services groups in advising clients on network security issues.
- Provide basic security training programs for all employees, and — when they access information — partners, business associates, and customers.
- In the event of a security incident, work with the appropriate authorities as directed by the executive.
- Work with external auditors to ensure that information security is adequate and evaluate external auditors to ensure that external auditors meet proper qualifications.

The Chief Security Officer has the following responsibilities:

- Ensure that the information security department is able to fulfill the above mandate.
- Hire personnel for the information security department.
- Hold regular meetings and set goals for information security personnel.
- Perform employee evaluations of information security personnel as directed by human resources.
- Ensure that information security staff receives proper training and certification where required.
- Participate in setting information security policies and procedures.
- Review all company procedures that involve information security.
- Manage the corporate information security policies and make recommendations for modifications as the needs arise.

Information Security Administrator

Job Specifications

The information security administrator will:

- Work with security analysts and application developers to code and develop information security rules, roles, policies, standards, etc.
- Analyze existing security rules to ensure no problems will occur as new rules are defined, objects added, etc.
- Work with other administrative areas in information security activities.
- Troubleshoot problems when they occur in the test and production environments.
- Define and implement access control requirements and processes to ensure appropriate information access authorization across the organizations.
- Plan and develop user administration and security awareness measures to safeguard information against accidental or unauthorized modification, destruction, or disclosure.
- Manage the overall functions of user account administration and the company-wide information security awareness training program according to corporate policies and federal regulations.
- Define relevant data security objectives, goals, and procedures.
- Evaluate data security user administration, resource protection, and security awareness training effectiveness.
- Evaluate and select security software products to support the assigned functions.
- Coordinate security software installation.
- Meet with senior management regarding data security issues.
- Participate in designing and implementing an overall data security program.
- Work with internal and external auditors as required.
- Ensure that user administration and information security awareness training programs adhere to HIPAA and other regulations.

EXHIBIT 72.1 Example Job Descriptions (continued)

Qualifications

- Human relations and communication skills to effectively interact with personnel from technical areas, internal auditors, and end users, promoting information security as an enabler and not as an inhibitor
- Decision-making ability to define data security policies, goals, and tactics, and to accurately measure these practices as well as risk assessments and selection of security devices including software tools
- Ability to organize and prioritize work to balance cost and risk factors and bring adequate data security measures to the information technology environments
- Ability to jointly establish measurable goals and objectives with staff, monitor progress on attainment of them, and adjust as required
- Ability to work collaboratively with IT and business unit management
- Ability to relate business requirements and risks to technology implementation for security-related issues
- Knowledge of role-based authorization methodologies and authentication technologies
- Knowledge of generally accepted security practices such as ISO 17799 standards
- Security administration experience
- Good communication skills
- Two to four years of security administration experience
- SSCP or CISSP certification a plus, but not required

- *Chief Security Officer.* The chief security officer (CSO) must raise security issues and help to develop solutions. This position must communicate directly with executive management and effectively communicate information security concerns and needs. The CSO will ensure security management is integrated into the management of all corporate systems and processes to assure that system managers and data owners consider security in the planning and operation of the system. This position establishes liaisons with external groups to take advantage of external information sources and to improve the dissemination of this information throughout the organization.
- *Information Protection Director.* This position oversees the information protection department and staff. This position communicates significant issues to the CSO, sets goals, and creates plans for the information protection department, including budget development. This position establishes liaisons that should be established with internal groups, including the information resources management (IRM) office and traditional security offices.
- *Information Protection Awareness and Training Manager.* This position oversees all awareness and training activities within the organization. This position communicates with all areas of the organization about information protection issues and policies on an ongoing basis. This position ensures that all personnel and parties involved with outsourcing and customer communications are aware of their security responsibilities.
- *Information Protection Technical/Network Manager.* This position works directly with the IT areas to analyze and assess risks within the IT systems and functions. This position stays abreast of new information security risks as well as new and effective information security tools. This position also analyzes third-party connection risks and establishes requirements for the identified risks.
- *Information Protection Administration Manager.* This position oversees user account and access control practices. This person should have a wide experience range over many different security areas.
- *Privacy Officer.* This position ensures the organization addresses new and emerging privacy regulations and concerns.
- *Internal Auditor.* This position performs audits within the corporate auditing area in such a way as to ensure compliance with corporate information protection policies, procedures, and standards.
- *Security Administrator.* The systems security administrator should participate in the selection and implementation of appropriate technical controls and security procedures, understand system vulnerabilities, and be able to respond quickly to system security problems. The security administrator is responsible for the daily administration of user IDs and system controls, and works primarily with the user community.

- *Information Security Oversight Committee.* This is a management information security forum established to provide direction and promote information protection visibility. The committee is responsible for review and approval of information security policy and overall responsibilities. Additionally, this committee is responsible for monitoring exposure to major threats to information assets, for reviewing and monitoring security incidents, and for approving major initiatives to enhance information security.

How Do You Effectively Maintain Separation of Duties?

When considering quality assurance for computer program code development, the principles of separation of duty are well-established. For example, the person who designs or codes a program must not be the only one to test the design or the code. You need similar separation of duties for information protection responsibilities to reduce the likelihood of accidental compromise or fraud. A good example is the 1996 Omega case where the network administrator, Tim Lloyd, was an employee who was responsible for everything to do with the manufacturing of computers. As a result, when Lloyd was terminated, he was able to add a line of program code to a major manufacturing program that ultimately deleted and purged all the programs in the system. Lloyd also had erased all the backup tapes, for which he also had complete control. Ultimately, the company suffered $12 million in damages, lost its competitive footing in the high-tech instrument and measurement market, and 80 employees lost their jobs as a result. If separation of duties had been in place, this could have been avoided.

Management must be become active in hiring practices (ensuring background checks) bonding individuals (which should be routine for individuals in all critical areas) and auditing and monitoring, which should be routine practices. Users should be recertified to resources, and resources to users, at least annually to ensure proper access controls are in place. Because the system administration group is probably placed within the confines of the computer room, an audit of physical and logical controls also needs to be performed by a third party.

Certain information protection duties must not be performed by the same person or within one area. For example, there should be separation of roles of systems operators, systems administrators, and security administrators, and separation of security-relevant functions from others. Admittedly, ideal separation can be costly in time and money, and often possible only within large staffs. You need to make information security responsibilities dependent upon your business, organization size, and associated risks. You must perform risk assessment to determine what information protection tasks should be centralized and what should be distributed. When considering separation of duties for information security roles, it is helpful to use a tool similar to the one in Exhibit 72.2.

How Large Should the Information Protection/Security Department Be?

Ah, if only there were one easy answer to the question of how large an information protection department should be. This is one of the most commonly asked questions I have heard at information security conferences over the past several years, and I have seen this question asked regularly within all the major information security companies. There is no "best practice" magic number or ratio. The size of an information protection department depends on many factors. These include, but are not limited to, the following:

- Industry
- Organization size
- Network diversification and size
- Number of network users
- Geographical locations
- Outsourced functions

Whatever size you determine is best for your organization, you need to ensure the staff you choose has a security background or, at least, has some basic security training.

Summary

This chapter reviewed a wide range of issues involved in creating an information protection program and department. Specifically:

EXHIBIT 72.2 Application Roles and Privileges Worksheet

Application System _____
Purpose/Description _____
Information Owner _____
Application/System Owner _____
Implementation Date _____

Role/Function	Group/Persons	Access Rights	Comments
User Account Creation			
Backups			
Testing			
Production Change Approvals			
Disaster Recovery Plans			
Disable User Accounts			
Incident Response			
Error Correction			
End-User Training			
Application Documentation			
Quality Assurance			
User Access Approvals			

- Organizational information protection responsibilities
- Roles of an information protection department
- Information protection budgeting
- Executive management support of information protection
- Where to place the information protection department within your organization
- Separation of information security duties
- Descriptions of information protection responsibilities

Accompanying this chapter is a tool to help you determine separation of information security duties (Exhibit 72.2) and some examples of information protection job descriptions to help you get your own written (Exhibit 72.1).

References

The following references were used to collect and support much of the information within this chapter, as well as a general reference for information protection practices. Other information was gathered from discussions with clients and peers throughout my years working in information technology as well as from widely publicized incidents related to information protection.

1. National Institute of Standards and Technology (NIST) publication, *Management of Risks in Information Systems: Practices of Successful Organizations*.
2. NIST publication, CSL Bulletin, August 1993, *Security Program Management*.
3. NIST *Generally Accepted System Security Principles* (GSSPs).
4. ISO 17799.
5. Organization for Economic Cooperation and Development's (OECD), *Guidelines for the Security of Information Systems*.
6. Computer Security Institute (CSI) and FBI joint annual *Computer Crime and Security Survey*.
7. *CIO Magazine*, 1-17-2002, The security spending mystery, by Scott Berinato.
8. *CIO Magazine*, 12-6-2001, Will security make a 360-degree turn?, by Sarah D. Scalet.
9. *CIO Magazine*, 8-9-2001, Another chair at the table, by Sarah D. Scalet.
10. *CIO Magazine*, 10-1-200, Protection money, by Tom Field.

73

Organizing for Success: Some Human Resources Issues in Information Security

Jeffrey H. Fenton, CBCP, CISSP and James M. Wolfe, MSM

In a holistic view, information security is a triad of people, process, and technology. Appropriate technology must be combined with management support, understood requirements, clear policies, trained and aware users, and plans and processes for its use. While the perimeter is traditionally emphasized, threats from inside have received less attention. Insider threats are potentially more serious because an insider already has knowledge of the target systems. When dealing with insider threats, people and process issues are paramount. Also, too often, security measures are viewed as a box to install (technology) or a one-time review. Security is an ongoing process, never finished.

This chapter focuses on roles and responsibilities for performing the job of information security. Roles and responsibilities are part of an operationally excellent environment, in which people and processes, along with technology, are integrated to sustain security on a consistent basis. *Separation of responsibilities*, requiring at least two persons with separate job duties to complete a transaction or process end-to-end, or avoiding a conflict of interest, is also introduced as part of organizing for success. This concept originated in accounting and financial management; for example, not having the same person who approves a purchase also able to write a check. The principle is applied to several roles in information technology (IT) development and operations, as well as the IT system development life cycle. All these principles support the overall management goal to protect and leverage the organization's information assets.

Information Security Roles and Responsibilities

This section introduces the functional components of information security, from a role and responsibility perspective, along with several other IT and business functional roles. Information security is much more than a specialized function; it is everyone's responsibility in any organization.

The Business Process Owner, Information Custodian, and End User

The *business process owner* is the manager responsible for a business process such as supply-chain management or payroll. This manager would be the focal point for one or more IT applications and data supporting the processes. The process owner understands the business needs and the value of information assets to support them. The International Standard ISO 17799, *Information Security Management*, defines the role of the information asset owner responsible for maintaining the security of that asset.[1]

The *information custodian* is an organization, usually the internal IT function or an outsourced provider, responsible for operating and managing the IT systems and processes for a business owner on an ongoing

basis. The business process owner is responsible for specifying the requirements for that operation, usually in the form of a service level agreement (SLA). While information security policy vests ultimate responsibility in business owners for risk management and compliance, the day-to-day operation of the compliance and risk mitigation measures is the responsibility of information custodians and end users.

End users interact with IT systems while executing business functional responsibilities. End users may be internal to the organization, or business partners, or end customers of an online business. End users are responsible for complying with information security policy, whether general, issue-specific, or specific to the applications they use. Educating end users on application usage, security policies, and best practices is essential to achieving compliance and quality.

In an era of budget challenges for the information security functions, the educated and committed end user is an information security force multiplier for defense-in-depth. John Weaver, in a recent essay, "Zen and Information Security,"[2] recommends turning people into assets. For training and awareness, this includes going beyond rules and alerts to make security "as second nature as being polite to customers," as Neal O'Farrell noted in his recent paper, "Employees: Your Best Defense, or Your Greatest Vulnerability?"[3] All users should be trained to recognize potential social engineering. Users should also watch the end results of the business processes they use. Accounting irregularities, sustained quality problems in manufacturing, or incorrect operation of critical automated temperature-control equipment could be due to many causes, including security breaches. When alert end users notice these problems and solve them in a results-oriented manner, they could identify signs of sabotage, fraud, or an internal hacker that technical information security tools might miss. End users who follow proper practices and alert management of suspicious conditions are as important as anti-virus software, intrusion detection, and log monitoring. Users who learn this holistic view of security can also apply the concepts to their homes and families.[4]

In today's environment, users include an increasing proportion of *non-employee* users, including temporary or contract workers, consultants, outsourced provider personnel, and business-partner representatives. Two main issues with non-employee users are nondisclosure agreements (NDAs) and the process for issuing and deleting computer accounts. Non-employee users should be treated as business partners, or representatives of business partners, if they are given access to systems on the internal network. This should include a written, signed NDA describing their obligations to protect sensitive information. In contrast with employees, who go through a formal human resources (HR) hiring and separation process, non-employee users are often brought in by a purchasing group (for temporary labor or consulting services), or they are brought in by the program manager for a project or outsourced activity. While a formal HR information system (HRIS) can alert system administrators to delete computer accounts when *employees* leave or transfer, *non-employees* who do not go through the HRIS would not generate this alert. Removing computer accounts for departed non-employees is an weak operational link in many organizations.

Information Security Functions

Information security functions fall into five main categories — policy/strategy/governance, engineering, disaster recovery/business continuity (DR/BC), crisis management and incident response/investigation, and administrative/operational (see Exhibit 73.1). In addition, information security functions have many interfaces with other business functions as well as with outsource providers, business partners, and other outside organizations.

Information security policy, strategy, and governance functions should be organized in an information security department or directorate, headed by an information security manager or director who may also be known as the chief information security officer (CISO). This individual directs, coordinates, plans, and organizes information security activities throughout the organization, as noted by Charles Cresson Wood.[5] The information security function must work with many other groups within and outside the organization, including physical security, risk management (usually an insurance-related group in larger companies), internal audit, legal, internal and external customers, industry peers, research groups, and law enforcement and regulatory agencies.

Within the information security function, policy and governance include the development and interpretation of written information security policies for the organization, an education and awareness program for all users, and a formal approval and waiver process. Any deviation from policy represents a risk above the acceptable level represented by compliance with policy. Such deviations should be documented with a formal waiver approval, including the added risk and additional risk mitigation measures applied, a limited term, and a plan to achieve compliance. Ideally, all connections between the internal network and any outside entity should be consolidated as much as possible through one or a few gateways and demilitarized zones (DMZs), with a

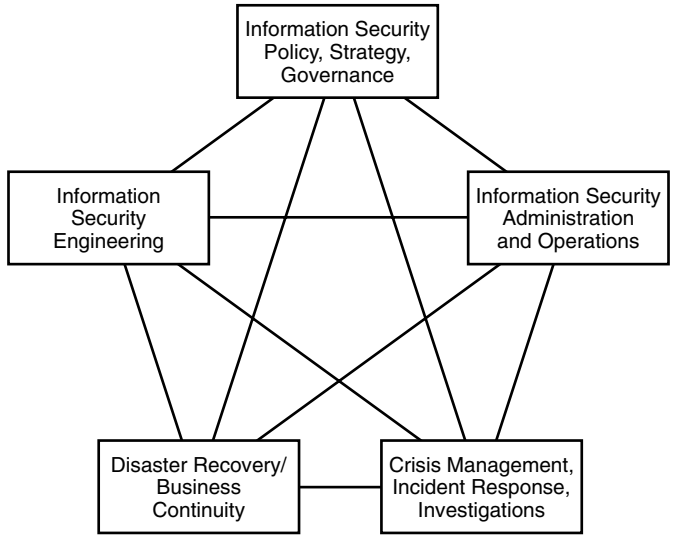


EXHIBIT 73.1 Five information security roles.

standard architecture and continuous monitoring. In very large organizations with decentralized business units, this might not be possible. When business units have unique requirements for external connectivity, those should be formally reviewed and approved by the information security group before implementation.

The security strategy role, also in the central information security group, includes the identification of long-term technology and risk trends driving the evolution of the organization's security architecture. The information security group should develop a security technology roadmap, planning for the next five years the organization's need for security technologies driven by risk management and business needs. Once the roadmap is identified, the security group would be responsible for identifying and integrating the products to support those capabilities. Evaluating new products is another part of this activity, and a formal test laboratory should be provided. In larger IT organizations, the security strategy function would work closely with an overall IT strategy function. The information security group should have project responsibility to execute all security initiatives that affect the entire organization.

Information security engineering is the function of identifying security requirements and bringing them to realization when a specific network or application environment is newly developed. While the information security group would set the policies as part of the policy and governance function, security engineers would assess the risks associated with a particular program (such as implementing a new enterprise resource planning [ERP] system), identify the applicable policies, and develop a system policy for the system or application environment. Working through the system development life cycle, engineers would identify requirements and specifications, develop the designs, and participate in the integration and testing of the final product. Engineering also includes developing the operational and change-control procedures needed to maintain security once the system is fielded. Information security engineering may be added to the central information security group, or it may be organized as a separate group (as part of an IT systems engineering function).

Disaster recovery/business continuity (DR/BC) includes responding to and recovering from disruptive incidents. While DR involves the recovery of IT assets, BC is broader and includes recovery of the business functions (such as alternative office space or manufacturing facilities). While DR and BC began by focusing on physical risks to availability, especially natural disasters, both disciplines have broadened to consider typically nonphysical events such as breaches of information confidentiality or integrity. Much of the planning component of DR/BC can utilize the same risk assessment methods as for information security risk assessments. In large organizations, the DR/BC group is often separate from the central information security group, and included in an operational IT function, because of DR's close relationship to computer operations and backup procedures. Because of the convergence of DR/BC applicability and methods with other information security disciplines, including DR/BC in the central information security group is a worthwhile option.

Crisis management is the overall discipline of planning for and responding to emergencies. Crisis management in IT began as a component of DR. With the broadening of the DR/BC viewpoint, crisis management needs to cover incident types beyond the traditional physical or natural disasters. For all types of incidents, similar principles can be applied to build a team, develop a plan, assess the incident at the onset and identify its severity, and match the response to the incident. In many organizations, the physical security and facilities functions have developed emergency plans, usually focusing on physical incidents or natural disasters, separate from the DR plans in IT. For this reason, an IT crisis management expert should ensure that IT emergency plans are integrated with other emergency plans in the organization. With the broadening of *crisis* to embrace nonphysical information security incidents, the integrative role must also include coordinating the separate DR plans for various IT resources. During certain emergencies, while the emergency team is in action, it may be necessary to weigh information security risks along with other considerations (such as rapidly returning IT systems or networks to service). For this reason, as well as for coordinating the plans, the integrative crisis management role should be placed in the central information security group. Information security crisis management can also include working with the public relations, human resources, physical security, and legal functions as well as with suppliers, customers, and outside law enforcement agencies.

Incident response has already been noted as part of crisis management. Many information security incidents require special response procedures different from responding to a physical disaster. These procedures are closely tied to monitoring and notification, described in the next two paragraphs. An organization needs to plan for responding to various types of information security attacks and breaches, depending on their nature and severity. Investigation is closely related to incident response, because the response team must identify when an incident might require further investigation after service is restored. Investigation is fundamentally different in that it takes place after the immediate emergency is resolved, and it requires evidence collection and custody procedures that can withstand subsequent legal scrutiny. Along with this, however, the incident response must include the processes and technology to collect and preserve logs, alerts, and data for subsequent investigation. These provisions must be in place and operational before an incident happens. The investigation role may be centralized in the information security group, or decentralized in large organizations provided that common procedures are followed. If first-line investigation is decentralized to business units in a large corporation, there should be a central information security group specialist to set technical and process direction on incident response planning and investigation techniques. For all incidents and crises, the lessons learned must be documented — not to place blame but to prevent future incidents, improve the response, and help the central information security group update its risk assessment and strategy.

Information security administration and operations include account management, privilege management, security configuration management (on client systems, servers, and network devices), monitoring and notification, and malicious code and vulnerability management. These administrative and operational functions are diverse, not only in their content but also in who performs them, how they are performed, and where they reside organizationally. Account and privilege management include setting up and removing user accounts for all resources requiring access control, and defining and granting levels of privilege on those systems. These functions should be performed by a central security operations group, where possible, to leverage common processes and tools as well as to ensure that accounts are deleted promptly when users leave or transfer. In many organizations, however, individual system administrators perform these tasks. Security configuration management includes configuring computer operating systems and application software, and network devices such as routers and firewalls, with security functions and access rules. This activity actually implements much of the organization's security policy. While the central information security group owns the policy, configuration management is typically distributed among system administrators and telecommunication network administrators. This is consistent with enabling the central information security group to focus on its strategic, policy, and governance roles.

Monitoring and notification should also be part of a central security operations function, with the ability to "roll up" alerts and capture logs from systems and network devices across the enterprise. Intrusion detection systems (IDSs) would also be the responsibility of this group. In many large organizations, monitoring and notification are not well integrated, with some locally administered systems depending on their own system administrators who are often overworked with other duties. As noted earlier, monitoring and notification processes and tools must meet the needs of incident response. The additional challenges of providing 24/7 coverage are also noted below.

Malicious code and vulnerability management includes deploying and maintaining anti-virus software, isolating and remediating infected systems, and identifying and correcting security vulnerabilities (in operating systems, software applications, and network devices). These activities require centrally driven technical and process disciplines. It is not enough only to expect individual desktop users to keep anti-virus software updated and individual system administrators to apply patches. A central group should test and *push* anti-virus updates. The central group should also test patches on representative systems in a laboratory and provide a central repository of alerts and patches for system and network administrators to deploy. Malicious code management is also closely tied to incident response. With the advent of multifunctional worms, and exploits appearing quickly after vulnerabilities become known, an infection could easily occur before patches or anti-virus signatures become available. In some cases, anomaly-based IDSs can detect unusual behavior before patches and signatures are deployed, bringing malicious code and vulnerability management into a closer relationship with monitoring. These central activities cross several functional boundaries in larger IT organizations, including e-mail/messaging operations, enterprise server operations, and telecommunications, as well as security operations. One approach is establishing a cross-functional team to coordinate these activities, with technical leadership in the central information security organization.

Distributed Information Security Support in Larger Organizations

Some of the challenges of providing security support in a large organization, especially a large corporation with multiple business units, have already been noted. Whether IT functions in general are centralized or distributed reflects the culture of the organization as well as its business needs and technology choices. In any organization, presenting the business value of the information security functions is challenging. Beyond simply preventing bad things from happening, security is an enabler for E-business. To make this case, the central information security group needs to partner with the business as its internal customer. Building a formal relationship with the business units in a large enterprise is strongly recommended.

This relationship can take the shape of a formal information protection council, with a representative from each division or business unit. The representative's role, which must be supported by business unit management, would include bringing the unique technical, process, and people concerns of security, as viewed by that business unit, to the information security group through two-way communication. The representatives can also assist in security training and awareness, helping to push the program to the user community. Representatives can also serve in a first-line role to assist their business units with the approval and waiver requests described earlier.

Information Security Options for Smaller Organizations

The most important information security problem in many smaller organizations is the lack of an information security function and program. Information security must have an individual (a manager, director, or CISO) with overall responsibility. Leaving it to individual system administrators, without policy and direction, will ensure failure. Once this need is met, the next challenge is to scale the function appropriately to the size and needs of the business. Some of the functions, which might be separate groups in a large enterprise, can be combined in a smaller organization. Security engineering and parts of security operations (account and privilege management, monitoring and notification, incident response, crisis management, and DR) could be combined with the policy, governance, and user awareness roles into the central information security group. The hands-on security configuration management of desktops, servers, and network devices should still be the separate responsibility of system and network administrators. In the earlier discussion, the role of an in-house test laboratory, especially for patches, was noted. Even in a smaller organization, it is strongly recommended that representative test systems be set aside and patches be tested by a system administrator before deployment.

For smaller organizations, there are special challenges in security strategy. In a smaller enterprise, the security technology roadmap is set by technology suppliers, as the enterprise depends on commercial off-the-shelf (COTS) vendors to supply all its products. Whatever the COTS vendors supply becomes the *de facto* security strategy for the enterprise. To a great extent, this is still true in large enterprises unless they have a business case to, and have or engage the expertise to, develop some of their own solutions. While a large enterprise can exert some influence over its suppliers, and should develop a formal technology strategy, smaller enterprises

should not overlook this need. If a smaller enterprise cannot justify a strategy role on a full-time basis, it could consider engaging external consultants to assist with this function initially and on a periodic review basis. Consultants can also support DR plan development. As with any activity in information security, doing it once is not enough. The strategy or the DR plan must be maintained.

Internal and External Audit

The role of auditors is to provide an independent review of controls and compliance. The central information security group, and security operational roles, should not audit their own work. To do so would be a conflict of interest. Instead, auditors provide a crucial service because of their independence. The central information security group should partner with the internal audit organization to develop priorities for audit reviews based on risk, exchange views on the important risks to the enterprise, and develop corrective action plans based on the results of past audits. The audit organization can recognize risks based on what it sees in audit results. External auditors may be engaged to provide a second kind of independent review. For external engagements, it is very important to specify the scope of work, including the systems to be reviewed, attributes to be reviewed and tested, and processes and procedures for the review. These ground rules are especially important where vulnerability scanning or penetration testing is involved.

Outsourcing Providers

Outsourcing providers offer services for a variety of information security tasks, including firewall management and security monitoring. Some Internet service providers (ISPs) offer firewall and VPN management. Outsourcing firewall management can be considered if the organization's environment is relatively stable, with infrequent changes. If changes are frequent, an outsourcing provider's ability to respond quickly can be a limiting factor. In contrast, 24/7 monitoring of system logs and IDSs can be more promising as an outsource task. Staffing one seat 24/7 requires several people. This is out of reach for smaller organizations and a challenge in even the largest enterprises. An outsourcing provider for monitoring can leverage a staff across its customer base. Also, in contrast with the firewall, where the organization would trust the provider to have privileged access to firewalls, monitoring can be done with the provider having no interactive access to any of the customer's systems or network devices. In all consulting and outsourcing relationships, it is essential to have a written, signed NDA to protect the organization's sensitive information. Also, the contract must specify the obligations of the provider when the customer has an emergency. If an emergency affects many of the same provider's customers, how would priority be determined?

To Whom Should the Information Security Function Report?

Tom Peltier, in a report for the Computer Security Institute,[6] recommends that the central information security group report as high as possible in the organization, at least to the chief information officer (CIO). The group definitely should *not* be part of internal audit (due to the potential for conflict of interest) or part of an operational group in IT. If it were part of an operational group, conflict of interest could also result. Peltier noted that operational groups' top priority is maintaining maximum system uptime and production schedules. This emphasis can work against implementing and maintaining needed security controls. The central information security group should also never be part of an IT system development group because security controls are often viewed as an impediment or an extra cost add-on to development projects. A security engineer should be assigned from the security engineering group to support each development project.

There are several issues around having the central information security group as part of the physical security organization. This can help with investigations and crisis management. The drawbacks are technology incompatibility (physical security generally has little understanding of IT), being perceived *only* as preventing bad things from happening (contrast with the business enabler viewpoint noted earlier), and being part of a group that often suffers budget cuts during difficult times. Tracy Mayor[7] presented a successful experience with a single organization combining physical security and information security. Such an organization could be headed by a chief security officer (CSO), reporting to the chief executive officer (CEO), placing the combined group at the highest level. The combined group could also include the risk management function in large enterprises, an activity usually focused on insurance risks. This would recognize the emerging role of insurance

for information security risks. The model can work but would require cultural compatibility, cross-training, management commitment, and a proactive partnership posture with customers. Another alternative, keeping information security and physical security separate, is to form a working partnership to address shared issues, with crisis management as a promising place to begin. Similarly, the CISO can partner with the risk management function.

Although the DR/BC function, as noted earlier, might be part of an operational group, DR/BC issues should be represented to upper management at a comparable level to the CISO. The CISO could consider making DR/BC a component of risk management in security strategy, and partnering with the head of the DR/BC group to ensure that issues are considered and presented at the highest level. Ed Devlin has recommended[8] that a BC officer, equal to the CISO, reports at the same high level.

Filling the Roles: Remarks on Hiring Information Security Professionals

One of the most difficult aspects of information security management is finding the right people for the job. What should the job description say? Does someone necessarily need specific information security experience? What are the key points for choosing the best candidate? Answering these questions will provide a clearer picture of how to fill the role effectively.

Note: This section outlines several procedures for identifying and hiring job candidates. It is strongly recommended to review these procedures with your human resources team and legal advisors before implementing them in your environment.

Job Descriptions

A description of the position is the starting point in the process. This job description should contain the following:[9]

- The position title and functional reporting relationship
- The length of time the candidate search will be open
- A general statement about the position
- An explicit description of responsibilities, including any specific subject matter expertise required (such as a particular operating system or software application)
- The qualifications needed, including education
- The desired attributes wanted
- Job location (or telecommuting if allowed) and anticipated frequency of travel
- Start date
- A statement on required national security clearances (if any)
- A statement on requirements for U.S. citizenship or resident alien status, if the position is associated with a U.S. Government contract requiring such status
- A statement on the requirements for a background investigation and the organization's drug-free workplace policy

Other position attributes that could be included are:

- Salary range
- Supervisor name
- Etc.

The general statement should be two to three sentences, giving the applicant some insight into what the position is. It should be an outline of sorts for the responsibilities section. For example:

> **General:** The information security specialist (ISS) uses current computer science technologies to assist in the design, development, evaluation, and integration of computer systems and networks to maintain system security. Using various tools, the ISS will perform penetration and vulnerability

analyses of corporate networks and will prepare reports that may be submitted to government regulatory agencies.

The most difficult part of the position description is the responsibilities section. To capture what is expected from the new employee, managers are encouraged to engage their current employees for input on the day-to-day activities of the position. This accomplishes two goals. First, it gives the manager a realistic view of what knowledge, skills, and abilities will be needed. Second, it involves the employees who will be working with the new candidate in the process. This can prevent some of the difficulties current employees encounter when trying to accept new employees. More importantly, it makes them feel a valued part of the process. Finally, this is more accurate than reusing a previous job description or a standard job description provided by HR. HR groups often have difficulty describing highly technical jobs. An old job description may no longer match the needs of a changing environment. Most current employees are doing tasks not enumerated in the job descriptions when they were hired.

Using the above general statement, an example of responsibilities might be:

- Evaluate new information security products using a standard image of the corporate network and prepare reports for management.
- Represent information security in the design, development, and implementation of new customer secured networks.
- Assist in customer support issues.
- Using intrusion detection tools; test the corporation's network for vulnerabilities.
- Assist government auditors in regulatory compliance audits.

Relevant Experience

When hiring a new security professional, it is important to ensure that the person has the necessary experience to perform the job well. There are few professional training courses for information security professionals. Some certification programs, such as the Certified Information System Security Professional (CISSP),[10] require experience that would not be relevant for an entry-level position. In addition, Lee Kushner noted, "… while certification is indeed beneficial, it should be looked on as a valuable enhancement or add-on, as opposed to a prerequisite for hiring."[11] Several more considerations can help:

- Current information security professionals on the staff can describe the skills they feel are important and which might be overlooked.
- Some other backgrounds can help a person transition into an information security career:
  - — Auditors are already trained in looking for minute inconsistencies.
  - — Computer sales people are trained to know the features of computers and software. They also have good people skills and can help market the information security function.
  - — Military experience can include thorough process discipline and hands-on expertise in a variety of system and network environments. Whether enlisted or officer grade, military personnel are often given much greater responsibility (in numbers supervised, value of assets, and criticality of missions) than civilians with comparable years of experience.
  - — A candidate might meet all qualifications except for having comparable experience on a different operating system, another software application in the same market space, or a different hardware platform. In many cases, the skills are easily transferable with some training for an eager candidate.
- A new employee might have gained years of relevant experience in college (or even in high school) in part-time work. An employee with experience on legacy systems may have critical skills difficult to find in the marketplace. Even if an employee with a legacy system background needs retraining, such an employee is often more likely to want to stay and grow with an organization. For a new college graduate, extracurricular activities that demonstrate leadership and discipline, such as competing in intercollegiate athletics while maintaining a good scholastic record, should also be considered.

The Selection Process

Selecting the best candidate is often difficult. Current employees should help with interviewing the candidates. The potential candidates should speak to several, if not all, of the current employees. Most firms use interviews,

yet the interview process is far from perfect. HR professionals, who have to interview candidates for many kinds of jobs, are not able to focus on the unique technical needs of information security. Any interview process can suffer from stereotypes, personal biases, and even the order in which the candidates are interviewed. Having current employees perform at least part of the interview can increase its validity.[12] Current employees can assess the candidate's knowledge with questions in their individual areas of expertise. Two additional recommendations are:

1. Making sure the interviews are structured with the same list of general questions for each candidate
2. Using a candidate score sheet for interviewers to quantify their opinions about a candidate

A good place to start is the required skills section and desired skills section of the position description. The required skills should be weighted about 70 percent of the score sheet, while the desired skills should be about 30 percent.

Filling an open position in information security can be difficult. Using tools like the position description[13] and the candidate score sheet (see Exhibits 73.2 and 73.-3) can make selecting a new employee much easier. Having current employees involved throughout the hiring process is strongly recommended and will make choosing the right person even easier.

Because information security personnel play a critical and trusted role in the organization, criminal and financial background checks are essential. Eric Shaw et al.[14] note that candidates should also be asked about past misuse of information resources. Resumes and references should be checked carefully. The same clearance procedures should apply to consultants, contractors, and temporary workers, depending on the access privileges they have. ISO 17799[15] also emphasizes the importance of these measures. Shaw and co-authors recommend working with HR to identify and intervene effectively when any employee (regardless of whether in information security) exhibits at-risk conduct. Schlossberg and Sarris[16] recommend repeating background checks annually for existing employees. HR and legal advisors must participate in developing and applying the background check procedures.

When Employees and Non-Employees Leave

The issue of deleting accounts promptly when users leave has already been emphasized. Several additional considerations apply, especially if employees are being laid off or any departure is on less than amicable terms. Anne Saita[17] recommends moving critical data to a separate database, to which the user(s) leaving does(do) not have access. Users leaving must be reminded of their NDA obligations. Saita further notes that the users' desktop computers could also contain backdoors and should be disconnected. Identifying at-risk behavior, as noted earlier, is even more important for the employees still working after a layoff who could be overworked or resentful.

Separation of Responsibilities

Separation of responsibilities, or segregation of duties, originated in financial internal control. The basic concept is that no single individual has complete control over a sequence of related transactions.[18] A 1977 U.S. federal law, the Foreign Corrupt Practices Act,[19] requires all corporations registering with the Securities and Exchange Commission to have effective internal accounting controls. Despite its name, this law applies even if an organization does no business outside the United States.[20] When separation of duties is enforced, it is more difficult to defraud the organization because two or more individuals must be involved and it is more likely that the conduct will be noticed.

In the IT environment, separation of duties applies to many tasks. Vallabhaneni[21] noted that computer operations should be separated from application programming, job scheduling, the tape library, the help desk, systems programming, database programming, information security, data entry, and users. Information security should be separate from database and application development and maintenance, system programming, telecommunications, data management or administration, and users. System programmers should never have access to application code, and application programmers should not have access to live production data. Kabay[22] noted that separation of duties should be applied throughout the development life cycle so that the person who codes a program would not also test it, test systems and production systems are separate, and operators cannot modify production programs. ISO 17799 emphasizes[23] that a program developer or tester with access to the production system could make unauthorized changes to the code or to production data. Conversely, compilers and other system utilities should also not be accessible from production systems. The earlier

EXHIBIT 73.2 Sample Position Description

Job Title: Information Security Specialist Associate
Pay Range: $40,000 to $50,000 per year
Application Date:01/25/03–02/25/03
Business Unit: Data Security Assurance
Division: Computing Services
Location: Orlando, FL
Supervisor: John Smith

General:

The Information Security Specialist Associate uses current computer science technologies to assist in the design, development, evaluation, and integration of computer systems and networks to maintain system security. Using various tools, the information security specialist associate will perform penetration and vulnerability analyses of corporate networks and will prepare reports that may be submitted to government regulatory agencies.

Responsibilities:

- Evaluate new information security products using a standard image of the corporate network and prepare reports for management.
- Represent information security in the design, development, and implementation of new customer secured network.
- Assist in day-to-day customer support issues.
- Using intrusion detection tools, test the corporation's network for vulnerabilities.
- Provide security and integration services to internal and commercial customers.
- Build and maintain user data groups in the Win NT environment.
- Add and remove user Win NT accounts.
- Assist government auditors in regulatory compliance audits.

Required Education/Skills:

- Knowledge of Windows, UNIX, and Macintosh operating systems
- Understanding of current networking technologies, including TCP/IP and Banyan Vines
- Microsoft Certified Systems Engineer certification
- Bachelor's degree in computer science or relevant discipline

Desired Education/Skills:

- Two years of information security experience
- MBA
- CISSP certification

discussion of system administration and security operations noted that account and privilege management should be part of a central security operations group separate from local system administrators. In a small organization where the same person might perform both these functions, procedures should be in place (such as logging off and logging on with different privileges) to provide some separation.[24]

Several related administrative controls go along with separation of duties. One control is requiring mandatory vacations each year for certain job functions. When another person has to perform a job temporarily, a fraud perpetrated by the regular employee might be noticed. Job rotation has a similar effect.[25] Another approach is dual control, requiring two or more persons to perform an operation simultaneously, such as accessing emergency passwords.[26]

Separation of duties helps to implement the principle of *least privilege*.[27] Each user is given only the minimum access needed to perform the job, whether the access is logical or physical. Beyond IT positions, every position that has any access to sensitive information should be analyzed for sensitivity. Then the security requirements of each position can be specified, and appropriately controlled access to information can be provided. When each position at every level is specified in this fashion, HR can focus background checks and other safeguards on the positions that truly need them. Every worker with access to sensitive information has security respon-

EXHIBIT 73.3 Candidate Score Sheet

Candidate Name: **Fred Jones**
Date: **1/30/2003**
Position: **Information Security Specialist Associate**

Required Skill	Knowledge Level[a]	Multiplier	Score
OS knowledge	2	0.2	0.4
Networking knowledge	2	0.2	0.4
Bachelor's degree	3	0.2	0.6
MCSE	2	0.1	0.2
Desired skill			
InfoSec experience	0	0.1	0
MBA	2	0.1	0.2
CISSP	0	0.1	0
Total			1.8

[a] Knowledge Level:
0 — Does not meet requirement
1 — Partially meets requirement
2 — Meets requirement
3 — Exceeds requirement

Knowledge level × Multiplier = Score

Note: It is strongly recommended to review your procedures with your human resources team and legal advisors.

sibilities. Those responsibilities should be made part of the job description[28] and briefed to the user annually with written sign-off.

Summary

This chapter has presented several concepts on the human side of information security, including:

- Information security roles and responsibilities, including user responsibilities
- Information security relationships to other groups in the organization
- Options for organizing the information security functions
- Staffing the information security functions
- Separation of duties, job sensitivity, and least privilege

Security is a triad of people, process, and technology. This chapter has emphasized the people issues, the importance of good processes, and the need to maintain security continuously. The information security function has unique human resources needs. Attention to the people issues throughout the enterprise helps to avoid or detect many potential security problems. Building processes based on separation of duties and least privilege helps build in controls organic to the organization, making security part of the culture while facilitating the business. Secure processes, when understood and made part of each person's business, are a powerful complement to technology. When the organization thinks and acts securely, the job of the information security professional becomes easier.

References

1. British Standard 7799/ISO Standard 17799: *Information Security Management,* London: British Standards Institute, 1999, Section 4.1.3.
2. Weaver, John, Zen and information security, available online at http://www.infosecnews.com/opinion/2001/12/19_03.htm.

3. O'Farrell, Neal, Employees: your best defense, or your greatest vulnerability?," in SearchSecurity.com, available online at (http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci771517,00.html).
4. O'Farrell, Neal, Employees: your best defense, or your greatest vulnerability?," in SearchSecurity.com, available online at (http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci771517,00.html).
5. Wood, Charles Cresson, *Information Security Roles & Responsibilities Made Easy,* Houston: PentaSafe, 2001, p. 72.
6. Peltier, Tom, Where should information protection report?, Computer Security Institute editorial archive, available online at http://www.gocsi.com/infopro.htm.
7. Mayor, Tracy, Someone to watch over you, *CIO*, March 1, 2001.
8. Devlin, Ed, Business continuity programs, job levels need to change in the wake of Sept. 11 attacks, *Disaster Recovery J.*, Winter 2002.
9. Bernardin, H. John and Russell, Joyce, *Human Resource Management: An Experimental Approach*, 2nd ed., New York: McGraw-Hill, 1998, pp. 73–101.
10. International Information System Security Certification Consortium (ISC)[2], available online at http://www.isc2.org/.
11. Quoted in Rothke, Ben, The professional certification predicament, *Comput. Security J.*, V. XVI, No. 2 (2000), p. 2.
12. Bernardin, H. John and Russell, Joyce, *Human Resource Management: An Experimental Approach*, 2nd ed., New York: McGraw-Hill, 1998, p. 161.
13. Bernardin, H. John and Russell, Joyce, *Human Resource Management: An Experimental Approach*, 2nd ed., New York: McGraw-Hill, 1998, pp. 499–507.
14. Shaw, Eric, Post, Jerrold, and Ruby, Keven, Managing the threat from within, *Inf. Security*, July 2000, p. 70.
15. British Standard 7799/ISO Standard 17799: *Information Security Management,* London: British Standards Institute, 1999, Sections 6.1.1–2.
16. Schlossberg, Barry J. and Sarris, Scott, Beyond the firewall: the enemy within, *Inf. Syst. Security Assoc. Password*, January, 2002.
17. Saita, Anne, The enemy within, *Inf. Security*, June 2001, p. 20.
18. Walgenbach, Paul H., Dittrich, Norman E., and Hanson, Ernest I., *Principles of Accounting*, 3rd ed., New York: Harcourt Brace Jovanovich, 1984, p. 244.
19. Walgenbach, Paul H., Dittrich, Norman E., and Hanson, Ernest I., *Principles of Accounting*, 3rd ed., New York: Harcourt Brace Jovanovich, 1984, p. 260.
20. Horngren, Charles T., *Cost Accounting: A Managerial Emphasis*, 5th ed., Englewood Cliffs, NJ: Prentice Hall, 1982, p. 909.
21. Vallabhaneni, S. Rao, *CISSP Examination Textbooks Vol. 1: Theory,* Schaumburg, IL: SRV Professional Publications, 2000, pp. 142, 311–312.
22. Kabay, M.E., Personnel and security: separation of duties, *Network World Fusion*, available online at http://www.nwfusion.com/newsletters/sec/2000/0612sec2.html.
23. British Standard 7799/ISO Standard 17799: *Information Security Management,* London: British Standards Institute, 1999, Section 8.1.5.
24. Russell, Deborah and Gangemi, G.T. Sr., *Computer Security Basics,* Sebastopol, CA: O'Reilly, 1991, pp. 100–101.
25. Horngren, Charles T., *Cost Accounting: A Managerial Emphasis*, 5th ed., Englewood Cliffs, NJ: Prentice Hall, 1982, p. 914.
26. Kabay, M.E., Personnel and security: separation of duties, *Network World Fusion*, available online at http://www.nwfusion.com/newsletters/sec/2000/0612sec2.html.
27. Garfinkel, Simson and Spafford, Gene, *Practical UNIX and Internet Security*, Sebastopol, CA: O'Reilly, 1996, p. 393.
28. Wood, Charles Cresson, Top 10 information security policies to help protect your organization against cyber-terrorism, p. 3, available online at http://www.pentasafe.com/.

74

Ownership and Custody of Data

William Hugh Murray, CISSP

This chapter introduces and defines the concepts of data owner and custodian; their origins and their emergence; and the rights, duties, privileges, and responsibilities of each. It describes how to identify the data and the owner and to map one to the other. It discusses the language and the tools that the owner uses to communicate his intention to the custodian and the user. Finally, it makes recommendations about how to employ these concepts within your organization.

Introduction and Background

For a number of years now we have been using the roles of data owner and custodian to assist us in managing the security of our data. These concepts were implicit in the way the enterprise acted, but we have only recently made them sufficiently explicit that we can talk about them. We use the words routinely as though there is general agreement on what we mean by them. However, there is relatively little discussion of them in the literature.

In the early days of mainframe access control, we simply assumed that we knew who was supposed to access the data. In military mandatory access control systems, the assumption was that data was classified and users were cleared. If the clearance of the user dominated the classification of the user, then access was allowed. There was the troublesome concept of need-to-know; but for the life of me, I cannot remember how we intended to deal with it. I assume that we intended to deal with it in agreement with the paper analogy. There would have been an access control matrix, but it was viewed as stable. It could be created and maintained by some omniscient privileged user, but no one seemed to give much thought to the source of his knowledge. (I recall being told about an A-level system where access could not be changed while the system was operational. This was not considered to be a problem because the system routinely failed about once a week. Rights were changed while it was offline.)

In time-sharing systems, access was similarly obvious. Most data was accessed and used only by its author and creator. Such sharing of his data as occurred was authorized in a manner similar to that in modern UNIX. That is, the creator granted privileges to the file system object to members of his own affinity group or to the world. While this is not sufficiently granular for today's large group sizes and populations, it was adequate at the time.

ACF2, the first access control for MVS, was developed in a university setting by systems programmers and for systems programmers. It was rules-based. The default rule was that a user could access data that he created. To facilitate this, the creator's name was forced as the high-level qualifier of the object name. Sharing was based upon the rules database. As with the access control matrix, creation and maintenance of this database required both privilege and omniscience. In practice, the privilege was assigned to a systems programmer. It was simply assumed that all systems programmers were omniscient and trustworthy; they were trusted by necessity. Over time, the creation and maintenance of the ACF2 rules migrated to the security staff. While I am sure that we

had begun to talk about ownership by that time, none of these systems included any concept of or abstraction for an object owner.

In reviewing my papers, the first explicit discussion of ownership that I find is in 1981; but by that time it was a fairly mature concept. It must have been a fairly intuitive concept to emerge whole without much previous discussion in the literature.

What is clear is that we must have someone with the authority to control access to data and to make the difficult decisions about how it is to be used and protected. We call this person the *author*. It is less obvious, but no less true, that the person who makes that decision needs to understand the sensitivity of the data. The more granular and specific that knowledge, the better the decision will be.

My recollection is that the first important system to externalize the abstraction of owner was RACF. (One of the nice things about having lived to this age is that the memories of your contemporaries are not good enough for them to challenge you.) RACF access control is list-based. The list is organized by resource. That is, there is a row for each object. The row contains the names of any users or defined and named groups of users with access to that resource and the type of access (e.g., create, read, write, delete) that they have. Each object has an owner and the name of that owner is explicit in the row. The owner might be a user or a group, that is, a business function or other affinity group. The owner has the implicit right to grant access or to add users or groups to the entry. For the first time we had a system that externalized the privilege to create and maintain the access control rules in a formal, granular, and independent manner.

Definitions

> **Owner,** n. One who owns; a rightful proprietor; one who has the legal or rightful title, whether he is the possessor or not.
>
> **— *Webster's Dictionary*, 1913**
>
> **Owner,** n. Principal or agent who exercises the exclusive right to use.
>
> **Owner,** n. The individual manager or representative of management who is responsible for making and communicating judgments and decisions on behalf of the organization with regard to the use, identification, classification, and protection of a specific information asset.
>
> **— Handbook of Information Security Management**
>
> **Zella G. Ruthberg and Harold F. Tipton, Editors, 1993**
>
> **Ownership,** n. The state of being an owner; the right to own; exclusive right of possession; legal or just claim or title; proprietorship.
>
> **Ownership,** n. The exclusive right to use.
>
> **Custodian,** n. One that guards and protects or maintains; especially: one entrusted with guarding and keeping property or records or with custody or guardianship of prisoners or inmates.
>
> **— Merriam-Webster's Collegiate Dictionary**
>
> **Custodian.** A designated person who has authorized possession of information and is entrusted to provide proper protection, maintenance, and usage control of the information in an operational environment.
>
> **— Handbook of Information Security Management**
>
> **Zella G. Ruthberg and Harold F. Tipton, Editors, 1993**

Policy

It is a matter of policy that management makes statements about the level of risk that it is prepared to take and whom it intends to hold accountable for protection. Owners and custodians are useful abstractions for assigning and distinguishing this responsibility for protection. Policy should require that owners be explicitly identified; that is, that the responsibility for protection be explicitly identified. While ownership is implicit, in

the absence of requiring that it be made explicit, the responsibility for the protection of information is often overlooked. Similarly, policy should make it explicit that custodians of data must protect it in accordance with the directions of the owner.

Roles and Responsibilities

Owner

At one level, the owner of institutional data is the institution itself. However, it is a fundamental characteristic of organizations that they assign their privileges and capabilities to individual members of the organization. When we speak of owner, we refer to that member of the organization to whom the organization has assigned the responsibility for a particular asset. (To avoid any possible confusion about the real versus the virtual owner of the data, many organizations eschew the use of *owner* in favor of some other word such as agent, steward, or surrogate. For our purposes, the owner is the assigned agent.)

This individual exercises all of the organization's rights and interests in the data. These include:

- Judging the asset's importance, value, and sensitivity
- Deciding how and by whom the asset may be used
- Specifying the business controls
- Specifying the protection requirements for the asset
- Communicating decisions to others (e.g., labeling the object with its classification)
- Acquiring and operating necessary automated controls over the assets
- Monitoring compliance and initiating corrective action

Note that these duties are not normally separable. That is to say that all must be assigned to the same agent. Specifically, the right to use cannot be separated from the responsibility to protect.

We should keep in mind that others might have some interest in an information asset. For example, while the institution may own a copy of information such as employee name and address in the pay record, the employee still has a proprietary interest in the data. While this interest may not rise to the level of ownership, it is still a material interest. For example, the employee has an interest in the accuracy and confidentiality of the data. In exercising its interest, the institution and its agents must honor these other interests.

Custodian

Even the dictionary definition recognizes that the idea of custodian includes one who is responsible for protecting records. This responsibility includes:

- Protecting the data in accordance with owner direction or agreement with the owner
- Exercising sound business judgment in the protection of data
- Reporting to the data owner on the discharge of his responsibilities

Suppliers of data processing services and managers of computers and storage devices are typically custodians of application data and software processed or stored on their systems. This may include paper input documents and printed reports.

Because it is these custodians who choose, acquire, and operate the computers and storage, they must provide the necessary access controls. The controls chosen must, at a minimum, meet the requirements specified by the owners. Better yet, they should meet the real requirements of the application, regardless of whether the owner of the data is able to recognize and articulate those requirements. Requirements to which the controls must answer include reliability, granularity, ease of use, responsiveness, and others.

Administrator

The owner may wish to delegate the actual operation of the access controls to a surrogate. This will be particularly true when the amount of special knowledge required to operate the controls exceeds the amount required to make the decisions about the use of the data.

Such an administrator is responsible for faithfully carrying out the intent of the owner. He should act in such a way that he can demonstrate that all of his actions were authorized by the responsible owner and that he acted on all such authorizations. This includes keeping records of what he did and the authorizations on which he acted.

User Manager

The duties of user management include:

- Enrolling users and vouching for their identities
- Instructing them in the use and protection of assets
- Supervising their use of assets
- Noting variances and taking corrective action

While the list of responsibilities is short, the role of user management may be the most important in the enterprise. This is because user management is closer to the use of the resources than any other managers.

User

Users are responsible for:

- Using the enterprise information and information processing resources only for authorized and intended purposes
- Effective use and operation of controls (e.g., choice of passwords)
- Performance of applicable owner and custodian duties
- Compliance with directions of owners and management
- Reporting all variances to owners, managers, and staff

Variances should be reported to at least two people. This reduces the probability that the variance is called to the attention of only the individual causing it. The owner of the resource and the manager of the user would be likely candidates for notification. Otherwise, use one line manager and one staff manager (e.g., audit or security staff).

Identifying the Information

Identifying the data to be protected might seem to be a trivial exercise. Indeed, before computers, it really was. The enterprise focused on major and persistent documents and on major functional files such as those of payroll records or payables. Focus was placed on those files that were special to the industry or enterprise. In banking, one worried about the records of deposits and loans; in insurance, one worried about policy master records. Managers focused on departmental records and used file cabinets as the objects of control and protection. Even when computers emerged, one might still have facused on the paper printout of the data rather than on the record on magnetic tape. When a megabyte was the size of a refrigerator, one identified it and protected its contents similarly to how one protected the contents of a file cabinet. As magnetic storage became sufficiently dense that the storage object was shared across a large number of data objects, we started to identify data sets. While we often think of a data set as analogous to the modern file, in fact it was a collection of logically related files that shared a name. The input file to a job, the output file from the job, and the archival version of that file might all be part of the same logical data set. The members of a data set were related in a formal way. While there are a small number of different types of data sets (e.g., partitioned, sequential, VSAM), members of all data sets within a type were related in a similar way. The information about the relationships was recorded in the metadata for the data set.

Therefore, for protection purposes, one made decisions about the named data set rather than about the physical objects that made them up. The number of data sets was sufficiently small that identifying them all was not difficult.

In modern systems, the data objects of interest are organized into (tree-structured) directories and files. A data set in a mainframe might correspond to a file or to all the files in a directory. However, the relationship between a directory and the files and other directories that are stored in it may be totally arbitrary. There are

conventions, but there are no fixed rules that can be consistently used to reduce the number of objects over which one must make decisions. For example, in one directory, programs and data may be stored together; while in the next one, programs and data may be stored in separate named subdirectories. A file name may be qualified by the name of the directory in which it is stored — and then again, it may not.

Therefore, for protection purposes, a decision may have to be made over every directory entry and possibly every file. The number of objects expands, perhaps even faster than the quantity of data. This is complicated further by the rapidly falling cost of storage. Cheap storage enables one to keep data longer and otherwise encourages growth in the number of data objects.

Data sets also had the advantage that the names tended to be unique within a system and, often, by convention, across an enterprise. In modern practice, neither objects nor names are unique even within a system, much less across an enterprise.

In modern systems, there is no single reference or handle that one can use to identify all data within an enterprise. However, most of them require some enterprise procedures or conventions. For example, one can store data according to its kind and, by inference, its importance.

- Enterprise data versus departmental, personal, or other
- Changeable versus fixed (e.g., balances versus transactions; programs versus data; drafts versus published documents; images versus text)
- Documents versus other
- Permanent versus temporary
- Business functional applications versus other (e.g., payroll, payables, sales) versus other (e.g., correspondence)
- Active versus archival
- Other enterprise-specific categories

Each of these distinctions can be useful. Different procedures may be required for each.

Identifying the Owner

Prior to the use of the computer, management did not explicitly identify the owners of information. This was, in part, because the information of interest was the functional data of the organization. This information included pay records, customer records, sales records, etc. Ownership and custody of the information were almost always in the same hands. When the computer came along, it separated custody from ownership. The computer function found itself with custody of the information. Management did not even mind very much until decisions needed to be made about the care of the records.

Management was particularly uncomfortable with decisions about access and security. They suddenly realized that one standard of care was not appropriate for all data and that they did not know enough about the data to feel comfortable making all the decisions. Everyone wanted discretion over the data but no one wanted responsibility. It was obvious that mistakes were going to be made. Often, by the time anyone recognized there was a problem, it was already a serious problem and resolving it was difficult.

By this time, there was often so much data that discovering its owner was difficult. There were few volunteers. It was not unusual for the custodians to threaten to destroy the data if the owner did not step forward and take responsibility.

Line Manager

One useful way to assign ownership is to say that line managers are responsible for all of the resources allocated to them to accomplish their missions. This rule includes the responsibility to identify all of those assets. This ensures that the manager cannot escape responsibility for an asset by saying that he did not know.

Business Function Manager

Although this is where the problem got out of hand, it is the easiest to solve. It is not difficult to get the managers of payroll or payables to accept the fact that they own their data. It is usually sufficient to simply

raise the question. When we finally got around to doing it, it was not much more difficult than going down the list of information assets.

Author

Another useful way to assign ownership is to say that the author or creator of a data object is its owner until and unless it is reassigned. This rule is particularly useful in modern systems where much of the data in the computer is created without explicit management direction and where many employees have discretion to create it. Like the first rule, it works by default. This is the rule that covers most of the data created and stored on the desktop.

Surrogate Owners

Even with functional data, problems still arise with shared data, as for example in modern normalized databases. One may go to great pains to eliminate redundant data and the inevitable inconsistencies, not to say inaccuracies, that go with it. The organization of the database is intended to reflect the relationships of the entities described rather than the organization of the owners or even the users. This may make mapping the data to its owners difficult.

An example is a customer master record that is shared by three or four different business functions. If one of the functions assumes ownership, the data may be operated for their benefit at the expense of the others. If it is not well managed, the other functions may start keeping their own copies with a loss of both accuracy and efficiency.

One solution to this problem is to create a surrogate function to act as the owner of the data. This surrogate acts as agent for his principals; he satisfies their ownership requirements while exercising their discretion. He is motivated to satisfy all of his customers equally. When conflicts arise between the requirements of one customer and another, he negotiates and resolves them.

In modern systems, shared functional data is usually stored in databases rather than in flat files. Such systems permit more granular control and more choices about the assignment of ownership. Control is no longer limited by the physical organization of the data and storage.

Classification and Labeling

One way for the owner to communicate his intentions about how to treat the information is to write instructions as metadata on the data object. A classification scheme provides an efficient language in which to write those instructions. The name of the class is both an assertion about the sensitivity of the data and the name of the set of protective measures to be used to protect it. The owner puts the label on the data object, and the custodian uses the associated protective measures.

The number of classes must be must be small enough for one to be able to habitually remember the association between the name of the class and the related controls. It must be large enough to ensure that all data receives the appropriate protection, while expensive measures are reserved to the data that really requires them.

We should prefer policies that enable us to detect objects that are not properly classified or labeled. Policies that require that all objects be labeled, even the least sensitive, make it easy to recognize omissions. Many organizations do not require that public data be labeled as such. This makes it difficult to distinguish between public data and data over which no decision has been made.

While paper feels natural and comfortable to us, it has severe limitations not shared by more modern media. It is bulky, friable, flammable, resistant to timely update, and expensive to copy or back up. On the other hand, it has an interesting kind of integrity; it is both tamper-resistant and tamper-evident. In paper systems, the label is immutably bound to the object and travels with it, but the controls are all manual. In automated systems, the label is no more reliable than the system and does not travel with the object beyond the system. However, controls can be based upon the label and automatically invoked. In mandatory access control systems, both the label and the controls are reliable. In discretionary access control systems, both the labels and the controls are less reliable but adequate for many applications and environments.

Cryptographic systems can be used to bind the label to the object so that the label follows the object in such a way that the object can only be opened in environments that can be relied upon to enforce the label and the associated controls. Certain high-integrity imaging systems (e.g., Adobe Acrobat) can bind the label in such way that the object cannot be displayed or printed without the label.

Access Control

The owner uses access controls to automatically direct and restrain who sees or modifies the data. Mandatory access controls ensure consistent application of management's policy across an entire system while minimizing the amount of administrative activity necessary to achieve it. Discretionary controls enable owners to implement their intent in a flexible way. However, consistent enforcement of policy may require more management attention and administrative activity.

Variance Detection and Control

It must be possible for the owner to observe and measure how custodians and others comply with his instructions. He must have visibility. This visibility may be provided in part by alarms, messages, confirmations, and reports. It may be provided in part by feedback from such staffs as operations, security administration, and audit.

The owner is interested in the reliability of the user identification and authentication (I&A) scheme. He is most likely to look to the audit report for this. Auditors should look at the fundamental strength of the I&A mechanism, log-on variances, the security of password change procedures where used, and weak passwords where these are possible.

The owner is also likely to look to the audit report for information on the integrity of the access control system and the authorization scheme. The auditors will wish to look to the suitability of the controls to the applications and environment. Are they application-specific or provided by the system? Are the controls appropriately granular and responsive to the owner? They will be interested in whether the controls are mandatory or discretionary, rules-based or list-based. They will wish to know whether the controls have been subjected to third-party evaluation, how they are installed and operated, and how they are protected from late change or other interference. They will want to know the number of privileged users of the system and how they are supervised.

Periodically, the owner may want to compare the access control rules to what he thinks he authorized. The frequency of this reconciliation will be a function of the number of rules and the amount of change.

The owner will be interested in denied attempts to access his data; repeated attempts should result in alarms. Some number of denied attempts are probably intended to be authorized and will result in corrections to the rules. Others may require follow-up with the user. The user will want to be able to detect all accesses to the data that he owns so that he can compare actual access to what he thinks he authorized. This information may be in logs or reports from logs.

Recommendations

- Policy should provide that ownership of all assets should be explicitly assigned. This helps to avoid errors of omission.
- Ownership of all records or data objects should be assigned to an appropriate level of granularity. In general, this means that there will be an owner for each document, file, folder, or directory, but not necessarily for each record or message.
- The name of the owner should be included in the metadata for the object.
- The classification or other reference to the protective measures should be included in the metadata for the object.
- Because few modern systems provide abstractions or controls for data classification or owner, this metadata should be stored in the object name or in the object itself.
- The owner should have responsive control over access. This can be through automated controls, administrators, or other surrogates.

- There should be a clear agreement between the owner and the custodian as to how the data will be protected. Where a classification and labeling system exists, this can be the basis of sensitivity labels on the object.
- Consider written agreements between owners and custodians that describe the protective measures to be used. As a rule, these agreements should be based upon offers made by the custodians.
- The owner should have adequate visibility into the operation and effectiveness of the controls.
- There should be prompt variance detection and corrective action.

Conclusion

The ideas of ownership and custody are fundamental to any information protection scheme. They enable management to fix responsibility and accountability for deciding how an object is to be protected and for protecting it in accordance with that decision. They are essential for avoiding errors of omission. They are essential for efficiency; that is, for ensuring that all data is appropriately protected while reserving expensive measures only for the data that requires them.

While management must be cautious in assigning the discretion to use and the responsibility to protect so as not to give away its own rights in the data, it must be certain that control is assigned with sufficient granularity that decisions can be made and control exercised. While identifying the proper owner and ensuring that responsibility for all data is properly assigned are difficult, both are essential to accountability.

Owners should measure custodians on their compliance, and management should measure owners on effectiveness and efficiency.

References

1. *Webster's Dictionary*, 1913.
2. *Handbook of Information Security Management*; Zella G. Ruthberg and Harold F. Tipton (Eds.), Auerbach (Boston): 1993.
3. *Merriam Webster's Collegiate Dictionary.*
4. *Handbook of Information Security Management*; Zella G. Ruthberg and Harold F. Tipton (Eds.), Auerbach (Boston): 1993.

75

Hiring Ex-Criminal Hackers

Ed Skoudis, CISSP

> Making their way, the only way they know how.
>
> That's just a little bit more than the law will allow.
>
> **— Waylon Jennings, "Good Ol' Boys"**
>
> **Theme song from *Dukes of Hazzard***

Suppose someone applies for a system administrator job, or, better yet, an open slot on your computer security team. The applicant is eminently qualified for the position, having wizard-like skills on the exact operating systems deployed throughout your organization. You need his skills, big time. However, the candidate poses a bit of a problem. This otherwise-stellar applicant has a bit of a spotty record with the criminal justice system. By spotty, I mean that your potential hire was found guilty of hacking a Fortune 500 company and stealing some sensitive data. He did the crime, but he has also done the time.

Should you still consider such a person for a position on your security team? Or, should you let bygones be bygones and just move forward? Some companies shy away from such individuals immediately. Others take a "Don't ask… Don't tell" stance. Still others actively embrace such people for their great skills. If your organization hires an ex-criminal hacker, would you be legally responsible if he damages a customer or supplier's computer systems? You could be found guilty of negligent hiring, whereby an employer is liable for taking a hiring risk and exposing customers, suppliers, and other employees to it.

This chapter analyzes the issues associated with hiring ex-criminal hackers so you can think through your own organization's approach to this issue. The chapter looks at both sides of the problem, and then the author states his opinion on the matter, for what it is worth. While the author attempts to evenhandedly argue both sides of this topic, keep in mind that the author does not necessarily agree with all of these arguments. Instead, the concepts raised are those most often advanced by proponents on either side of this divide.

The discussion in this chapter does *not* refer to non-criminal hackers. Remember, as used in the computer underground, the term "hacker" does not by itself imply that the person has done wrong. People who have hacking skills may have acquired them completely lawfully, by studying computer security or conducting legitimate penetration testing against consenting targets, such as their employers or customers. There are many of these "white-hat" hackers in the information technology business. The author himself falls into this white-hat category, as do many others, and would like to think we are very hirable without concerns.

This chapter analyzes the question of whether to hire hackers who have an actual prior criminal conviction, or are known to have been involved in criminal activity but may have not been prosecuted (yet). We refer to them as ex-criminal hackers because they were either busted and did some time in jail or are known to have committed crimes. In other words, we are talking about actual former black hats or deeply gray hats.

Why This Matters

One might wonder if this analysis really matters that much. Actually, it really does (of course I think that… I would not be writing about it if I didn't.) But, think about it. Information technology (IT) carries and stores the lifeblood of most organizations today: information. The people who run this technology have tremendous access to the most sensitive information an organization has: personnel employment and health records, sensitive customer data, legal and regulatory compliance information, comprehensive financial results, and perhaps even launch codes. Just to keep the organization running, the IT department often acts as a high-tech priesthood given wide-open access to the very soul of the business.

If IT has a bad egg as an employee, the damage that can be done to an organization's finances, reputation, and very existence might be devastating. Inside personnel know how to hit an organization where it hurts, undermining technology and processes to maximize not only their own personal gain but also the damage inflicted on their target. Looking at statistics regarding computer crime compiled annually by the Computer Security Institute and the FBI, the number of attacks from insiders and outsiders is virtually the same.[1] However, the cost of damages from computer attacks commonly perpetrated by insiders (insider net abuse, financial fraud, and theft of proprietary information) significantly outweighs the cost of attacks by outsiders. That is because insiders know how to cause trouble for their organizations.

The 2002 CSI/FBI survey also indicated that 65 percent of organizations would not consider hiring reformed hackers as consultants; 17 percent of others would consider it; while the remaining just do not know. In this author's experience, even the 65 percent of those who say they would rule out hiring ex-criminal hackers do not have explicit policies regarding this decision or even very detailed background checks to enforce it. Therefore, even among those whose guts tell them not to hire ex-criminal hackers, many unwittingly hire them without understanding their background. Is this wise? Let us explore the case for and against hiring ex-criminal hackers in more detail.

The Case for Hiring Ex-Criminal Hackers

> Yes, I am a criminal. My crime is that of curiosity. My crime is that of judging people by what they say and think, not by what they look like. My crime is that of outsmarting you, something that you will never forgive me for. I am a hacker, and this is my manifesto. You may stop this individual, but you can't stop us all…
>
> **— From the *Hacker Manifesto,* written by "The Mentor" in the mid-1980s**

This creed by "The Mentor" is still very relevant today, as it highlights many of the issues associated with hackers and the computer underground, including whether organizations should employ ex-criminal hackers. We analyze some of the issues brought up in the *Hacker Manifesto,* as well as related topics. The arguments for hiring ex-criminal hackers fall into three general categories: questions associated with who is really to blame, doubt about how dangerous computer attackers really are, and society's need for exceptional technical talent.

It's Not Really Their Fault…

> I went to the lost and found department at my local shopping mall. I told the kid behind the counter that I'd lost my youthful exuberance. He said they'd call me if it turned up.
>
> **— William J. Basile, my college roommate**

One of the primary arguments for hiring ex-criminal attackers involves looking at whether we can really assign blame; and, if we do, who is really at fault. First, consider the focus of our criminal justice system — reform. By definition, a penitentiary is where someone repents for past crimes, and is reformed to become a contributing member of society. After their release, they have done their time, and should be able to contribute fully to society. Harshly turning down such people from employment may doom them to perpetuate their life of crime. For crime in general, the recidivism rate is far lower when someone returns to society as a productive member

of the workforce, especially for young people.[2] Turning the other cheek, as it were, may help them have a positive impact on society. They have paid for their past sins, and it is time for forgiveness. Who is a potential employer to judge when the criminal justice system has already not only judged, but punished?

Furthermore, young people commit many computer crimes in their high-school or college years. Such perpetrators are not hardened criminals; they are merely satisfying their youthful wanderlust by exploring computer systems. As with many young people, they are merely pushing the boundaries of their environment to understand how the world works. If they do not really cause much damage, can we really damn them for simply discovering vulnerabilities and pushing the boundaries of human knowledge? Is that not what being young is all about? This line of argument fills the pages of the always-interesting and often-provocative *2600* magazine.[3] This self-titled, "Hacker Quarterly" magazine is published every three months and can be found in most major bookstores' magazine rack. In addition to some technical content describing attacks, the magazine also actively promotes the culture of disaffected youth exploring computers for fun and learning. According to this mind-set, these noble adventurers are not setting out to do damage, and are simply misunderstood by a society either too evil or too stupid to understand the subtleties of the computer underground.

Also, our overall society seems to encourage adventuresome computer hacking. Consider recent movies like *The Matrix* from 1999, or its 2003 sequels. In those movies, a corrupt culture tries to stifle an innocent computer hacker who may expose its ultimate lie. In another classic hacker movie, 1983's *WarGames,* a hacker is the ultimate hero, saving the world from a nuclear holocaust (which, of course, he accidentally triggered in the first place). In these and many other examples, the hackers are the good guys, trying to save the world from corruption. Does it make sense to limit job opportunities to such people simply because they have followed the lead given by our mass entertainment culture?

Are They Really That Dangerous, or Do They Help?

> Want to play a game?
>
> **— WOPR, the computer from the movie *WarGames***

A second and related argument associated with hiring ex-criminal computer attackers involves a consideration of the real damage done in a large number of computer attacks. According to this argument, a computer attack involves minimal real-world damage, with an attacker just exploring a network and copying some files. No lives are in jeopardy, and usually, minimal real-world losses are incurred. However, the attacker may find himself in jail simply because his case was novel and his target especially juicy. For cases with little or no real-world damage, computer attackers should be given another chance at using their skills for good.

Also, numerous people in the computer security industry have gotten started by youthful exploration of computer systems with little harm to society as a whole. Some of the most skilled computer security personnel today cut their teeth by surreptitiously breaking into other people's computers. Sure, goes this argument, now that we are all grown up, we recognize the errors of our youth. If we put everyone in jail who learned computer security by breaking into systems, we just may decimate the computer security industry. Furthermore, some of the folks who encourage tough penalties for computer crime are, in fact, hypocrites, given their own shady pasts. While such people may criticize those who were unlucky enough to get caught, they themselves were just as guilty of computer attacks when they were youngsters.

This argument is bolstered by the wonderful contributions of some high-profile individuals who have bent or even broken the law in computer and related attacks in the past. For example, consider Steve Jobs, the celebrated founder and current CEO of Apple Computer, Inc. Back in college, Jobs entered the hardware business not by selling candy-colored, easy-to-use computer systems. Instead, he made money the old-fashioned way (at least for the 1970s): he sold blue-box hardware that generated specific tones allowing users to explore or defraud the public telephone system. Although Jobs was never charged with a computer-related crime, clearly his exploits were not in the best interests of the telephone company. Yet, looking at the sum total of his activities in the computer field, Jobs has greatly improved the computer industry, helping to introduce the personal computer and then the graphical user interface to the masses, birthing Apple Computer, and then saving Apple from near extinction.

Another example involves Kevin Poulsen, one of the best journalists in the computer security industry today. Poulsen once served significant jail time for some elaborate attacks against a large California-based telephone company.[4] But that is his past; he is now helping advance the cause of computer security as the chief editor of

the online security news and editorial section of SecurityFocus.com. Poulsen's past is checkered; his current stuff is extremely helpful in understanding how to and why we should secure our systems.

This argument extends to numerous other individuals. Much of the computer and Internet industries was built by people who push the limits of both technology and the law. These concerns point to the often-blurred line between computer professionals and computer attackers, the indistinct separation of white hats and black hats into a gray goo. Let's face it, if Jobs, Poulsen, and others built their technical and business savvy, as well as our overall networked world, by illegally tapping into computers and helping others to do so, today's computer attackers may be tomorrow's computer security professionals, professors, CEOs, or even presidents. I can just picture the bumper stickers now — Kevin for President.[1] Watch out!

But We Need Them…

Another area for consideration on the issue of hiring ex-criminal attackers involves our society's need for technically sophisticated personnel. Although a recent recession has furloughed many IT professionals, people with very strong security skills remain in high demand. Looking at the vast numbers of gaping holes in corporate networks and major software packages, it is clear that businesses just cannot get enough good security people to shore up their networks against attack. Putting our best and brightest in prison and never hiring them after they have been reformed is a waste of some very valuable human capital. Given the great contribution these folks can make, as compared with the costs of keeping someone on public assistance or in prison, society as a whole benefits from having ex-criminal hackers gainfully employed.

Focusing on the computer security industry, ex-criminal hackers understand computer attacks far better than anyone else does. They truly know the hacker mind-set. While they may or may not have the best skills in conducting overall computer security architecture, such people are among the best in doing detailed penetration tests. For such testing, one needs to think like an attacker and employ the skills and mind-set associated with deep, focused analysis on ripping apart networks, operating systems, and applications. The best penetration tests are done by those who not only consider today's known vulnerabilities, but also look deeper for new holes and exploits. Sometimes, ex-criminal hackers are the absolute best at doing this.

In fact, many of the major vulnerabilities discovered today are found by those labeled "gray hats," people who may be in trouble with the law but continue to do computer research. If one looks beyond some of their bravado and unusual culture, these people may actually be helping the information security industry do research, understand problems, and fix vulnerabilities before the serious bad guys do. Our underlying technology is so severely feeble from a security perspective that finding and pointing out these vulnerabilities is really valuable. On a daily basis, major vulnerabilities are discovered in systems of all types: desktops, servers, personal digital assistants, routers…you name it. If it has software in it, chances are that someone has found security flaws in it, and quite often that person is a reformed, ex-criminal hacker.

Gobbles, a group of security researchers, found some major security vulnerabilities, including a significant flaw allowing complete remote compromise of Apache Web servers in mid-2002. Their brash style, together with their penchant for full disclosure including the release of easy-to-use exploitation code, have rubbed many in the computer security industry the wrong way. However, would you rather have Gobbles discover and publish such findings, or a major terrorist group or foreign country's cyber-warfare troops exploit such holes in a massive attack against the world's infrastructure? Clearly, full disclosure from Gobbles is the better (although perhaps not the best) alternative, as it allows us to fix our problems. Despite Gobbles' strong gray-hat status, they have helped improve Apache's security.

Similarly, Adrian Lamo has broken into and explored the sensitive inner networks of *The New York Times,* Yahoo, and WorldCom. Although he has publicly admitted that such adventures may run afoul of the law, Lamo points out how he has helped these companies secure themselves. Lamo's "victims" have expressed gratitude for his open attitude of sharing information about his exploits with these companies before going public. By discovering flaws in our systems, Lamo, Gobbles, and many others run up against the law and in some cases explicitly violate it. However, in doing so, they ultimately improve the state of computer security by making us focus on computer problems.

Consider a biological analogy that sheds some light on this whole issue. According to recent research, if children are not exposed to any common colds while they are under age ten, their immune systems are in fact weaker as they grow up. As youngsters, they have not built up strength and immunities. In a similar way, computer attackers represent colds periodically impacting the computer industry. Just like colds, they build our defenses by making us harden our systems and deploy patches. That way, we will be much better off when

a really serious computer attack occurs. For example, when the Code Red worm spread rapidly in July 2001, it was not only a nuisance. In fact, it made many of us patch our systems and revisit our computer incident handling capabilities. In a counter-intuitive way, some computer attackers help improve computer security by actually attacking our systems in violation of the law.

On top of that argument, we also have to consider what happens if we do not employ the ex-criminal attackers in helping improve computer security. We may very well miss some big vulnerabilities. If ex-criminal hackers cannot use their skills for good, they will use them for evil. By hiring such individuals, the computer industry can keep some of our best and brightest people focused on improving computer security, rather than unraveling the network and systems from underneath us. If these people are gainfully employed in the computer business when they discover vulnerabilities, they will be more likely to share their findings in a responsible way, disclosing it to the appropriate vendors and helping to seek a positive solution for the problem.

One analogy for this situation involves the dilemma over Russian nuclear scientists. After the end of the Cold War, these brilliant researchers were no longer needed to design and build bombs for the now-defunct Soviet Union. Many people fear that, with hard economic times in Russia and a skill set that cannot be readily applied to other jobs, these scientists may help rogue states or terrorists fulfill their nuclear attack fantasies. Because of this concern, the international community has set up programs to employ such scientists in managing and even safely destroying nuclear stockpiles. In a similar way, if we do not utilize our ex-criminal hackers, the criminal underground may hire them to conduct seriously nasty attacks. A computer attacker who has served jail time may have made contact with non-computer criminals while in prison. If the ex-criminal hackers cannot find a means to support themselves using their computer skills because they are blackballed from employment, they may turn to their "friends" from prison for funding. Nastier computer attacks result. By hiring such individuals and directing them toward good, we help to alleviate this sort of problem.

The Case against Hiring Ex-Criminal Hackers

As one might guess, not everyone agrees with the line of arguments above (now, there is an understatement!). So, how do critics respond? Let us take a look at their critiques, lining them up in the same order as the arguments presented above.

But It Really Is Their Fault

Ex-criminal hackers have already demonstrated that they cannot be trusted with access to computer systems. Many of them have been judged in a criminal justice system with safeguards to protect the innocent. "Sure," goes the argument from many organizations, "we believe in reforming criminals and forgiveness in general, but it's not *our* organization's job to spread forgiveness and improve the world by putting ourselves at risk." Most organizations are in business to either make a profit or deliver services to a constituency. Management and employees of these organizations have a fiduciary responsibility to protect customers, employees, and shareholders from unnecessary risks. Hiring ex-criminals into an information technology department and giving them access to a network with sensitive data to help make the world a better place is not a palatable trade-off for most organizations.

Not hiring ex-criminal hackers can also have a deterring effect. Especially in cases of computer crime involving young people, strong penalties will discourage them from turning to a life of crime. Indeed, right now, some elements of the computer underground perversely joke that if they do ever get busted, they will do their time in jail and become highly paid security consultants after they get out. As a society, it is just not right to reward malfeasance with the promise of six-figure salaries after a year or so in prison. By reversing this logic and making sure that committing computer crime means that you seriously damage your career in technology, we can dampen young people's interest in computer crime. Instead, they may turn their skills to responsible and beneficial computer research, rather than breaking into systems.

Additionally, the idea that it is really not a criminal's fault because *The Matrix* and *WarGames* glorify hacking just shifts blame from the legitimate perpetrator. There are numerous movies that glorify lewd behavior or even mass murder, but we do not decriminalize these activities. Even Robin Hood preached stealing from the rich and giving to the poor, yet we still criminalize theft. We simply do not rely on Hollywood to define our hiring practices, let alone our criminal justice penalties.

They Really Can Be Dangerous!

> Let's play Global Thermonuclear War.
>
> **— David Lightman,**
>
> **Matthew Broderick's character in the movie *WarGames***

Although it may be true that some computer security personnel and other technology industry luminaries skirted the law over the past three decades, this fact does not exonerate the current generation of computer criminals. In the 1970s, 1980s, and early 1990s, computers in general and the Internet in particular were far less important to the functioning of our society. The Robert Tappan Morris, Jr. Worm took down major components of the early Internet in November 1988. Yes, this story did make the evening news back then, but it resulted in little real damage. Today, with information technology permeating our financial, healthcare, and government systems, even a less virulent attack could cause many orders of magnitude more damage, disabling the Internet, causing vital systems to crash, and possibly damaging life and limb. Self-replicating worms, distributed denial-of-service, and highly automated computer attack tools can be very dangerous. With such technologies, the criminally minded hacker could wreak havoc purposely or even accidentally.

Sadly, the playful hacking of yesteryear is truly obsolete, now that our world is incredibly dependent on computers. It is not cute anymore. It is time for people to act responsibly with computer technology.

But We Do Not Need Them That Badly!

Let us now turn our attention to the argument about hiring ex-criminals because we really need their technical skills. True, our society really needs people with strong computer skills. However, we need employees with the *proper* skill set and attitude. For the vast majority of IT occupations, the skills needed to break into a computer system are not the same skills needed to defend a system from attack. Consider a system administrator, whose job it is to provide care and feeding to dozens of workstation and server systems. This job title is probably one of the most common roles in your IT organization that has daily access to very sensitive data. A good system administrator needs the following skills:

- Knowledge of how to keep machines up and running
- Insight into how the operating system functions at a fairly detailed level, including networking, a variety of services, and user-level applications
- Problem-solving proficiency to troubleshoot difficulties
- The ability to document and follow detailed processes, such as system configuration guides and back-up/restoration procedures
- Talent for writing simple scripts to automate tasks needed to keep the system running
- Understanding of how to configure systems securely, hardening them against attacks
- The ability to apply and test patches distributed by a vendor
- An aptitude for recognizing suspicious activities and reporting them to an incident handling team

Many ex-criminal hackers do not really have these basic system administration skills. As a general rule, in both the real world and information technology, it is much easier to break things than to build them up and maintain them. Some attackers can construct elaborate methods for absolutely ripping apart a system without breaking a sweat, but have not mastered the most basic ideas of how to keep the system running. Sure, some attackers may be able to write the code for a mutating kernel module to stealthily conquer a machine, but can they troubleshoot a flaky network connection while keeping hundreds of users happy? For many of these people, the answer is an emphatic, "No!" because their skills and attitudes do not match the job requirements.

I received strong confirmation of this point at the DefCon conference in August 2002. This annual hacker fest, held in Las Vegas, Nevada, includes a highly competitive Capture the Flag competition. In this game, teams of hackers, enthusiasts, and computer professionals are pitted against each other to vie for the highest score in a 28-hour hackathon. You get points by hacking into the other team's system, but lose points if they hack into your machine. Therefore, in the Capture the Flag contest, both offense and defense are critical. The contest starts your adrenaline pumping and remains very intense, as dozens of top-notch hackers from around

the world are hammering your system simultaneously. During the contest this year, a friend of mine reflected the intensity of the sport by shouting expletives. "I know how to hack into these @#$%^ machines," he exclaimed, "but darned if I can stop someone else from getting into my own box!" He had attack skills, but his defense was not up to snuff.

Now, some ex-criminal hackers really do have the skills needed to be superb system administrators, but they also carry a lot of excess and damaging baggage with them. Although employers want these skills, they emphatically do not want system administrators who know how to rip apart systems. Most organizations do not want to hire system administrators, no matter how good they are, who can code elaborate hacks if they have demonstrated in the past that they have used their skills illegally. These organizations would rather have someone who may be less gifted technically, but can do a solid job without jeopardizing the organization.

Beyond system administrators, there are some jobs that really do require computer attack skills, in particular, ethical hacking. Ethical hackers penetrate systems on behalf of the systems' owner to find holes before malicious attackers do. With knowledge of the vulnerabilities, the organizations can deploy defenses based on what the ethical hackers discover. As organizations get more serious about measuring their true security stance, the ethical hacking business continues to grow, employing thousands of very talented security personnel throughout the world. To be effective, these people need skills for breaking into computers. However, the very nature of ethical hacking jobs, with their deep access into very sensitive computer systems, necessitates very careful hiring practices for these roles. Ex-criminal hackers in such positions could be extremely dangerous. They have already demonstrated the illegal use of their skills and could use a role as an "ethical" hacker to simply commit more crimes.

Let us look at the argument that criminal attackers actually make us more secure by pointing out our weaknesses before serious bad guys do major harm. Ethical hackers can serve this same function, provided that organizations actually establish an ethical hacking function. As the computer industry sorts out the liability issues associated with unsecure software and computer attacks, ethical hacking very well may become even more commonplace than it is today. Increasingly, with companies striving to limit their liability and manage risk, ethical hacking will help to measure and enforce a standard minimal set of security practices.

Sorting It All Out

So, both sides of this argument are emphatic about the logic of their respective positions. What should we make of these arguments, and should your company consider hiring ex-criminal hackers? In this author's opinion, most organizations today should avoid hiring ex-criminal attackers. Because most IT positions do involve some level of very sensitive access, you should carefully screen your potential hires to understand any computer crime activities in their past.

However, there are a small number of job roles where computer attack skills actually come in handy: vulnerability research and reporting. Vulnerability researchers do not attack particular companies' computer systems. Instead, they look for holes in computer systems in a laboratory environment, without sensitive real-world data. Their job involves finding security problems so that vendors can fix their systems, and ethical hackers can test for these holes. Universities, software vendors, governments, security consultants, and more hard-core technical publications employ such people to find vulnerabilities and figure out fixes to the problems they discover. Here is one area where ex-criminal hackers can actually make some significant contributions. Using the analogy of the out-of-work Russian nuclear scientists who get employment helping secure or destroy warhead stockpiles, our society can actually use ex-criminal hackers for vulnerability research and reporting.

However, such employment does bring risks. These ex-criminal hackers who are now doing research have to be carefully monitored to make sure their skills are being used for good. You certainly do not want to pay people to find vulnerabilities, and have them share them with criminals, foreign adversaries, or terrorists, all the while hiding the results of their research from their employer. A careful mentor program, as described below, can really help to make sure the ex-criminal hacker's skills are being used for good purposes.

Beware! Recruiting Legal Issues Need HR Support

Before finalizing the decision of whether you would want to hire ex-criminal hackers, let us discuss some important limitations you may face in finding out where your job applicants fit on the black-hat/white-hat spectrum. When interviewing and making hiring decisions, you must keep in mind any restrictions imposed

by your own Human Resources organization, as well as employment laws and regulations. The U.S. Equal Employment Opportunity Commission (EEOC) does not have any explicit restrictions regarding whether or not to hire ex-convicts. However, the EEOC has determined that a blanket exclusion of employees with criminal convictions could be discriminatory, in that it may have a disparate impact on minorities.[7] Therefore, such issues are generally handled on a case-by-case basis and depend heavily on the risk and sensitivity of the particular job position. As discussed above, many IT jobs and especially information security jobs are highly sensitive, but that does not mean that you can do whatever you want on this issue. Make sure you label job requisitions for IT personnel, and especially security personnel, as being very sensitive, requiring a clean background check.

This ambiguity in laws can be a major problem in establishing your own policies. Based on the lack of clear regulations on this point, many companies prohibit interviewers from asking job applicants about criminal background activities, unless a clean slate is an explicit, *bona fide* job requirement. Additionally, in many companies, you can only ask about actual criminal convictions, and not mere indictments or arrests. So, you may be allowed to find out that a job applicant was convicted of unsuccessfully trying to hack into a system and steal one million dollars. However, you may *not* be able to find out about another job applicant who successfully stole ten times that amount, but was acquitted on a technicality. Because the former case resulted in conviction but the latter was dismissed, you may only get the useful information about the first.

Your best bet here is to check with your Human Resources organization. After all, these folks get paid to know about the laws in your area regarding recruitment and to interpret those laws within your organization. Get a copy of any restrictions on interview questions or hiring limitations in writing from your HR organization before moving forward.

Background Checks That Really Mean Something

One of the most important things you can do to ensure the trustworthiness of your employee base is good old-fashioned background checks of potential hires. Start with investigating the references included in the candidate's resume. Some organizations just assume that the recruiter or headhunter who identified the candidate double-checked all references. Unfortunately, in the vast majority of cases, that is just not true. To conduct a thorough interview process, call each reference and verify the candidate's background and skills. Any discrepancies could indicate a big problem that you can nip in the bud in the interview process.

Beyond calling references, you may want to consider checking with the National Fraud Center (http://www.nationalfraud.com) or other background checking services to see if they have any records indicating fraudulent activity by the interviewee. These services are available for a nominal fee and can provide significant value to an organization. A record with the National Fraud Center is a significant red flag in the hiring process.

Although it has less value, you also may want to check the credit history of the potential employee. Credit histories have less value in the employment process simply because a large debt load and even a history of failure to pay debts may simply indicate that person really just needs a job. Credit problems do not necessarily indicate the risk factor of a potential hire. Carefully consider your policy on credit checks, and document in writing how you will use this information in your hiring decisions. What would you do if someone has bad credit? Would you not hire them? You may determine that credit checks do not really provide you the information you need to make hiring decisions.

Many companies also perform drug testing before any new employees can start a job. While some people consider these tests invasive, they are becoming quite commonplace. (I personally do not think such tests are very persuasive in determining someone's criminal background with respect to computer attacks.)

Reference checks, fraud reviews, credit checks, and drug tests are not enough to ensure the trustworthiness of employees for extremely sensitive job positions. Consider ethical hacking consultants who are paid to break into the networks of clients who request penetration tests. These employees have access to the keys to their clients' kingdom, and permission to storm the castle looking for valuables. Likewise, the leaders of an information security team and chief system administrators have access to all information stored on an organization's computers. For these highly sensitive positions, when possible, hire only people that you have known for at least one year. For these tasks, promote from within, or use people whose backgrounds you have personally witnessed for over a year to ensure you can trust them. Such a policy can obviously limit the speed of growth of your organization, but it is a good start in establishing the trustworthiness of the top of your IT and security groups.

Establish a Mentor Program

One of the most effective things you can do to help detect suspicious activity by new employees in an IT organization is to develop a mentor program. After doing strong reference and background checks, assign every new employee a mentor who is a more senior, trusted member of staff. Each new hire should get a mentor for six to twelve months. Mentors are officially tasked as part of their job description with supporting new employees in their transition to the company.

In addition to helping the new employee, the mentor also acts as the eyes and ears of the company. The mentor can ensure that the new employee has the skills and attitude necessary to do the job, without exposing the company to risk. If mentors suspect that new hires have ill-will toward the company or are conducting insider attacks, they should report their concerns to management. This is not to say that mentors should be Big Brother, silently stalking every move of the new hire. However, mentors should have general knowledge of the activities of their assigned new hires. Not only can mentors help improve security through detecting and even preventing insider attacks, they can also be quite helpful in improving the productivity of new employees by getting them up to speed quickly.

We Are from the Government and We Are Here to Help

If you decide to hire ex-criminal hackers and you work for a U.S.-based company, you could benefit from a program established by the U.S. Department of Labor to help lower the financial risk companies face when hiring high-risk employees, such as ex-convicts. To encourage employers to hire such people, this federally funded Bonding Program is available to employers free of charge. The Department of Labor highlights the benefits of this program at its Web site as follows:[8]

> Jobseekers who have in the past committed a fraudulent or dishonest act, or who have demonstrated other past behavior which casts doubt upon their credibility or honesty, often experience a special barrier to gaining employment due to their personal backgrounds. Such persons are routinely classified as "at-risk" job applicants.
>
> These jobseekers, whose past life experience raises an obstacle to their future ability to secure employment, could benefit from the Federal Bonding Program. Created in 1966 by the U.S. Department of Labor, the Federal Bonding Program helps to alleviate employers concerns that at-risk job applicants would be untrustworthy workers by allowing them to purchase fidelity bonds to indemnify them for loss of money or property sustained through the dishonest acts of their employees… It is like a "guarantee" to the employer that the person hired will be an honest worker.

Keep in mind, however, that the bond only covers up to U.S. $5000 in damages. Admittedly, in a computer attack, $5000 in damages can occur in milliseconds. Still, this insurance program, which is operated for the Department of Labor by Travelers Property Casualty, may be helpful.

You can expand the coverage beyond $5000, but the additional coverage costs come out of your pockets, and not the taxpayers'. Additionally, if, instead of doing interviews, you are the one looking for a job and have a spotty record, you can get bonded yourself, to help assuage any concerns a potential employer may face.

Beyond Employee Issues: Consultants and Contractors

A final but very important point to consider regarding the potential insider threat of ex-criminal hackers goes beyond the borders of your own organization. Sure, you would never hire someone who was widely known throughout the computer underground as "Death Kiddie" and served five years in prison for wreaking hacking havoc on another company, but what about the firms you hire for IT consulting or outsourcing? Contractors, consultants, or even temporary employees could easily be attacking your organization from the inside.

There have been cases where a temp gets a job with a particular organization for a few short weeks just for the purposes of installing backdoors and other hacking tools on the organization's internal systems. After the brief stint as a temp is over, the attacker covertly controls these hacking tools from the privacy of his own home. Furthermore, some of the world's largest information security consulting firms hire ex-criminal hackers

or sub-contract their security business to ex-criminals. These people may be assigned to your ethical hacking exercises, firewall deployments, or security design tasks if you contract for consulting services from such companies. Do you trust these people? Do their hiring practices regarding ex-criminal hackers meet your own internal policies?

To deal with this problem, you need to be aware of the threat and require your contractors and temp agencies to carefully screen the applicants they send to your company. Similarly, before signing a contract for a project with a consulting company, ask about the consultant's hiring practices with respect to background checks and employing ex-criminal hackers. Make sure that your consultant's answer to this question lines up with your own company's philosophy and policies.

Conclusion

Most information security organizations do not pay much attention to the criminal backgrounds of their own employee base. You should carefully consider what impact such backgrounds should have on your hiring process, and coordinate your explicit policies with your Human Resources organization. Do not shun ex-criminal hackers for every job, but instead, carefully consider the particular job requirements and risks. By carefully structuring your own hiring program, as well as selecting contractors and consultants with a similar philosophy, you can make sure your organization is properly protected.

Note

1. Kevin Mitnick, noted computer attacker of the 1980s and early 1990s, served a lengthy jail sentence. During his incarceration, a significant movement sprung up trying to get Mitnick released. Spearheaded by *2600* magazine, this movement is recognized for its widespread distribution of "Free Kevin" bumper stickers. For more information, see References 5 and 6.

References

"Computer Security Issues and Trends: 2002 CSI/FBI Computer Crime and Security Survey," Richard Power, April 2002, http://www.gocsi.com/press/20020407.html.

"Analysis of Recidivism Rates for Participants of the Academic/Vocational/Transition Education Programs offered by the Virginia Department of Correctional Education," June 2000, http://www.easternlincs.org/correctional_education/Hull.pdf

2600 Magazine, subscription information online at http://www.2600.org/magazine/.

The Watchman: The Twisted Life and Crimes of Serial Hacker Kevin Poulsen, Jonathan Littman, Little Brown, 1997.

The Fugitive Game: Online with Kevin Mitnick, Jonathan Littman, Little Brown, 1997.

Takedown, Tsutomu Shimomura and John Markoff, Hyserion, 1996.

"Hiring Managers Face Challenges with 'High-Risk' Candidates," article by Jerry L. Ledford, September 2001, Smart Pros, http://finance.pro2net.com/x28047.xml.

Federal Bonding Program Information from the Department of Labor: http://wtw.doleta.gov/documents/fedbonding.asp.

Information Security and Personnel Practices

Edward H. Freeman

In the past few years, the corporate world's image of the personnel function has undergone a significant change. An organization's employees are now considered a corporate resource and asset, requiring constant care and management. Changing legal conditions affecting personnel practices have underscored the need for clearly defined and well-publicized policies on a variety of issues.

The corporation and the employee have specific legal and ethical responsibilities to each other, both during and after the period of employment. Hiring and termination criteria, trade secrets, and noncompetition clauses are all issues that can cause serious legal problems for a corporation and its employees.

This chapter addresses personnel issues as they relate to information systems security, particularly hiring and termination procedures. Methods to protect both the corporation and the employee from unnecessary legal problems are discussed, and problems regarding trade secrets and noncompetition clauses are reviewed.

THE PROFESSIONAL ENVIRONMENT

The information systems and information security professions are in a vibrant and exciting industry that has always operated under a unique set of conditions. The industry relies on the unquestioned need for absolute confidentiality, security, and personal ethics. An organization and its reputation can be destroyed if its information security procedures are perceived as being inadequate or unsatisfactory. Yet, misuse or outright theft of software and confidential information can be relatively easy to accomplish, is profitable, and is often difficult to detect. Innovations can be easily transferred when an employee leaves the corporation, and information

systems personnel have always been particularly mobile, moving among competitors on a regular basis.

These factors are extremely important as they relate to the corporation and its personnel practices. A newly hired programmer or security analyst, whose ethical outlook is largely unknown to management, may quickly have access to extremely sensitive and confidential information and trade secrets. Unauthorized release of this information could destroy the corporation's reputation or damage it financially. An employee who has just accepted a position with a major competitor may have access to trade secrets that are the foundation of the corporation's success.

HIRING PRACTICES

Corporations must take special care during the interview to determine each candidate's level of personal and professional integrity. The sensitive nature and value of the equipment and data that employees will be handling require an in-depth screening process. At a minimum, this should include a series of comprehensive interviews that emphasize integrity as well as technical qualifications. References from former employers should be examined and verified.

The best way to verify information from an employment application is to conduct a thorough reference check with former supervisors, co-workers, teachers, and friends listed by the applicant on the application. Former employers are usually in the best position to rate the applicant accurately, providing a candid assessment of strengths and weaknesses, personal ethics, and past earnings, among other information.

Many employers have become increasingly cautious about releasing information or making objective statements that rate former personnel. Such employees have successfully sued corporations and supervisors for making derogatory statements to prospective employers. Many employers will furnish written information only about the applicant's dates of employment, positions held, and salaries earned, choosing to ignore more revealing questions. Often, an informal telephone check may reveal more information than would be obtained by a written request. If two large employers regularly hire each others' employees, it would be worthwhile for their personnel managers to develop a confidential personal relationship.

Use of a reference authorization and hold-harmless agreement can help raise the comfort level of the former employer and get more complete information from a job applicant's previous employer. In such an agreement, the applicant authorizes the disclosure of past employment information and releases both the prospective employer and the previous employer from all claims and liabilities arising from the release of such

information. An employer who uses such an agreement should require every job applicant to sign one as a condition of applying for employment. A copy of the agreement is then included with the request for references sent to the previous employer.

When sending or responding to a reference request that includes a reference authorization waiver and hold-harmless agreement, it is important for employers to make sure that the form:

- Is signed by the job applicant.
- Releases the employer requesting the information as well as the previous employer from liability.
- Clearly specifies the type of information that may be divulged.

A responding employer should exercise extreme caution before releasing any written information about a former employee, even if the former employee has signed a reference authorization waiver. Only information specifications permitted by the waiver should be released. If there is any ambiguity, the former employer should refuse to release the requested information. The former employer is safest if only the date of hire, job title, and date of termination are released.

TRADE SECRETS

A trade secret is a "formula, pattern, device, or compilation of information which is used in one's business, and which gives an opportunity to obtain an advantage over competitors who do not know or use it." (Restatement of Torts, Section 757 [1939].) This advantage may be no more than a slight improvement over common trade practice, as long as the process is not common knowledge in the trade. A process or method which is common knowledge within the trade is not considered a trade secret and will not be protected. For example, general knowledge of a new programming language or operating system that an employee may gain on the job is not considered a trade secret. The owner of a trade secret has exclusive rights to its use, may license another person to use the innovation, and may sue any person who misappropriates the trade secret.

Trade secret protection does not give rights that can be enforced against the public, but rather against only those individuals and organizations that have contractual or other special relations with the trade secret owner. Trade secret protection does not require registration with government agencies for its creation and enforcement; instead, protection exists from the time of the invention's creation and arises from the developer's natural desire to keep his or her invention confidential.

Strict legal guidelines to determine whether a specific secret qualifies for trade secret protection have not been established. To determine

whether a specific aspect of a computer software or security system qualifies as a trade secret, the court will consider the following questions:

- Does the trade secret represent an investment of time or money by the organization which is claiming the trade secret?
- Does the trade secret have a specific value and usefulness to the owner?
- Has the owner taken specific efforts and security measures to ensure that the matter remains confidential?
- Could the trade secret have been independently discovered by a competitor?
- Did the alleged violator have access to the trade secret, either as a former employee or as one formerly involved in some way with the trade secret owner? Did the organization inform the alleged violator that a secrecy duty existed between them?
- Is the information available to the public by lawful means?

Trade secret suits are based primarily on state law, not federal law. If the owner is successful, the court may grant cash damages or injunctive relief, which would prevent the violator from using the trade secret.

Trade Secrets and Personnel Practices

Because information systems and security professionals often accept new positions with competitors, organizations seeking to develop and protect their information assets must take special care to determine each candidate's level of personal and professional integrity. The sensitive nature and value of the equipment and data that employees will be handling require an in-depth screening process. At a minimum, this should include a series of comprehensive pre-employment interviews that emphasize integrity as well as technical qualifications. Careful reference checking is essential.

When an employee joins the firm, the employment contract should expressly emphasize the employee's duty to keep certain types of information confidential both during and after the employee's tenure. The contract should be written in clear language to eliminate any possibility of misunderstanding. The employee must sign the agreement before the first day of work as a condition of employment and it should be permanently placed in his or her personnel file. A thorough briefing on security matters gives the employee initial notice that a duty of secrecy exists, which may help establish legal liability against an employee who misuses proprietary information.

These secrecy requirements should be reinforced in writing on a regular basis. The organization should inform its employees that it relies on trade secret law to protect certain proprietary information resources and that

the organization will enforce these rights. All employees should be aware of these conditions of employment.

The entrance interview provides the best opportunity to determine whether new employees have any existing obligations to protect the confidential information of their former employers. If such an obligation exists, a written record should be entered into the employee's personnel file, outlining the scope and nature of this obligation. In extreme cases and after consultation with legal counsel, it may become necessary to reassign the new employee to an area in which this knowledge will not violate trade secret law. Such actions reduce the risk that the former employer will bring an action for trade secret violation.

The employee should acknowledge in writing that he or she is aware of this obligation and will not disclose any trade secrets of the former employer in the new position. In addition, the employee should be asked if he or she has developed any innovations that may be owned by the former employer.

The organization should take special care when a new employee recently worked for a direct competitor. The new employer should clearly emphasize and the new employee should understand that the employee was hired for his or her skills and experience, not for any inside information about a competitor. The employee should never be expected or coerced into revealing such information as part of his or her job. Both parties should agree not to use any proprietary information gained from the employee's previous job.

Trade Secrets and the Terminating Employee

Even when an employee leaves the organization on excellent terms, certain precautions regarding terms of employment must be observed. The employee should be directed to return all documents, records, and other information in his or her possession concerning the organization's proprietary software, including any pertinent notes (except those items the employee has been authorized in writing to keep).

During the exit interview, the terms of the original employment agreement and trade secret law should be reviewed. The employee should then be given a copy of the agreement. If it is appropriate, the employer should write a courteous, nonaccusatory letter informing the new employer of the specific areas in which the employee has trade secret information. The letter should be sent with a copy of the employee's employment agreement. If the new employer has been notified of potential problems, it may be liable for damages resulting from the wrongful disclosure of trade secrets by the new employee.

NONCOMPETITION CLAUSES

Many firms require new employees to sign a noncompetition clause. In such an agreement, the employee agrees not to compete with the employer by starting a business or by working for a competitor for a specific time after leaving the employer. In recent years, the courts have viewed such clauses with growing disfavor; the broad scope of such agreements severely limits the former employee's career options, and the former employer has no obligations in return.

Such agreements, by definition, constitute a restraint on free trade and are not favored by courts.To be upheld by the court, such agreements must be considered reasonable under the circumstances. Most courts analyze three major factors when making such determinations:

- Whether the specific terms of the agreement are stricter than necessary to protect the employer's legitimate interests.
- Whether the restraint is too harsh and oppressive for the employee.
- Whether the restraint is harmful to the interests of the public.

If an employer chooses to require a noncompetition clause from its employees, care should be taken to ensure that the conditions are only as broad as are necessary to protect the employer's specific, realistic, limited interests. Clauses which prohibit an employee from working in the same specific application for a short time (one to three years) are usually not considered unreasonable.

For example, a noncompetition clause which prohibits a former employee for working for a direct competitor for a period of two years may be upheld by the court, whereas a clause which prohibits a former employee from working in any facet of information processing or information security will probably not be upheld.

The employer should enforce the clause only if the former employee's actions represent a genuine threat to the employer. The court may reject broad restrictions completely, leaving the employer with no protection at all.

PRECAUTIONARY MEASURES

Organizations can take several precautionary steps to safeguard their information assets. Perhaps the most important is to create a working atmosphere that promotes employee loyalty, high morale, and job satisfaction. Employees should be aware of the need for secrecy and of the ways inappropriate actions could affect the company's success.

Organizations should also ensure that their employees' submissions to technical and trade journals do not contain corporate secrets. Trade secrets lose their protected status once the information is available to the

public. Potential submission to such journals should be cleared by technically proficient senior managers before submission.

Intelligent restrictions on access to sensitive information should be adopted and enforced. Confidential information should be available only to employees who need it. Audit trails should record who accessed what information, at what times, and for how long. Sensitive documents should be marked confidential and stored in locked cabinets; they should be shredded or burned when it is time to discard them. (It should be noted that some courts have held that discarded documents no longer remain under the control of the creator and are in the public domain.) Confidential programs and computer-based information should be permanently erased or written over when it is time for their destruction. These measures reduce the chance of unauthorized access or unintentional disclosure.

To maintain information security, organizations should follow these steps in their personnel practices:

- Choose employees carefully. Personal integrity should be as important a factor in the hiring process as technical skills.
- Create an atmosphere in which the levels of employee loyalty, morale, and job satisfaction are high.
- Remind employees, on a regular basis, of their continuous responsibilities to protect the organization's information.
- Establish procedures for proper destruction and disposal of obsolete programs, reports, and data.
- Act defensively when an employee must be discharged, either for cause or as part of a cost reduction program. Such an employee should not be allowed access to the system and should be carefully watched until he or she leaves the premises. Any passwords used by the former employee should be immediately disabled.
- Do not be overly distrustful of departing employees. Most employees who resign on good terms from an organization do so for personal reasons, usually to accept a better position or to relocate. Such people do not wish to harm their former employer, but only to take advantage of a more suitable job situation. Although the organization should be prepared for any contingency, suspicion of former employees is usually unfounded.
- Protect trade secrets in an appropriate manner. Employees who learn new skills on the job may freely take those skills to another employer, as long as trade secrets are not revealed.
- Use noncompetition clauses only as a last resort. The courts may not enforce noncompetition clauses, especially if the employee is unable to find suitable employment as a result.

76

Information Security Policies from the Ground Up

Brian Shorten, CISSP, CISA

Security is people-based. As Bruce Schneier says in *Secrets & Lies*, "If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology." The first step in a coordinated security process is a security policy.

Reasons for a Policy

It cannot be stated too strongly that the security policy is the foundation on which all security is based. Ironically, when trying to introduce a policy, a security practitioner may encounter resistance from a senior management structure, which sees the one-off purchase of an anti-virus application as the solution to all security problems. In such circumstances, it follows that the security practitioner must explain to senior management the purpose of a policy.

A formal security policy, signed by the CEO, defines how the company intends to handle security and states that the company is not only concerned about security, but intends to take it seriously. Note the phrase "signed by the CEO." This is an important part of the overall process. It is vital that staff can see that there is management buy-in right from the top. Although sign-off from the security manager or director is good, it does not convey the same message. After all, as some staff members see it, the security manager or director is expected, and paid, to care about security.

So, what meaning does the policy put into words? The information security policy tells staff members what they CAN do, what they CANNOT do, what they MUST do, and what their RESPONSIBILITIES are.

What Should Be in a Policy

There are many books written on what should be contained in a policy. Some say that the policy should be short, a series of bulleted points covering only one side of a sheet of paper. Some even give examples, which can be adopted and modified for the practitioner's own company.

Although a short document may have more chance of being read by its intended audience, most of these samples are basically mission statements, which must still be supported by a more detailed policy. The author suggests that the mission statement be used as a personal foreword, signed by the CEO, to the policy.

Policy versus Procedures

A policy states what should be done. Procedures define how to implement the policy. For example, if the policy says, "All applications must have a password," the procedure would detail exactly how the password for each application is to be created and maintained.

Contents of the Policy

The following issues should be addressed by the policy.

Access Control Standards

Users should have access to the information and applications they require to perform their job functions, and no more. A discretionary access control policy must be implemented to provide users with that level of access. Users are responsible for managing the necessary changes to their passwords. Where possible, users will be automatically prompted to change their passwords every 30 days.

Accountability

It is important that users are held accountable for all actions carried out under their user IDs. Users must ensure that when they are away from their desks, their computer is in a secure state (i.e., the screen saver is activated with password protection, or in "lock workstation" mode.

Audit Trails

The actions carried out by users must be recorded and logged. Specifically, the following actions should be logged:

- A minimum of 30 days of user sign-on and sign-off details
- All unauthorized attempts to read, write, and delete data and execute programs
- Applications must provide detailed audit trails of data changes, when required by the business

It is the data owner's responsibility to identify such audit trail requirements.

Backups

All software and user data will be backed up to alternative media on a regular basis and kept in a secure area. The frequency of the backups, which must be specified in the policy, will be appropriate to the importance of the system and the data that would need to be recovered in the event of a failure.

Business Continuity Plans

The tendency is to concentrate on information security systems when considering a business continuity plan (BCP). There should be a contingency plan for all computer services that support critical systems, and that plan should have been designed, implemented, and tested. The BCP should identify those services that are critical to the operation of the business, and ensure that contingency plans are in place. These contingency plans need to take into account a variety of disaster recovery scenarios.

Disposal of Media

The manner in which hardware and storage media — such as disk drives, floppy disks, and CD-ROMs that contain confidential data — are destroyed when no longer required must be carefully considered. An unauthorized person can retrieve data from media if it has not been obliterated correctly. Use of the ERASE, DELETE, and FORMAT functions is not sufficient. There are many freely available applications that can easily reverse these functions. Therefore, methods should be used that can overwrite media so data cannot be retrieved, or products should be used that degauss the media so data is obliterated and cannot be read. For confidential data, the media may require physical measures to render it unreadable — destroying hard drives with a hammer, shredding floppy disks, cutting CD-ROMs. The policy should lay down the agreed-to method for this disposal, depending on media type and the data in question.

Disposal of Printed Matter

Despite this being the age of the paperless office, many people prefer to print documents and write their comments. In such circumstances, it is easy to forget that the confidentiality of the printed data is unchanged by being printed — confidential data remains confidential. Once printed, these sheets containing confidential data must be disposed of carefully, and not in the nearest waste bin. All staff must have convenient access to a shredder. The shredder used must be cross-cut to reduce the chances that an unauthorized person, using sticky tape, could reconstruct the sheet.

Downloading from the Internet

Most businesses currently give their staff members access to the Internet. Although such access is usually intended for business use only, the security practitioner must ensure that the policy advises staff clearly on how that access is to be used, both to maximize the use of bandwidth and to prevent illegal acts from being carried out. The policy must state very clearly that Internet access is provided for business use only. Employees who have doubts as to what is correct business use should be advised to consult their line management for approval prior to accessing Internet information. Staff should be expressly forbidden to access, load, view, print, copy, or in any way handle obscene material from any source using company facilities.

Information Ownership

It is important that all data be assigned an owner who can make a decision as to who should be able to access that data. Because this decision is a business decision, the owner should be from the business and possess a good knowledge of business processes and the data.

Management Responsibilities

Managers, at all levels, have responsibilities for information security. These responsibilities may be mainly to ensure that all their staff members understand and comply with the policy, but such responsibilities need to be laid out in the policy itself to remove any misunderstanding. Each person holding a management or supervisory position is responsible for noting and reporting any deviations from the policy.

Modems and Analog Lines

Modems allow the use of an analog line, which circumvents the firewall and exchange gateway. Therefore, it follows that there is no anti-virus check on any data to and from the modem. Analog lines are now used by faxes, conference phones, and video phones. Some desk phones also require analog lines for the facilities they provide to users, such as voicemail. For these reasons, the security practitioner must ensure that the installation of analog lines for **any** use is prohibited unless prior authorization is given after the requestor has provided a business case for the line as full justification. It also follows that when a modem is in use, there must be no simultaneous connection to the company network, to prevent any computer virus from being "imported" to the network.

Off-Site Repairs to Equipment

Although most companies have an internal department to repair equipment, there are occasions when those repairs will need to either be sent off-site, or for a third party to come to the company to make repairs. It is vital to be sure who has access to company equipment and company data. If the data could be classified as confidential, it should be removed from any media before any non-company member of staff is allowed to work on the equipment.

Physical Security

Security is multi-layered; physical may be considered the first level of security. Although authorization and authentication processes control logical access, physical access security measures are required to protect against the threats of loss and damage to the computing-based equipment and information. All assets and materials are required to be protected from unauthorized use or removal, or damage, whether accidental or deliberate. The physical security policy of the company ensures that information systems, their peripherals, removable storage media, electrical services, and communications services are protected from unauthorized access and from damage as far as possible, consistent with a cost-efficient operation.

Portable Devices

The days are long gone when a PC was so heavy it could not easily be moved from a desk. Laptop computers are now as powerful as desktops, and create new problems because portability makes laptops easy to take out of the office, and easy to steal. Users must be made aware that such equipment issued to them is their responsibility, both in and out of the office. Not only can the laptop be stolen and therefore lost to the company, but any information on the laptop will also be lost or compromised if not encrypted. The security practitioner should always consider that the information may well have a higher value than the replacement cost of the laptop. For example, consider the information on the merger or takeover of one global company by another.

The security practitioner should also think about the growing use of various personal digital assistants (PDAs) such as PalmPilots, Psion organizers, etc. These are extremely vulnerable because they have a high value and are extremely portable. In addition, users often download documents from the company systems to a personal PDA for convenience; such equipment often does not have more than rudimentary security.

Users must be made aware that care of PDAs must be taken when traveling to avoid their loss or compromise, and that they must not be left unattended in public areas. When left in cars, houses, or hotel rooms, users must take all possible measures to ensure their security. As a method to persuade users to take care of laptops, a process should be used to request that laptop users confirm that they still have the laptop in their possession when they return to the office.

Staff Responsibilities

Just as managers have specific responsibilities by virtue of their positions, staff members also have responsibilities for security, the most fundamental of which is the protection of the company's information assets. For employees to carry out these responsibilities, they are required to:

- Understand and comply with the company's security policies.
- Know and follow all instructions for controlling access to, and use of, the company's computer equipment.
- Know and follow all instructions governing the secure handling of the company's information assets.
- Keep all passwords secret and be aware that they must never be given to anyone.
- Be aware that some actions are expressly forbidden for staff. Forbidden actions could include:
  - Installing, executing, downloading, uploading, or in any other way introducing third-party software onto company computer equipment, or in any way changing, deleting, or reconfiguring the standard desktop without written authority (prior to the installation) from both the IT security department and the IT department.
  - Abuse of any special account privileges that may have been granted to that staff member.
- Understand that each employee is responsible for noting and reporting any deviations from the company's security policy.

The security practitioner must ensure that all staff members realize that the computer equipment, software, and services provided by the company are for authorized business use only, and that staff members must not use the equipment for any other purpose unless authorized in writing to do so by their line manager. At this stage, staff members must be warned that violation of the security policy is deemed a serious offense and may result in disciplinary action.

Use of E-Mail

With so much of modern business dependent on e-mail, the security policy must ensure that the company's attitude toward staff members' use of e-mail is well-known. It should also be considered that, legally, an e-mail carries the same weight as a letter on company letterhead. In the recent past in the United Kingdom, an e-mail with a derogatory comment about a rival company was legally held to be the responsibility of the original company. In this case, the rival company sued the original company, despite the fact that the e-mail was initially between two employees, and not "official." The aggrieved company sued the original company, which had money for costs, rather than the employees, who had none.

Staff members must be made aware that the company provides internal mail and e-mail facilities for business use only. Many companies currently allow staff members to send and receive personal e-mails using the company system. In these circumstances, staff members must know that this is a concession that must not be abused, either by the number of e-mails sent or the time taken from the business day to deal with personal e-mails.

Such personal use must be at the discretion of the user's line manager.

As described, personal use of the company e-mail system may be permitted. However, provision should be made for monitoring or reviewing all e-mails into and out of the company. There are reasons why this may be necessary — the authorities may present a warrant to view e-mails as part of an investigation or the company itself may have the need to follow up on a fraud involving company systems and finances.

The security practitioner should also be aware of the decisions of recent legal findings on personal e-mail. If the policy says, "No personal e-mails sent or received," but the practice is that staff members do send and

receive e-mails without any comment or censure from managers, the courts will be guided by the practice, rather than the policy, and find accordingly.

The policy should contain a clear warning to staff that no employee or user of the company e-mail system should have any expectation of privacy with respect to any electronic mail sent or received. The company may, at any time and without prior notification, monitor, review, audit, or control any aspect of the mail system, including individual accounts. It follows that this process should have built-in internal control processes that are subject to audit, to ensure that the ability to review e-mail is not abused

The policy should address the contents of e-mails, and include reference to attachments to e-mails, which themselves may pose a risk to company systems. Such a reference could be:

- No computer software, files, data, or document that may give rise to violation of any policy, law, license agreement, or copyright law should be attached to or sent with any e-mail communication.
- Inappropriate use of the e-mail system(s) may result in disciplinary action. "Inappropriate use" is the dissemination of any text, software (or part thereof), or graphics (including moving graphics) that violate any company policy.
- In addition, any mail, the content of which is considered profane, sexist, racist, sexual, or in any way discriminatory to any minority, is also "inappropriate use."
- Employees are responsible for checking files received via e-mail for viruses and content.
- Any mail received by employees that breaches policy must be reported to the Security Department immediately.

Viruses

Despite the best efforts of the anti-virus industry, and IT and security professionals, computer viruses continue to be distributed globally. Staff members should be aware that they have a part to play in the anti-virus process, and that it is essential that any data files that come into the company are virus checked before being loaded to the data network. Any questions regarding virus checking should be directed to the Help Desk. Staff members should not be discouraged from reporting to management or the IT department if they believe they have detected a virus.

Workstation Security

There is a real threat to the security of systems when a user leaves a terminal logged in to a system and the terminal is left unattended; this terminal can then be used by an unauthorized person. In such a circumstance, the unauthorized person can use the terminal and access the system, just as if the authorized user were present, without having to know or guess the authorized user's sign-on or password. For this reason, users must be advised not to leave a terminal logged in, without use of a password-protected screen saver. Some systems may themselves have a process whereby inactivity of the keyboard or mouse will automatically prevent use of the terminal unless the authorized user enters a password. If such a process exists, the policy should be written to require its use. For a similar reason, users should not be allowed to be signed on to the same system at multiple terminals simultaneously.

Privacy

Although most companies do not have the resources, or the reason, to monitor e-mails on a regular basis, there will be occasions when it will be necessary to check the e-mail of a particular staff member. The security practitioner should prepare for that occasion by ensuring that the policy spells out the company's stance on privacy. An example of such a statement might be:

> No employee or user of the company mail system(s) should have any expectation of privacy with respect to any electronic mail sent or received. The company may, at any time without prior notification, monitor, review, audit or control any aspect of the mail systems, including individual accounts. This process has internal control processes and is subject to audit.

By using such a statement, staff members will then be aware that the facility to monitor e-mail exists, but that is bound by checks and balances.

Noncompliance

Having written a policy that specifies what behavior is expected of staff members, it is necessary for the security practitioner to ensure that the policy also contains a reference to the consequences of noncompliance. Such stated consequences may simply be that *non-compliance may result in disciplinary action,* which should suffice. Note the use of the word "may." This leaves management with various options for disciplinary action, which can run from a verbal warning to dismissal.

Legislation

With the increase in global trading, it is vital that security practitioners become conversant with the various legislation relevant to the different aspects of information security. This is becoming more and more vital as more and more companies operate on an international basis, having offices, staff, and customers in many countries. In this case, the policy must make reference to all relevant legislation, and include the relevant legislation for every location where the company has staff members who are expected to comply with the policy. For a company with offices throughout the world, this would be a separate appendix.

Other Issues

It is important to make the policy a document that can be utilized by staff members. To this end, the security practitioner must include separate appendices for choosing secure passwords and advice on good security practice. The security practitioner should consider that the overall security policy is an umbrella document that forms the basis of separate implementing security policies, while standards and baselines, which form the next level, can be application-specific.

The overall policy should not be too specific. Specifying "must have a password that meets current standards" is better than stating the exact size, format, and make-up of the password. After all, the company will have several applications requiring a password, and it is certain that different rules will apply in each case.

In addition, there are others in the company who have input to the process of creating the policy. The Legal department should be involved to ensure that the wording of the policy is correct; it is particularly important that the human rights legislation is taken into account, particularly in the sections covering staff responsibilities. The Human Resources department needs to confirm that the company disciplinary process is adequate for the task. If the policy specifies a disciplinary action for staff members who do not comply with the policy, there must be willingness on the part of the company, and the Human Resources department, to take that action — otherwise, the policy is useless.

The company's Data Protection Officer must be involved to ensure that the policy complies with the data protection legislation in all relevant countries.

The First Steps in Writing a Security Policy

In starting the process of creating a security policy, the security practitioner has several resources. The international standard ISO I7799, created by a group of international companies to form the basis of a security policy, started life as a guideline issued by the Department of Trade and Industry in the United Kingdom, then became a British Standard, BS 7799, before being adopted as ISO 17799. Professional peers, such as other security practitioners with the CISSP designation, can also offer advice and support. Books are also available for the security practitioner to consult.

The security practitioner has other considerations that are more allied with the culture and environment of the company concerned, particularly if this is the first policy for that company. This is where you need to consider the culture of the company.

The following gives a real-life example:

> The company, with 300 staff members, had one floor in a shared building and there had been problems with outsiders coming in and property being stolen. The first draft policy said, "all staff must wear the identity badge issued to them," and "all staff are to challenge anyone not known to them." This is not too excessive. However, because the CEO did not like all staff to wear an identity badge, because he himself felt self-conscious doing so, the policy was changed to "all staff must have an identity badge." Senior managers balked at challenging strangers because they said they would take forever to get to the bathroom in the morning. This section of the policy became, "if you see

someone in your area who you don't recognize, you should query this with departmental managers or HR." In such cases, the security practitioner has to accept the culture, amend the first policy, and review it again in a few months. No surprise: the thefts of property continued.

The lesson for the security practitioner to learn here is that the policy must cover all staff members: if the policy says, "wear a badge," it sends the wrong signal if senior management and higher take the view that "everyone knows me" and leave their identity cards in their wallets.

Once the policy is drafted, the security practitioner must ensure that all interested parties are involved and invited to make comments. These parties are Legal, Audit, Human Resources, and Data Protection as previously mentioned, plus the IT department. Any member of the board who has shown an interest should also be included. After comments are invited, and any necessary changes made to the policy, the security practitioner should submit the policy to the board for acceptance and sign-off by the CEO.

It is important to cover all issues before submitting the draft. It should only be submitted to the board once for acceptance; having to make changes and return will only weaken the security practitioner's credentials as the company security guru.

The Next Steps

Once the policy is written, accepted by the board, and signed by the CEO, the security practitioner must ensure that the policy is read and accepted by staff members. There are various methods for this, all of which should be considered by the security practitioner; these include:

- Print enough copies for all staff members, and distribute them throughout the company.
- Have the Human Resources department send a copy to all new staff members with the new joiner details.
- E-mail a copy to all staff members.
- Place a copy on a server that all staff members can access, and e-mail the shortcut to all staff members.
- Place a copy on the company intranet and e-mail the shortcut to all staff members.
- Place posters advising staff members of the policy in staff refreshment areas.
- Issue mouse pads with security hints to all staff members.
- Use log-on banners for various applications that contain security advice.

However, having considered the several ways to communicate the policy to staff, security practitioners must be selective in their application to avoid having staff get so many copies that they switch off and ignore the message.

It is important to have staff agreements that they have read, and will comply with, the policy. These agreements will provide useful evidence should any staff members dispute the fact that they have read and understood the policy after having committed some act that contravenes the policy.

Whichever method the security practitioner selects to send the policy to staff, it is vital to receive back a signed document of agreement or a specific e-mail acknowledging acceptance of the policy. Either method of acceptance can be utilized. However, for the security practitioner, a form that the user can read, sign, and return is preferable. The form can then be kept by HR and constitute part of the staff member's file.

Reviewing the Policy

The security practitioner must remember that a security policy is a "living document" that must be reviewed regularly and updated as necessary. This should occur at least every six months. There are several issues to be addressed as part of the review, including:

- The policy must continue to be relevant. References to outdated equipment must be removed. The policy may refer to floppy disks although there are no PCs with floppy disk drives in the company.
- Processes may have changed. If the policy on computer viruses refers only to virus scanning floppy disks, although the company has moved to virus scanning on all servers and terminals, the policy needs to be updated.
- New technology may have been introduced since the policy was written.
- Senior managers may now be issued personal digital assistants (PDAs).

Once the policy has been reviewed and updated, it must be resubmitted to the board for acceptance and signed again by the CEO.

Staff Awareness

The security practitioner must be aware that although it is the responsibility of the security department to produce and maintain the security policy, security is a process that should involve all staff members. If staff members see security as something that is an obstacle to their work, they will not take on their proper responsibility, and worse, will go out of their way to find a work-around to any security measure they do not consider necessary.

The security practitioner needs staff members to understand why security is important, and that they themselves are being protected. A staff awareness process will follow the process discussed earlier. Again, care should be taken to be selective in their application to avoid reaching such overload that staff members switch off and ignore the message.

The security practitioner should remember that it is not possible to be everywhere at once; an educated staff can go a long way toward acting on the behalf of the practitioner.

Educated users are more likely to pick a good password, challenge a stranger, or lock the PC when going for coffee, if they are aware of the consequences of not doing so.

Conclusion

The security policy is the mainstay of security, and the security practitioner must remain aware of the different issues to be addressed — legal, physical, systems, staff education. The security practitioner must not only be aware of the issues, but must also become a master of them.

References

1. Thomas R. Peltier, *Information Security Policies, Procedures, and Standards,* New York: Auerbach Publications, 2001.
2. Mark B. Desman, *Building an Information Security Awareness Program,* New York: Auerbach Publications, 2002.

77

Policy Development

Chris Hare, CISSP, CISA

This chapter introduces the reason why organizations write security policy. Aside from discussing the structure and format of policies, procedures, standards, and guidelines, this chapter discusses why policies are needed, formal and informal security policies, security models, and a history of security policy.

The Impact of Organizational Culture

The culture of an organization is very important when considering the development of policy. The workplace is more than just a place where people work. It is a place where people congregate to not only perform their assigned work, but to socialize and freely exchange ideas about their jobs and their lives.

It is important to consider this culture when developing policies. The more open an organization is, the less likely that policies with heavy sanctions will be accepted by the employees. If the culture is more closed, meaning that there is less communication between the employees about their concerns, policies may require a higher degree of sanctions. In addition, the tone, or focus, of the policy will vary from softer to harder.

Regardless of the level of communication, few organizations have their day-to-day operations precisely documented. This highly volatile environment poses challenges to the definition of policy, but it is even more essential to good security operations.

The History of Security Policy

Security policy is defined as the set of practices that regulate how an organization manages, protects, and assigns resources to achieve its security objectives. These security objectives must be tempered with the organization's goals and situation, and determine how the organization will apply its security objectives. This combination of the organization's goals and security objectives underlie the management controls that are applied in nearly all business practices to reduce the risks associated with fraud and human error.

Security polices have evolved gradually and are based on a set of security principles. While these principles themselves are not necessarily technical, they do have implications for the technologies that are used to translate the policy into automated systems.

Security Models

Security policy is a decision made by management. In some situations, that security policy is based on a security model. A security model defines a method for implementing policy and technology. The model is typically a mathematical model that has been validated over time. From this mathematical model, a policy is developed. When a model is created, it is called an informal security model. When the model has been mathematically validated, it becomes a formal model. The mathematics associated with the validation of the model is beyond the scope of this chapter, and will not be discussed. Three such formal security models are the Bell-LaPadula, Biba, and Clark-Wilson security models.

The Bell–LaPadula Model

The Bell–LaPadula, or BLP, model is a confidentiality-based model for information security. It is an abstract model that has been the basis for some implementations, most notably the U.S. Department of Defense (DoD) *Orange Book*. The model defines the notion of a secure state, with a specific transition function that moves the system from one security state to another. The model defines a fundamental mode of access with regard to read and write, and how subjects are given access to objects.

The secure state is where only permitted access modes, subject to object are available, in accordance with a set security policy. In this state, there is the notion of preserving security. This means that if the system is in a secure state, then the application of new rules will move the system to another secure state. This is important, as the system will move from one secure state to another.

The BLP model identifies access to an object based on the clearance level associated with both the subject and the object, and then only for read-only, read-write, or write-only access. The model bases access on two main properties. The *simple security property,* or *ss-property*, is for read access. It states that an object cannot read material that is classified higher than the subject. This is called "no read up." The second property is called the *star property,* or **-property,* and relates to write access. The subject can only write information to an object that is at the same or higher classification. This is called "no-write-down" or the "confinement property." In this way, a subject can be prevented from copying information from one classification to a lower classification.

While this is a good thing, it is also very restrictive. There is no discernment made of the entire object or some portion of it. Neither is it possible in the model itself to change the classification (read as downgrade) of an object.

The BLP model is a discretionary security model as the subject defines what the particular mode of access is for a given object.

The Biba Model

Biba was the first attempt at an integrity model. Integrity models are generally in conflict with the confidentiality models because it is not easy to balance the two. The Biba model has not been used very much because it does not directly relate to a real-world security policy.

The Biba model is based on a hierarchical lattice of integrity levels, the elements of which are a set of subjects (which are active information processing) and a set of passive information repository objects. The purpose of the Biba model is to address the first goal of integrity: to prevent unauthorized users from making modifications to the information.

The Biba model is the mathematical dual of BLP. Just as reading a lower level can result in the loss of confidentiality for the information, reading a lower level in the integrity model can result in the integrity of the higher level being reduced.

Similar to the BLP model, Biba makes use of the *ss-property* and the **-property*, and adds a third one. The *ss-property* states that a subject cannot access/observe/read an object of lesser integrity. The **-property* states that a subject cannot modify/write-to an object with higher integrity. The third property is the *invocation property*. This property states that a subject cannot send messages (i.e., logical requests for service) to an object of higher integrity.

The Clark–-Wilson Model

Unlike Biba, the Clark–Wilson model addresses all three integrity goals:

1. Preventing unauthorized users from making modifications
2. Maintaining internal and external consistency
3. Preventing authorized users from making improper modifications

Note: Internal consistency means that the program operates exactly as expected every time it is executed. External consistency means that the program data is consistent with the real-world data.

The Clark–Wilson model relies on the well-formed transaction. This is a transaction that has been sufficiently structured and constrained as to be able to preserve the internal and external consistency requirements. It also requires that there be a separation of duty to address the third integrity goal and external consistency. To accomplish this, the operation is divided into sub-parts, and a different person or process has responsibility for a single sub-part. Doing so makes it possible to ensure that the data entered is consistent with that information which is available outside the system. This also prevents people from being able to make unauthorized changes.

EXHIBIT 77.1 BLP and Biba Model Properties

Property	BLP Model	Biba Model
ss-property	A subject cannot read/access an object of a higher classification (no-read-up)	A subject cannot observe an object of a lower integrity level
*-property	A subject can only save an object at the same or higher classification (no-write-down)	A subject cannot modify an object of a higher integrity level
Invocation property	Not used	A subject cannot send logical service requests to an object of higher integrity

Exhibit 77.1 compares the properties in the BLP and Biba models.

These formal security models have all been mathematically validated to demonstrate that they can implement the objectives of each. These security models are only part of the equation; the other part is the security principles.

Security Principles

In 1992, the Organization for Economic Cooperation and Development (OECD) issued a series of guidelines intended for the development of laws, policies, technical and administrative measures, and education. These guidelines include:

1. *Accountability*. Everyone who is involved with the security of information must have specific accountability for their actions.
2. *Awareness*. Everyone must be able to gain the knowledge essential in security measures, practices, and procedures. The major impetus for this is to increase confidence in information systems.
3. *Ethics*. The method in which information systems and their associated security mechanisms are used must be able to respect the privacy, rights, and legitimate interests of others.
4. *Multidisciplinary principle*. All aspects of opinion must be considered in the development of policies and techniques. These must include legal, technical, administrative, organizational, operational, commercial, and educational aspects.
5. *Proportionality*. Security measures must be based on the value of the information and the level of risk involved.
6. *Integration*. Security measures should be integrated to work together and establish defensive depth in the security system.
7. *Timeliness*. Everyone should act together in a coordinated and timely fashion when a security breach occurs.
8. *Reassessment*. Security mechanisms and needs must be reassessed periodically to ensure that the organization's needs are being meet.
9. *Democracy*. The security of the information and the systems where it is stored must be in line with the legitimate use and information transfer of that information.

In addition to the OECD security principles, some addition principles are important to bear in mind when defining policies. These include:

10. *Individual accountability*. Individuals are uniquely identified to the security systems, and users are held accountable for their actions.
11. *Authorization*. The security mechanisms must be able to grant authorizations for access to specific information or systems based on the identification and authentication of the user.
12. *Least privilege*. Individuals must only be able to access the information that they need for the completion of their job responsibilities, and only for as long as they do that job.
13. *Separation of duty*. Functions must be divided between people to ensure that no single person can a commit a fraud undetected.
14. *Auditing*. The work being done and the associated results must be monitored to ensure compliance with established procedures and the correctness of the work being performed.

15. *Redundancy*. This addresses the need to ensure that information is accessible when required; for example, keeping multiple copies on different systems to address the need for continued access when one system is unavailable.
16. *Risk reduction*. It is impractical to say that one can completely eliminate risk. Consequently, the objective is to reduce the risk as much as possible.

There are also a series of roles in real-world security policy that are important to consider when developing and implementing policy. These roles are important because they provide distinctions between the requirements in satisfying different components of the policy. These roles are:

1. *Originator*: the person who creates the information
2. *Authorizer*: the person who manages access to the information
3. *Owner*: may or may not be a combination of the two previous roles
4. *Custodian*: the user who manages access to the information and carries out the authorizer's wishes with regard to access
5. *User*: the person who ultimately wants access to the information to complete a job responsibility

When looking at the primary security goals — confidentiality, integrity, and availability — security policies are generally designed around the first two goals, confidentiality and integrity. Confidentiality is concerned with the privacy of, and access to, information. It also works to address the issues of unauthorized access, modification, and destruction of protected information. Integrity is concerned with preventing the modification of information and ensuring that it arrives correctly when the recipient asks for it.

Often, these two goals are in conflict due to their different objectives. As discussed earlier, the Bell–LaPadula model addresses confidentiality, which, incidentally, is the objective of the Trusted Computing Standards Evaluation Criteria developed by the U.S. Department of Defense.

The goal of integrity is defined in two formal security models: Biba and Clark–Wilson. There is no real-world security policy based on the Biba model; however, the objectives of the European ITSEC criteria are focused around integrity.

Availability is a different matter because it is focused on ensuring that the information is always available when needed. While security can influence this goal, there are several other factors that can positively and negatively influence the availability of the information.

The Chinese Wall policy, while not a formal security model per se, is worth being aware of. This policy sees that information is grouped according to information classes, often around conflicts of interest. People frequently need to have access to information regarding a client's inside operations to perform their job functions. In doing so, advising other clients in the same business would expose them to a conflict of interest. By grouping the information according to information classes, the provider cannot see other information about its client. The Chinese Wall is often used in the legal and accounting professions.

However, the scope of security policy is quite broad. To be successful, the security policy must be faithfully and accurately translated into a working technical implementation. It must be documented and specified unambiguously; otherwise, when it is interpreted by human beings, the resulting automated system may not be correct. Henceforth, it is absolutely essential that the definition of the policy be as specific as possible. Only in this manner is it possible for the translation of security policy to an automated implementation to be successful.

In addition, several policy choices must be made regarding the computing situation itself. These include the security of the computing equipment and how users identify themselves. It is essential to remember that confidentiality and integrity are difficult to combine in a successful security policy. This can cause implementation problems when translating from the written policy to an automated system. The organization's real-world security policy must reflect the organization's goals.

The policy itself must be practical and usable. It must be cost-effective, meaning that the cost of implementing the policy must not be higher than the value of the assets being protected. The policy must define concrete standards for enforcing security and describe the response for misuse. It must be clear and free of jargon, in order to be understood by the users. Above all, the policy must have the support of the highest levels of senior management. Without this, even the best security policy will fail.

It is also very important that the policy seek the right balance between security and ease of use. If one makes it too difficult for the users to get their jobs done, then one negatively impacts business and forces the users to find ways around the security implementation. On the other hand, if one leans too much to ease of use, one may impact the organization's security posture by reducing the level of available security.

Why Does One Need Policy?

People have understood the need for security for a long time. Ever since an individual has had something of value that someone else wanted, they associated security with the need for the protection of that asset. Most people are familiar with the way that banks take care of our money and important documents by using vaults and safety deposit boxes. If the banks did not have policies that demonstrated how they implement appropriate protection mechanisms, the pubic would lose faith in them.

Security itself has a long history, and computers have only recently entered that history. People have installed locks on their doors to make it more difficult for thieves to enter, and people use banks and other technologies to protect their valuables, homes, and families. The military has long understood the need to protect its information from the enemy. This has resulted in the development of cryptography to encode messages so that the enemy cannot read them.

Many security techniques and policies are designed to prevent a single individual from committing fraud alone. They are also used to ensure supervisory control in appropriate situations.

The Need for Controls

Policy is essential for the people in the organization to know what they are to do. There are a number of different reasons for it, including legislative compliance, maintaining shareholder confidence, and demonstrating to the employee that the organization is capable of establishing and maintaining objectives.

There are a number of legal requirements that require the development of policies and procedures. These requirements include the duty of loyalty and the duty of care. The duty of loyalty is evident in certain legal concepts, including the duty of fairness, conflict of interest, corporate opportunity, and confidentiality. To avoid a conflict of interest situation, individuals must declare any outside relationships that might interfere with the enterprise's interests. In the duty of fairness, when presented with a conflict of interest situation, the individual has an obligation to act in the best interest of all affected parties.

When presented with material inside information such as advance notices on mergers, acquisitions, patents, etc., the individual will not use it for personal gain. Failing to do so results in a breach of corporate opportunity.

These elements have an impact should there be an incident that calls the operation into question. In fact, in the United States, there are federal sentencing guidelines for criminal convictions at the senior executive level, where the sentence can be reduced if there are policies and procedures that demonstrate due diligence. That means that having an effective compliance program in place to ensure that the corporation's policies, procedures, and standards are in place can have a positive effect in the event of a criminal investigation into the company.

For example, the basic functions inherent in most compliance programs

- Establish policies, procedures, and standards to guide the workforce.
- Appoint a high-level manager to oversee compliance with the policies, procedures, and standards.
- Exercise due care when granting discretionary authority to employees.
- Ensure that compliance policies are being carried out.
- Communicate the standards and procedures to all employees.
- Enforce the policies, standards, and procedures consistently through appropriate disciplinary measures.
- Implement procedures for corrections and modification in case of violations.

The third element from a legal perspective is the Economic Espionage Act of 1996 in the United States. The EEA, for the first time, makes the theft of trade secret information a federal crime, and subjects criminals to penalties including fines, imprisonment, and forfeiture. However, the EEA also expects that the organization who owns the information is making reasonable efforts to protect that information.

In addition to the legal requirements, there are also good business reasons for establishing policies and procedures. It is a well-accepted fact that it is important to protect the information that is essential to an organization, just like it is essential to protect the financial assets.

This means that there is a need for controls placed on the employees, vendors, customers, and other authorized network users. With growing requirements to be able to access information from any location on the globe, it is necessary to have organizationwide set of information security policies, procedures, and standards in place.

With the changes in the computing environment from host-based to client/server-based systems, the intricacies of protecting the environment have increased dramatically. The bottom line then is that good controls make good business sense. Failing to implement good policies and procedures can lead to a loss in shareholder and market confidence in the company should there be an incident that becomes public.

In writing the policies and procedures, it is necessary to have a solid understanding of the corporation's mission, values, and business operations. Remember that policies and procedures exist to define and establish the controls required to protect the organization, and that security for security's sake is of little value to the corporation, its employees, or the shareholders.

Searching for Best Practices

As changes take place and business develops, it becomes necessary to review the policy and ensure that it continues to address the business need. However, it is also advisable for the organization to seek out relationships with other organizations and exchange information regarding their best practices. Continuous improvement should be a major goal for any organization. The review of best industry practices is an essential part of that industry improvement, as is benchmarking one organization against several others.

One organization may choose to implement particular policies in one way, while another does it in a completely different fashion. By sharing information, security organizations can improve upon their developed methods and maintain currency with industry.

There are a number of membership organizations where one can seek opinions and advice from other companies. These include the Computer Security Institute Public Working forums and the International Information Integrity Institute (I-4). There are other special-interest groups hosted by engineering organizations, such as the Association for Computing Machinery (ACM).

As in any situation, getting to that best practice, whether it be the manufacturing of a component or the implementation of a security policy, takes time.

Management Responsibilities

In the development and implementation of policy, management has specific responsibilities. These include a clear articulation of the policy, being able to live up to it themselves, communicating policy, and providing the resources needed to develop and implement it. However, management is ultimately responsible to the legislative bodies, employees, and shareholders to protect the organization's physical and information assets. In doing so, management has certain legal principles that it must uphold in the operation of the organization and the development of the policies that will govern how the organization works.

Duty of Loyalty

Employees owe to their employers a legal duty of honesty, loyalty, and utmost good faith, which includes the avoidance of conflict of interest and self-interest. In carrying out the performance of their day-to-day responsibilities, empoyees are expected to act at all times in their employers' best interest unless the responsibility is unlawful. Any deviation from this duty that places an employee's interest above the employer's can be considered a breach of the employee's duty of care, loyalty, or utmost good faith. Fiduciary employees will owe a higher standard of care than ordinary employees.

If a manager knows that an employee may be putting his or her own interest above that of the employer's, it is incumbent upon the manager to warn the employee, preferably in writing, of the obligation to the employer. The manager should also advise the employer of the situation to prevent her or him from also being held accountable for the actions of the employee.

Conflict of Interest

Conflict of interest can be defined as an individual who makes a decision with the full knowledge that it will benefit some, including himself, and harm others. For example, the lawyer who knowingly acts on behalf of two parties who are in conflict with each other, is a conflict of interest.

Duty of Care

The duty of care is where the officers owe a duty to act carefully in fulfilling the important tasks assigned to them. For example, a director shall discharge his or her duties with the care and prudence an ordinary person would exercise in similar circumstances, and in a manner that he or she believe is in the best interests of the enterprise.

Furthermore, managers and their subordinates have a responsibility to provide for systems security and the protection of any electronic information stored therein, even if they are not aware of this responsibility. This comes from the issue of negligence, as described in the Common Law of many countries.

Even if the organization does cause a problem, it may not be held fully responsible or liable. Should the organization be able to demonstrate that it:

- Took the appropriate precautions,
- Employed controls and practices that are generally used,
- Meets the commonly desired security control objectives,
- Uses methods that are considered for use in well-run computing facilities
- Used common sense and prudent management practices,

then the organization will be said to have operated with due care, as any other informed person would.

Least Privilege

Similar to its counterpart in the function role, the concept of least privilege means that a process has no more privilege than what it really needs in order to perform its functions. Any modules that require "supervisor" or "root" access (i.e., complete system privileges) are embedded in the kernel. The kernel handles all requests for system resources and permits external modules to call privileged modules when required.

Separation of Duties/Privilege

Separation of duties is the term applied to people, while separation of privilege is the systems equivalent. Separation of privilege is the term used to indicate that two or more mechanisms must agree to unlock a process, data, or system component. In this way, there must be agreement between two system processes to gain access.

Accountability

Accountability is being able to hold a specific individual responsible for his or her actions. To hold a person accountable, it must be possible to uniquely and effectively identify and authenticate that person. This means that an organization cannot hold an individual responsible for his or her actions if that organization does not implement a way to uniquely identify each individual. There are two major themes: (1) the identification and authentication of that individual when the user accesses the system; and (2) the validation that the individual initiated or requested a particular transaction.

Management Support for Policy

Management support is critical to the success of any initiative, be it the development of a new product or service, or the development of a policy. If senior management does not approve the intent behind the activity, then it will not be successful. This is not restricted to the development of the organization's security policy, but any activity. However, security policy can both raise and address significant issues in any organization. Obtaining management support is often the most difficult part of the planning process.

Planning for Policy

Planning and preparation are integral parts of policy, standards, and procedure development, but are often neglected. Included in the preparation process is all of the work that must be done. Policy lays out the general

requirements to take; the standards define the tools that are to be used; and the procedures provide employees with the step-by-step instructions to accomplish it.

Well-written procedures never take the place of supervision, but they can take some of the more mundane tasks and move them out to the employees. Employees use policy to provide information and guidance in making decisions when their managers are not available. The policy should identify who is responsible for which activity.

An effective set of policies can actually help the organization achieve two key security requirements: separation of duties and rotation of assignments. No single individual should have complete control over a complete process from inception to completion. This is an element in protecting the organization from fraud.

Planning during policy development must include attention to security principles. For example, individuals who are involved in sensitive duties should be rotated through other assignments on a periodic basis. This removes them from sensitive activities, thereby reducing their attractiveness as a target. Rotation of duties can also provide other efficiencies, including job efficiency and improvement. The improvement aspect is achieved as the result of moving people through jobs so that they do not develop short-cuts, errors creeping into the work, or a decrease in quality.

Once the policies are established, it is necessary to define the standards that will be used to support those policies. These standards can include hardware, software, and communications protocols to who is responsible for approving them.

There is no point in progressing through these steps unless there is a communication plan developed to get the information out to the employees and others as appropriate. This is particularly important because management does not have the luxury of sitting down with every employee and discussing his or her responsibility. However, management does have a responsibility to communicate to every user in an ongoing fashion about the contents of the policy and the employee's responsibilities in satisfying it.

The ability to provide the information to the employees is an essential part of the development of the policies, standards, and procedures. Through these vehicles, the employees will understand how they should perform their tasks in accordance with the policies.

Part of the planning process involves establishing who will write the policies and related documents, who will review them, and how agreement on the information contained is reached. For example, there are a number of experts who are consulted when establishing how management's decision will be written to allow for subsequent implementation. These same experts work with writers, management, and members from the community of interest to ensure that the goals of the policy are realistic and achievable. In addition to these people who effectively write the policy, additional resources are required to ensure that the policies are reasonable. For example, Human Resources and Legal are among the other specialists who review the policy.

The Policy Management Hierarchy

There are essentially five layers in the policy management hierarchy. These are illustrated in Exhibit 77.2.

Legislation has an impact on the organization regardless of its size. The impact ranges from revenue and taxation, to handling export-controlled material. Legislation is established by government, which in turn often creates policy that may or may not be enacted in legislation.

The second layer — policy — references the policy that is developed by the organization and approved by senior management and describes its importance to the organization. Standards are derived from the policy. The standard defines specific, measurable statements that can be used to subsequently verify compliance.

The fourth layer — procedures — consists of step-by-step instructions that explain what the user must do to implement the policy and standards. The final layer — guidelines — identifies things that the organization would like to see its members do. These are generally recommendations; and while the standards are mandatory, guidelines are optional.

There may be one additional layer, which is inserted between the standards and the procedures. This layer addresses practices, which can be likened to a process. The standard defines what must be done; the practice defines why and how; while the procedures provide specific step-by-step instructions on the implementation. These documents are discussed later in this chapter, including their format and how to go about writing them.

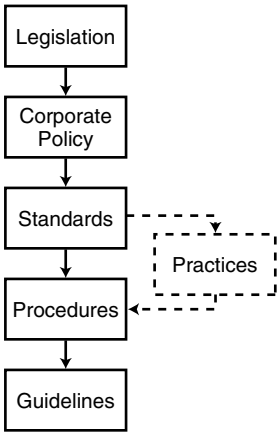


EXHIBIT 77.2 Policy management hierarchy.

The Types of Policy

There are three major classifications of policy, one of which has been discussed: regulatory, advisory, and informative. It is also important to note that an organization can define specific policies applicable to the entire organization, while individual departments may provide policy for themselves.

Regulatory

Regulatory policy is not often something that an organization can work around. Rather, they must work with them. Governments and regulatory and governing bodies that regulate certain professions, such as medicine and law, typically create this type of policy. In general, organizations that operate in the public interest, such as safety or the management of public assets, or that are frequently held accountable to the public for their actions, are users of regulatory policy.

This type of policy consists of a series of legal statements that describe in detail what must be done, when it must be done, who does it, and can provide insight as to why it is important to do it. Because large numbers of groups use these policies, they share the use and interpretation of these policies for their organizations. In addition to the common objectives of confidentiality, integrity, and availability (CIA), there are two premises used to establish regulatory policy.

The first is to establish a clearly consistent process. This is especially true for organizations involved with the general public, and they must show the uniformity with how regulations are applied without prejudice. Second, the policy establishes the opportunity for individuals who are not technically knowledgeable in the area to be sure that the individuals who are responsible are technically able to perform the task.

Regulatory policies often have exclusions or restrictions regarding their application. Frequently, regulatory policies are not effective when people must make immediate decisions based on the facts before them. This is because many situations present many different outcomes. Establishing a policy that is capable of addressing all possible outcomes results in a policy that is highly complex, difficult to apply, and very difficult to enforce.

Advisory

An advisory policy provides recommendations often written in very strong terms about the action to be taken in a certain situation or a method to be used. While this appears to be a contradiction of the definition of policy, advisory policy provides recommendations. It is aimed at knowledgeable individuals with information to allow them to make decisions regarding the situation and how to act.

Because it is an advisory policy, the enforcement of this policy is not applied with much effort. However, the policy will state the impact for not following the advice that is provided within the policy. While the specific

impacts may be stated, the policy provides informed individuals with the ability to determine what the impacts will be should they choose an alternate course of action.

The impacts associated with not following the policy can include:

- Omission of information that is required to make an informed decision
- Failure to notify the correct people who are involved in making the decision or complete the process
- Missing important deadlines
- Lost time in evaluating and discussing the alternatives with auditors and management

It is important to consider that the risks associated with not following the advisory policy can be significant to the organization. The cost of lost productive time due to the evaluation of alternatives and discussions alone can have a significant impact on the organization, and on determining the validity and accuracy of the process.

Advisory policies often have specific restrictions and exclusions. For example, the advisory policy may set out that latitude in determining the course of action can only be extended to experienced individuals, while less-experienced persons must follow the policy as defined, with little opportunity for individual decision making. It is also important that any exceptions to the policy be documented and what is to be done when those situations are encountered.

Informative

The third type of policy is informative in nature, the purpose of which is to communicate information to a specific audience. That audience is generally any individual who has the opportunity or cause to read the policy. This policy implies no actions or responsibilities on the part of the reader and no penalty is imposed for not following the policy.

Although informative policies typically carry less importance than regulatory or advisory policies, they can carry strong messages about specific situations to the audience. Due to the wide audience intended for informational policies, references to other, more specific policies are made to provide even more information. This means that the distribution of the informative policies can be conducted with little risk to the organization, keeping policies that contain more sensitive information for a limited distribution.

Corporate versus Departmental

The only difference between corporate and departmental policy is the scope. For example, the organization may specify policy regarding how customer interactions will be handled. Specific organizations may choose to define policy about how to handle customer interactions specific to that department. There is no other difference other than the corporate or organizational policy applies to the entire organization, while departmental policy is specific to only that department. With the scope being narrowed, the process of reviewing and approving the policy can be much shorter due to the reduced number of people that must review it and express their opinions about it.

Program versus Topic Policy

Aside from these major policy types, it is important to make the distinction between program and topic policy. Program policy is used to create an organization's overall security vision, while topic-specific policies are used to address specific topics of concern. In addition to the topic policies are application-specific policies that are used to protect specific applications or systems.

Writing Policy

Having examined the different types of policy, the importance of management support and communication of the new policy, and why policy is needed in an organization, we now turn to the process of writing policy for the organization.

Exhibit 77.3 Reviewing Principles while Developing Policies

Policy Statement	Principle 1	Principle 2
Entire policy statement	If this principle applies, then put an X in this column.	If this principle applies, then put an X in this column.

Topics

Every organization must develop a basic set of policies. These can normally be found as a document prepared by the organization and can be used by an information security professional to reinforce the message as needed. Policy is the result of a senior management decision regarding an issue. Consequently, there is a wide range of topics available. These include:

1. Shared beliefs
2. Standards of conduct
3. Conflict of interest
4. Communication
5. Electronic communication systems
6. Internet security
7. Electronic communication policy
8. general security policy
9. Information protection policy
10. Information classification

This is not an all-inclusive list, but is intended to identify those areas that are frequently targeted as issues. It is not necessary to identify all of the policy topic areas before getting started on the development. It is highly likely that one policy may make reference to another organizational policy, or other related document.

There is a specific format that should be used in any policy, but it is important that if there are already policies developed in an organization, one must make the new policies resemble the existing ones. This is important to ensure that when people read them, they see them as policy. If a different style is used, then it is possible that the reader might not associate them with policy, despite the fact that it is identified as a policy.

The Impact of Security Principles on Policy Development

The organization should select some quantity of security principles that are important to it. When developing policies and related documents, the chosen principles should be reconsidered from time to time, and a review of the correlation of the policy (or standard, procedure, and guidelines) to the chosen principles should be performed. This can easily be done through the implementation of a matrix as shown in Exhibit 77.3.

In the matrix, the desired principles are listed across the top of the matrix, and the policy statements are listed down the left-hand column. An "X" is marked in the appropriate columns to illustrate the relationship between the principle and the policy statement. By correlating the principles to the policy (or policy components), the policy writer can evaluate their success. This is because the principles should be part of the objectives or mission of the organization. If there is a policy or component that does not address any principles, then that policy or component should be reviewed to see if it is really necessary, or if there is a principle that was not identified as required. By performing this comparison, the policy writer can make changes to the policy while it is under development, or make recommendations to senior management regarding the underlying principles.

Policy Writing Techniques

When writing the policy, it is essential that the writer consider the intended audience. This is important because a policy that is written using techniques that are not understood by the intended audience will result in confusion and misinterpretation by that audience.

Language

Using language that is appropriate to the intended audience is essential. The language must be free of jargon and as easy to understand as possible. The ability of the user community to understand the policy allows them

to determine what their responsibilities are and what they are required to do to follow the policy. When the policy is written using unfamiliar language, misinterpretations regarding the policy result.

Focus

Stay focused on the topic that is being addressed in the policy. By bringing in additional topics and issues, the policy will become confusing and difficult to interpret. An easy rule of thumb is that for each major topic, there should be one policy. If a single policy will be too large (i.e., greater than four pages), then the topic area should be broken down into sub-topics to ensure that it focuses on and covers the areas intended by management.

Format

Policy is the cornerstone of the development of an effective information security architecture. The policy statement defines what the policy is, and is often considered the most effective part of the policy. The goal of an information security policy is to maintain the integrity, confidentiality, and availability of information resources. The basic threats that can prevent an organization from reaching this goal include theft, modification, destruction, or disclosure, whether deliberate or accidental.

The term "policy" means different things to different people. Policy is management's decision regarding an issue. Policy often includes statements of enterprise beliefs, goals, and objectives, and the general means for their attainment in a specified subject area.

A policy statement itself is brief and set at a high level. Because policies are written at a high level, supporting documentation must be developed to establish how employees will implement that policy. Standards are mandatory activities, actions, rules, or regulations that must be performed in order for the policy to be effective.

Guidelines, while separate documents and not included in the policy, are more general statements that provide a framework on which procedures are based. While standards are mandatory, guidelines are recommendations. For example, an organization could create a policy that states that multi-factor authentication must be used, and in what situations. The standard defines that the acceptable multi-factor authentication tools include specific statements regarding the accepted and approved technologies.

Remember that policies should:

1. Be easy to understand
2. Be applicable
3. Be do-able
4. Be enforceable
5. Be phased in
6. Be proactive
7. Avoid absolutes
8. Meet business objectives

Writing policy can be both easy and difficult at the same time. However, aside from working with a common policy format, the policy writer should remember the attributes that many journalists and writers adhere to:

- *What*. What is the intent of the policy?
- *Who*. Who is affected? What are the employee and management responsibilities and obligations?
- *Where*. Where does the policy apply? What is the scope of the policy?
- *How*. What are the compliance factors, and how will compliance be measured?
- *When*. When does the policy take effect?
- *Why*. Why is it necessary to implement this policy?

In considering the policy attributes, it is easier for the policy writer to perform a self-evaluation of the policy before seeking reviews from others. Upfront self-assessment of the policy is critical. By performing the self-assessment, communication and presentation of the policy to senior management will be more successful. Self-assessment can be performed in a number of ways, but an effective method is to compare the policy against the desired security principles.

It is important for the policy writer to ascertain if there are existing policies in the organization. If so, then any new policies should be written to resemble the existing policies. By writing new policies in the existing

format, organization members will recognize them as policies and not be confused or question them because they are written in a different format.

A recommended policy format includes the following headings:

- *Background*: why the policy exists
- *Scope*: who the policy affects and where the policy is required
- *Definitions*: explanations of terminology
- *References*: where people can look for additional information
- *Coordinator/Policy Author*: who sponsored the policy, and where do people go to ask questions
- *Authorizing Officer*: who authorized the policy
- *Effective Date*: when the policy takes effect
- *Review Date*: when the policy gets reviewed
- *Policy Statements*: what must be done
- *Exceptions*: how exceptions are handled
- *Sanctions*: what actions are available to management when a violation is detected

While organizations will design and write their policies in a manner that is appropriate to them, this format establishes the major headings and topic areas within the policy document. The contents of these sections are described later in this chapter in the section entitled "Establishing a Common Format."

Defining Standards

Recall that a standard defines what the rules are to perform a task and evaluate its success. For example, there is a standard that defines what an electrical outlet will look like and how it will be constructed within North America. As long as manufacturers follow the standard, they will be able to sell their outlets; and consumers will know that if they buy them, their appliances will fit in the outlet.

The definition of a standard is not easy because implementation of a standard must be validated regularly to ensure that compliance is maintained. Consider the example of an electrical outlet. If the manufacturing line made a change that affected the finished product, consumers would not be able to use the outlet, resulting in lost sales, increased costs, and a confused management, until the process was evaluated against the standards.

Consequently, few organizations actually create standards unless specifically required, due to their high implementation and maintenance costs.

A recommended format for standards documents includes the following headings:

- *Background*: why the standard exists
- *Scope*: who requires the standard and where is it required
- *Definitions*: explanations of terminology
- *References*: where people can look for additional information
- *Coordinator/Standards Author*: who sponsored the standard, and where do people go to ask questions
- *Authorizing Officer*: who authorized the standard
- *Effective Date*: when the standard takes effect
- *Review Date*: when the standard gets reviewed
- *Standards Statements*: what the measures and requirements are

While organizations will design and write their standards in a manner that is appropriate to them, this format establishes the major headings and topic areas within the policy document.

It is important to emphasize that while the standard is important to complete, its high cost of implementation maintenance generally means that the lifetime, or review date, is at least five years into the future.

Defining Procedures

Procedures are as unique as the organization. There is no generally accepted approach to writing a procedure. What will determine how the procedures look in the organization is either the standard that has been developed

previously or an examination of what will work best for the target audience. It can be said that writing the procedure(s) is often the most difficult part, due to the amount of detail involved.

Due to the very high level of detail involved, writing a procedure often requires more people than writing the corresponding documents. Consequently, the manager responsible for the development of the procedure must establish a team of experts, such as those people who are doing the job now, to document the steps involved. This documentation must include the actual commands to be given, any arguments for those commands, and what the expected outcomes are.

There are also several styles that can be used when writing the procedure. While the other documents are written to convey management's desire to have people behave in a particular fashion, the procedure describes how to actually get the work done. As such, the writer has narrative, flowchart, and play script styles from which to choose.

The narrative style presents information in paragraph format. It is conversational and flows nicely, but it does not present the user with easy-to-follow steps. The flowchart format provides the information in a pictorial format. This allows the writer to present the information in logical steps. The play script style, which is probably used more than any other, presents step-by-step instructions for the user to follow.

It is important to remember that the language of the procedure should be written at a level that the target audience will be able to understand. The key procedure elements as discussed in this chapter are identifying the procedure needs, determining the target audience, establishing the scope of the procedure, and describing the intent of the procedure.

A recommended format for procedure documents includes the following headings:

- *Background*: why the procedure exists, and what policy and standard documents it is related to
- *Scope*: who requires the procedure and where is it required
- *Definitions*: explanations of terminology
- *References*: where people can look for additional information
- *Coordinator/Procedure Author*: who sponsored the procedure, and where do people go to ask questions
- *Effective Date*: when the procedure takes effect
- *Review Date*: when the standard gets reviewed
- *Procedure Statements*: what the measures and requirements are

While organizations will design and write their procedures in a manner that is appropriate to them, this format establishes the major headings and topic areas within the policy document.

Defining Guidelines

Guidelines, by their very nature, are easier to write and implement. Recall that a guideline is a set of nonbinding recommendations regarding how management would like its employees to behave. Unlike the other documents that describe how employees must perform their responsibilities, employees have the freedom to choose what guidelines, if any, they will follow. Compliance with any guideline is totally optional.

Policy writers often write the guidelines as part of the entire process. This is because as they move through the documents, there will be desired behaviors that cannot be enforced, but are still desired nonetheless. These statements of desired behavior form the basis for the guidelines.

Similar to the other documents, a recommended format for guideline documents includes the following headings:

- *Background*: why the guideline exists, and what policy and standard documents it is related to
- *Scope*: who requires guidelines and where are they required
- *Definitions*: explanations of terminology
- *References*: where people can look for additional information
- *Coordinator/Guidelines Author*: who sponsored the guidelines, and where do people go to ask questions
- *Effective Date*: when the standard guidelines take effect
- *Review Date*: when the standard guidelines get reviewed
- *Standards Statements*: what the measures and requirements are

Unlike the other documents, it is not necessary to have an approver for a guideline. As it is typically written as part of a larger package, and due to its nonbinding nature, there is no approving signature required.

Publishing the Policy

With the documents completed, they must be communicated to the employees or members of the organization. This is done through an employee policy manual, departmental brochures, and online electronic publishing. The success of any given policy is based on the level of knowledge that the employees have about it. This means that employees must be aware of the policy. For this to happen, the organization must have a method of communicating the policy to the employees, and keeping them aware of changes to the policy in the future.

Policy Manual

Organizations have typically chosen to create policy manuals and provide a copy to each individual. This has been effective over time because the policies were immediately available to those who needed to refer to them. However, other problems, such as maintenance of the manuals, became a problem over time. As new updates were created, employees were expected to keep their manuals updated. Employees would receive the updated manual, but due to other priorities would not keep their manuals up-to-date. This resulted in confusion when an issue arose that required an examination of policy.

Even worse, organizations started to see that the high cost of providing a document for each member of the organization was having a negative effect on their profit lines. They began to see that they were getting little value from their employees for the cost of the manuals. Consequently, organizations began to use electronic publishing of their policies as their communication method.

Departmental Brochures

Not all policies are created for the entire organization. Individual department also had to create policies that affected their individual areas. While it was possible to create a policy manual for the department, it was not practical from an expense perspective. Consequently, departments would create a brochure with the policies that pertained only to their area.

Putting the Policy Online

With the growth of the personal computer and the available access to the information online, more and more organizations have turned to putting the policies online. This has allowed for increased speed in regard to getting new policies and updates communicated to employees.

With the advent of the World Wide Web as a communication medium, organizations are using it as *the* method of making policies available. With hyperlinks, they can link to other related documents and references.

Awareness

However, regardless of the medium used to get the information and policies to the employees, they must be made aware of the importance of remaining up-to-date with the policies that affect them. And even the medium must be carefully selected. If all employees do not have access to a computer, then one must provide the policies in printed form as well. An ongoing awareness program is required to maintain the employee's level of knowledge regarding corporate policies and how they affect the employee.

Establishing a Common Format

A common format makes it easier for readers to understand the intent of the policy and its supporting documents. If there have been no previous written policies or related documents, creating a common format will be simple. If there is an existing format used within an organization, it becomes more difficult. However, it is essential that the writer adapt the layout of written documents to match that which is already in use. Doing so will ensure that the reader recognizes the document for what it is, and understands that its contents are sanctioned by the organization. The format and order of the different sections was presented earlier in the chapter, but is repeated here for conciseness:

- *Background* (all)
- *Scope* (all)

- *Definitions* (all)
- *References* (all)
- *Coordinator/Document Author* (all)
- *Authorizing Officer* (policy, standard, procedure)
- *Effective Date* (all)
- *Review Date* (all)
- *Disposal* (all)
- *Document Statements* (all)
- *Exceptions* (policy)
- *Sanctions* (policy)

Each of these sections should appear in the document unless otherwise noted. There are sections that can be considered as part of one document, while not part of another. To retain consistency, it is recommended that they appear in the order listed throughout all the documents.

In the following chapter sections, the term "document" is used to mean either a policy, standard, procedure, or guideline.

Background

It is important that the document include a statement providing some information on what has prompted the creation of the document. In the case of a new policy, what prompted management's decision, as new policy is generally created as a reaction to some particular event. The other documents would indicate that it references the new policy and why that document is required to support the new policy. By including the background on the situation in the document, one provides a frame of reference for the reader.

Scope

In some situations, the document is created for the benefit of the entire corporation, while others are applicable to a smaller number of people. It is important that the scope define where the document is applicable to allow people to be able to determine if the policy is applicable to them.

Definitions

It is essential that the documents, with the exception of the procedure, be as free as possible from technical jargon. Within documents other than the procedure, technical jargon tends to confuse the reader. However, in some situations, it is not possible to prevent the use of this terminology. In those situations, the effectiveness of the document is improved by providing explanations and definitions of the terminology.

Reference

Any other corporate documentation, including other policies, standards, procedures, and guidelines, that provides important references to the document being developed should be included. This establishes a link between the policy and other relevant documents that may support this policy, or that this policy may support.

If creating the document as an HTML file for publishing on the Web, then it is wise to include hyperlinks to the other related documentation.

Coordinator/Author

The coordinator or author is the sponsor who developed and sought approval for the document. The sponsor is identified in the policy document to allow any questions and concerns to be addressed to the sponsor. However, it is also feasible that the policy author is not the coordinator identified in the policy. This can occur when the policy has been written by a group of people and is to be implemented by a senior manager.

Authorizing Officer

Because senior management is ultimately responsible for the implementation of policy, it is important that a member of that senior management authorize the policy. Often, the senior executive who accepts responsibility is also responsible for the area concerned. For example, the Chief Information Officer will assume responsibility for information systems policies, while the Chief Financial Officer assumes responsibility for financial policies.

If the standard is to be defined as a corporate standard, then the appropriate member of senior management should authorize the standard. If the standard is for one department's use, then the senior manager of that department approves it. Procedures are generally only for a department and require a senior manager's approval.

Guidelines do not need approval unless they are for implementation within the company. In such situations, the senior manager responsible for the function should approve them.

Effective Date

This is the date when the document takes effect. When developing policy, it is essential that support be obtained for the policy, and sufficient time for user education be allowed before the policy takes effect. The same is true for the supporting documents, because people will want access to them when the policy is published.

Review Date

The review date establishes when the document is to be reviewed in the future. It is essential that a review period be established because all things change with time. Ideally, the document should make a statement that establishes a time period and whenever circumstances or events warrant a review. By establishing a review date, the accuracy and appropriateness of the document can be verified.

Disposal

In the event that the document is classified or controlled in some manner within the organization, then specific instructions regarding the disposal are to be indicated in this section. If there are no specific instructions, the section can be omitted, or included with a statement indicating that there are no special instructions.

Document Statement(s)

The policy statement typically consists of several text lines that describe what management's decision was. It is not long, and should be no more than a single paragraph. Any more than that, and the policy writer runs the risk of injecting ambiguity into the policy. However, the policy statements are to be clear enough to allow employees to determine what the required action is.

Statements within a standard must be of sufficient length to provide the detail required to convey the standard. This means that the standard can be quite lengthy in some situations.

Procedure statements are also quite detailed as they provide the exact command to be executed, or the task to be performed. Again, these can be quite lengthy due to the level of detail involved.

Exceptions

This section is generally included only in policy documents. It is advisable to include in the policy document a statement about how exceptions will be handled. One method, for example, is to establish a process where the exception is documented, an explanation provided about why an exception is the most practical way to handle the situation. With this done, the appropriate management is identified and agreement is sought, where those managers sign the exception. Exceptions should have a specific lifetime; for example, they should be reviewed and extended on an annual basis.

Violations and Sanctions

This section is generally included only in policy documents. The tendency is for organizations to sacrifice clarity in the policy for sanctions. The sanctions must be broad enough to provide management with some flexibility when determining what sanction is applied. For example, an organization would not dismiss an employee for a minor infraction. It is necessary that Human Resources and Legal review and approve the proposed sanctions.

Using a Common Development Process

A common process can be used in the creation of all these documents. The process of creating them is often managed through a project management approach if the individual writing them requires a number of other people to be involved and must coordinate their time with other projects. While it is not necessary, using this process in conjunction with a project management approach can ensure that management properly supports the document writing effort. One example of a process to use in defining and developing these documents consists of several phases as seen in Exhibit 77.4. Each of these development phases consists of discrete tasks that must be completed before moving on to the next one.

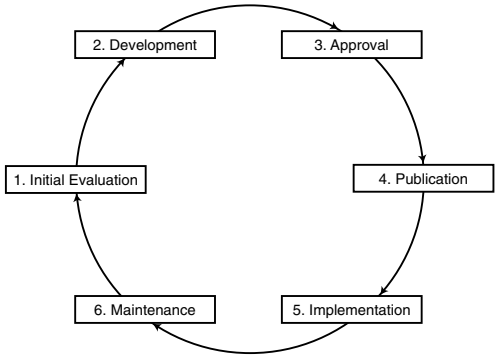


EXHIBIT 77.4 Defining and developing documents.

Phase One: Initial and Evaluation Phase

A written proposal to management is submitted that states the objectives of the particular document (policy, standard, etc.) and the need it is supposed to address. Management will then evaluate this request to satisfy itself that the expected benefit to the organization justifies the expected cost. If it does, then a team is assembled to develop and research the document as described in Phase Two. Otherwise, the submitter is advised that no further action will take place.

Phase Two: Development Phase

In the development phase, funding is sought from the organization for the project. The organization can choose to assemble a new team, or use one that was previously used for another project. The team must work with management to determine who will be responsible for approving the finished document.

The structure of the team must be such that all interested parties (stakeholders) are represented and the required competency exists. The team should include a representative from management, the operations organization responsible for implementation (if appropriate), the development team, a technical writer, and a member of the user community that will ultimately be a recipient of the service or product.

By including a representative from management, they can perform liaison duties with the rest of the organization's management, legal, and other internal organizations as required. The development team is essential to provide input on the requirements that are needed when the product or service is being developed or assembled into the finished product. Operations personnel provide the needed input to ensure that the document can actually be put into practice once it is completed. The user community cannot be ignored during the development phase. If they cannot accept the terms of the document, having their input upfront rather than later can shorten the development process. Finally, the technical writer assists in the creation of the actual language used in the document. While most people feel they can write well, the technical writer has been trained in the use of language.

Remember that unless the members of this team have these roles as their primary responsibility, they are all volunteers. Their reward is the knowledge that they have contributed to the content of the standard and the recognition of their expertise by virtue of having their names published in the document.

This team is the heart of the development process. The technical requirements are put forward, designed, and worded by the experts on the team. These people discuss and debate the issues until final wording is agreed upon. Consensus is the key, as unanimity is not often achieved.

As the draft is developed through a number of iterations and approaches the original design objectives, it is made available to the general population within the organization for review and comment. The review period generally lasts 30 days and allows for input from those outside the team.

During this review period, the document should be tested in a simulated exercise. For example, if the document being developed is a procedure, then a less-experienced person should be able to successfully perform the tasks based on the information within the procedure. If they cannot, then there is a deficiency that must be addressed prior to approval.

After the comments have been deliberated by the team and it feels that the document is technically complete, it moves on to Phase Three.

Phase Three: Approval Phase

When the team has completed the design phase, the document is presented to the appropriate body within the organization. Some organizations will have formalized methods for approving policy, while others will not. It is necessary during the development phase to establish who the approving body or person is.

The document is presented to the approving body and a discussion of the development process ensues, highlighting any reasons that the team felt were important considerations during development. The document is "balloted" by the approving body, and any negative issues should be addressed prior to approval of the document.

Phase Four: Publication Phase

Finally, the document is translated (if required) and published within the organization. At this point, the document is ready for implementation as of the effective date. In some situations, the effective date may be the date of publication.

Phase Five: Implementation

During implementation, the various groups affected by the new document commence its implementation. This implementation will be different, depending on where it is being placed into use. For example, a user's perspective will be different from that of an operational team. While the document is being used, people should be encouraged to send their comments and questions to the coordinator. These comments will be important during the review or maintenance phase.

Phase Six: Maintenance Phase

As decided during the development phase, the document is reviewed on the review date. During this review, the continuing viability of the document is decided. If the document is no longer required, then it is withdrawn or cancelled. If viability is determined and changes are needed, the team jumps into the development cycle at Phase Two and the cycle begins again.

Summary

This chapter has examined why policy is important to information security and some issues and areas concerning the development of that policy. Information Security Policy establishes what management wants done to protect the organization's intellectual property or other information assets. Standards are used to establish a common and accepted measurement that people will use to implement this policy. Procedures provide the details — the how of the implementation — while guidelines identify the things that management would like to see implemented.

Policy is an essential and important part of any organization because it identifies how the members of that organization must conduct themselves. To the information security manager, policy establishes what is important to the organization and what defines the shape of the work that follows.

References

1. Peltier, Thomas, *Information Security Policies, A Practitioner's Guide,* Auerbach Publications, 1999.
2. Kovacich, Gerald, *Information Systems Security Officer's Guide,* Butterworth-Heinemann, 1998.

Risk Analysis and Assessment

Will Ozier

There are a number of ways to identify, analyze, and assess risk and there is considerable discussion of "risk" in the media and among information security professionals. But, there is little real understanding of the process and metrics of analyzing and assessing risk. Certainly everyone understands that "taking a risk" means "taking a chance," but a risk or chance of what, is often not so clear.

When one passes on a curve or bets on a horse, one is taking a chance of suffering harm/injury or financial loss — an undesirable outcome. We usually give a degree of more or less serious consideration to such an action before taking the chance, so to speak. Perhaps we would even go so far as to calculate the odds (chance) of experiencing the undesirable outcome and, further, take steps to reduce the chance of experiencing the undesirable outcome.

To effectively calculate the chance of experiencing the undesirable outcome, as well as its magnitude, one must be aware of and understand the elements of risk and their relationship to each other. This, in a nutshell, is the process of risk analysis and assessment.

Knowing more about the risk, one is better prepared to decide what to do about it — accept the risk as now assessed (go ahead and pass on the blind curve or make that bet on the horses), or mitigate the risk. To mitigate the risk is to do something to reduce the risk to an acceptable level (wait for a safe opportunity to pass or put the bet money in a savings account with interest).

There is a third choice, to transfer the risk, i.e., buy insurance. However prudent good insurance may be, all things considered, having insurance will not prevent the undesirable outcome. Having insurance will only serve to make some compensation — almost always less than complete — for the loss. Further, some risks — betting on a horse — are uninsurable.

The processes of identifying, analyzing and assessing, mitigating, or transferring risk are generally characterized as Risk Management.

There are a few key questions at the core of the Risk Management process:

1. What could happen (threat event)?
2. If it happened, how bad could it be (threat impact)?
3. How often could it happen (threat frequency, annualized)?
4. How certain are the answers to the first three questions (recognition of uncertainty)?

These questions are answered by analyzing and assessing risk.

Uncertainty is the central issue of risk. Sure, one might pass successfully on the curve or win big at the races, but does the gain warrant taking the risk? Do the few seconds saved with the unsafe pass warrant the possible head-on collision? Are you betting this month's paycheck on a long shot to win? Cost/benefit analysis would most likely indicate that both of these examples are unacceptable risks.

Prudent management, having analyzed and assessed the risks by securing credible answers to these four questions, will almost certainly find there to be some unacceptable risks as a result. Now what? Three questions remain to be answered:

1. What can be done (risk mitigation)?
2. How much will it cost (annualized)?
3. Is it cost effective (cost/benefit analysis)?

Answers to these questions, decisions to budget and execute recommended activities, and the subsequent and ongoing management of all risk mitigation measures — including periodic reassessment — comprise the balance of the Risk Management process.

Managing the risks associated with information in the information technology (IT)environment, Information Risk Management, is an increasingly complex and dynamic task. In the budding Information Age, the technology of information storage, processing, transfer, and access has exploded, leaving efforts to secure that information effectively in a never-ending catch-up mode. For the risks potentially associated with information and information technology to be identified and managed cost-effectively, it is essential that the process of analyzing and assessing risk is well understood by all parties — and executed on a timely basis. This chapter is written with the objective of illuminating the process and the issues of risk analysis and assessment.

TERMS AND DEFINITIONS

To discuss the history and evolution of information risk analysis and assessment, several terms whose meanings are central to this discussion should first be defined.

Annualized Loss Expectancy (ALE) — This discrete value is derived, classically, from the following algorithm (see also the definitions for single loss expectancy [SLE] and annualized rate of occurrence [ARO] below):

$$\text{SINGLE LOSS EXPECTANCY} \times \text{ANNUALIZED RATE OF OCCURRENCE} = \text{ANNUALIZED LOSS EXPECTANCY}$$

To effectively identify the risks and to plan budgets for information risk management, it is helpful to express loss expectancy in annualized terms. For example, the preceding algorithm will show that the **ALE** for a threat (with an **SLE** of $1,000,000) that is expected to occur only about once in 10,000 years is ($1,000,000 divided by 10,000) only $100.00. When the expected threat frequency (**ARO**) is factored into the equation, the significance of this risk factor is addressed and integrated into the information risk management process. Thus, the risks are more accurately portrayed, and the basis for meaningful cost/benefit analysis of risk reduction measures is established.

Annualized Rate of Occurrence (ARO) — This term characterizes, on an annualized basis, the frequency with which a threat is expected to occur. For example, a threat occurring once in 10 years has an **ARO** of 1/10 or 0.1; a threat occurring 50 times in a given year has an **ARO** of 50.0. The possible range of frequency values is from 0.0 (the threat is not expected to occur) to some whole number whose magnitude depends on the type and population of threat sources. For example, the upper value could exceed 100,000 events per year for minor, frequently experienced threats such as misuse-of-resources. For an example of how quickly the number of threat events can mount, imagine a small organization — about 100 staff members — having logical access to an information processing system. If each of those 100 persons misused the system only once a month, misuse events would be occurring at the rate of 1,200 events per year. It is useful to note here that many confuse **ARO** or frequency with the term and concept of probability (defined below). While the statistical and mathematical significance of these frequency and probability metrics tend to converge at about 1/100 and become essentially indistinguishable below that level of frequency or probability, they become increasingly divergent above 1/100 to the point where probability stops — at 1.0 or certainty — and frequency continues to mount undeterred, by definition.

Exposure Factor (EF) — This factor represents a measure of the magnitude of loss or impact on the value of an asset. It is expressed as a percent, ranging from 0% to 100%, of asset value loss arising from a threat event.

This factor is used in the calculation of single loss expectancy (**SLE**), which is defined below.

Information Asset — This term, in general, represents the body of information an organization must have to conduct its mission or business. A specific information asset may consist of any subset of the complete body of information, i.e., accounts payable, inventory control, payroll, etc. Information is regarded as an intangible asset separate from the media on which it resides. There are several elements of value to be considered: First is the simple cost of replacing the information, second is the cost of replacing supporting software, and third through fifth is a series of values that reflect the costs associated with loss of the information's confidentiality, availability, and integrity. Some consider the supporting hardware and netware to be information assets as well. However, these are distinctly tangible assets. Therefore, using tangibility as the distinguishing characteristic, it is logical to characterize hardware differently than the information itself. Software, on the other hand, is often regarded as information.

These five elements of the value of an information asset often dwarf all other values relevant to an assessment of information-related risk. It should be noted that these elements of value are not necessarily additive for the purpose of assessing risk. In both assessing risk and establishing cost-justification for risk-reducing safeguards, it is useful to be able to isolate the value of safeguard effects among these elements.

Clearly, for an organization to conduct its mission or business, the necessary information must be present where it is supposed to be, when it is supposed to be there, and in the expected form. Further, if desired confidentiality is lost, results could range from no financial loss if confidentiality is not an issue, to loss of market share in the private sector, to compromise of national security in the public sector.

Qualitative/Quantitative — These terms indicate the (oversimplified) binary categorization of risk metrics and information risk management techniques. In reality, there is a spectrum across which these terms apply, virtually always in combination. This spectrum may be described as the degree to which the risk management process is quantified. If all elements — asset value, impact, threat frequency, safeguard effectiveness, safeguard costs, uncertainty, and probability — are quantified, the process may be characterized as fully quantitative.

It is virtually impossible to conduct a purely quantitative risk management project, because the quantitative measurements must be applied to the qualitative properties, i.e., characterizations of vulnerability, of the target environment. For example, "failure to impose logical access control" is a qualitative statement of vulnerability. However, it is possible to conduct a purely qualitative risk management project. A vulnerability analysis, for

example, may identify only the absence of risk-reducing countermeasures, such as logical access controls. Even this simple qualitative process has an implicit quantitative element in its binary — yes/no — method of evaluation. In summary, risk analysis and assessment techniques should be described not as either qualitative or quantitative but in terms of the degree to which such elementary factors as asset value, exposure factor, and threat frequency are assigned quantitative values.

Probability — This term characterizes the chance or likelihood, in a finite sample, that an event will occur or that a specific loss value may be attained should the event occur. For example, the probability of getting a six on a single roll of a die is 1/6, or 0.16667. The possible range of probability values is 0.0 to 1.0. A probability of 1.0 expresses certainty that the subject event will occur within the finite interval. Conversely, a probability of 0.0 expresses certainty that the subject event will not occur within the finite interval.

Risk — The potential for harm or loss, best expressed as the answer to those four questions:

- What could happen? (What is the threat?)
- How bad could it be? (What is the impact or consequence?)
- How often might it happen? (What is the frequency?)
- How certain are the answers to the first three questions? (What is the degree of confidence?)

The key element among these is the issue of uncertainty captured in the fourth question. If there is no uncertainty, there is no "risk," per se.

Risk Analysis — This term represents the process of analyzing a target environment and the relationships of its risk-related attributes. The analysis should identify threat vulnerabilities, associate these vulnerabilities with affected assets, identify the potential for and nature of an undesirable result, and identify and evaluate risk-reducing countermeasures.

Risk Assessment — This term represents the assignment of value to assets, threat frequency (annualized), consequence (i.e., exposure factors), and other elements of chance. The reported results of risk analysis can be said to provide an assessment or measurement of risk, regardless of the degree to which quantitative techniques are applied. For consistency in this article, the term risk assessment hereafter is used to characterize both the process and the results of analyzing and assessing risk.

Risk Management — This term characterizes the overall process. The first phase, risk assessment, includes identification of the assets at risk and their value, risks that threaten a loss of that value, risk-reducing measures, and the budgetary impact of implementing decisions related to the acceptance, mitigation, or transfer of risk. The second phase of risk management

includes the process of assigning priority to, budgeting, implementing, and maintaining appropriate risk-reducing measures. Risk management is a continuous process.

Safeguard — This term represents a risk-reducing measure that acts to detect, prevent, or minimize loss associated with the occurrence of a specified threat or category of threats. Safeguards are also often described as controls or countermeasures.

Safeguard Effectiveness — This term represents the degree, expressed as a percent, from 0% to 100%, to which a safeguard may be characterized as effectively mitigating a vulnerability (defined below) and reducing associated loss risks.

Single Loss Expectancy or Exposure (**SLE**) **—** This value is classically derived from the following algorithm to determine the monetary loss (impact) for each occurrence of a threatened event:

ASSET VALUE × EXPOSURE FACTOR = SINGLE LOSS EXPECTANCY

The **SLE** is usually an end result of a business impact analysis (BIA). A BIA typically stops short of evaluating the related threats' **ARO** or their significance. The **SLE** represents only one element of risk, the expected impact, monetary or otherwise, of a specific threat event. Because the BIA usually characterizes the massive losses resulting from a catastrophic event, however improbable, it is often employed as a scare tactic to get management attention — and loosen budgetary constraints — often unreasonably.

Threat — This term defines an event (e.g., a tornado, theft, or computer virus infection), the occurrence of which could have an undesirable impact.

Uncertainty — This term characterizes the degree, expressed as a percent, from 0.0% to 100%, to which there is less than complete confidence in the value of any element of the risk assessment. Uncertainty is typically measured inversely with respect to confidence, i.e., if confidence is low, uncertainty is high.

Vulnerability — This term characterizes the absence or weakness of a risk-reducing safeguard. It is a condition that has the potential to allow a threat to occur with greater frequency, greater impact, or both. For example, not having a fire suppression system could allow an otherwise minor, easily quenched fire to become a catastrophic fire. The expected frequency (**ARO**) and the exposure factor (**EF**) for major and catastrophic fire are both increased as a consequence of not having a fire suppression system.

CENTRAL TASKS OF INFORMATION RISK MANAGEMENT

The following sections describe the tasks central to the comprehensive information risk management process. These tasks provide concerned

management with credible decision support information regarding the identification and valuation of assets potentially at risk, an assessment of risk, and cost-justified recommendations for risk reduction. Thus, the execution of well-informed management decisions whether to accept, mitigate, or transfer risk cost-effectively is supported. The degree of quantitative orientation determines how the results are characterized, and, to some extent, how they are used. Each of these tasks is discussed below.

Establish Information Risk Management (IRM) Policy

A sound IRM program is founded on a well thought out IRM policy infrastructure that effectively addresses all elements of information security. Generally Accepted Information Security Principles GASSP) currently being developed, based on an Authoritative Foundation of supporting documents and guidelines, will be helpful in executing this task.

IRM policy should begin with a high-level policy statement and supporting objectives, scope, constraints, responsibilities, and approach. This high-level policy statement should drive subordinate policy, from logical access control to facilities security to contingency planning.

Finally, IRM policy should be communicated effectively — and enforced — to all parties. Note that this is important for both internal control and external control — EDI, the web, and the internet — for secure interface with the rest of the world.

Establish and Fund an IRM Team

Much of IRM functionality should already be in place — logical access control, contingency planning, etc. However, it is likely that the central task of IRM, risk assessment, has not been built into the established approach to IRM or has, at best, been given only marginal support.

At the most senior management level possible, the tasks and responsibilities of IRM should be coordinated and IRM-related budgets cost-justified based on a sound integration and implementation of the risk assessment process. At the outset, the IRM team may be drawn from existing IRM-related staff. The person charged with responsibility for executing risk assessment tasks should be an experienced IT generalist with a sound understanding of the broad issues of information security and the ability to "sell" these concepts to management. This person will need the incidental support of one who can assist at key points of the risk assessment task, i.e., scribing a Modified Delphi information valuation (see below for details).

In the first year of an IRM program, the lead person could be expected to devote 50 to 75% of his/her time to the process of establishing and executing the balance of the IRM tasks, the first of which follows immediately

below. Funds should be allocated (1) according to the above minimum staffing, and (2) to acquire, and be trained in the use of, a suitable automated risk assessment tool — $25 to 35K.

Establish IRM Methodology and Tools

There are two fundamental applications of risk assessment to be addressed (1) determining the current status of information security in the target environment(s) and ensuring that associated risk is managed (accepted, mitigated, or transferred) according to policy, and (2) assessing risk strategically. Strategic assessment assures that the risks associated with alternative strategies are effectively considered before funds are expended on a specific change in the IT environment, a change that could have been shown to be "too risky." Strategic assessment allows management to effectively consider the risks associated with various strategic alternatives in its decision making process and weigh those risks against the benefits and opportunities associated with each alternative business or technical strategy.

With the availability of proven automated risk assessment tools, the methodology is, to a large extent, determined by the approach and procedures associated with the tool of choice. An array of such tools is listed at the end of this chapter. Increasingly, management is looking for quantitative results that support a credible cost/benefit analysis and budgetary planning.

Identify and Measure Risk

Once IRM policy, team, and risk assessment methodology and tool are established and acquired, the first risk assessment will be executed. This first risk assessment should be scoped as broadly as possible, so that (1) management is provided with a good sense of the current status of information security, and (2) management has a sound basis for establishing initial risk acceptance criteria and risk mitigation priorities.

Project Sizing. This task includes the identification of background, scope, constraints, objectives, responsibilities, approach, and management support. Clear project sizing statements are essential to a well-defined and well-executed risk assessment project. It should also be noted that a clear articulation of project constraints (what is not included in the project) is very important to the success of a risk assessment.

Threat Analysis. This task includes the identification of threats that may adversely impact the target environment. This task is important to the success of the entire IRM program and should be addressed, at least initially, by risk assessment experts to ensure that all relevant risks are adequately

considered. One without risk management and assessment experience may fail to consider a threat, whether of natural causes or the result of human behavior, that stands to cause substantial harm or loss to the organization. Some risk assessment tools, such as BDSS(tm), help to preclude this problem by assuring that all threats are addressed as a function of expert system knowledge bases.

Asset Identification and Valuation. This task includes the identification of assets, both tangible and intangible, their replacement costs, and the further valuing of information asset availability, integrity, and confidentiality. These values may be expressed in monetary (for quantitative) or nonmonetary (for qualitative) terms. This task is analogous to a BIA in that it identifies the assets at risk and their value.

Vulnerability Analysis. This task includes the qualitative identification of vulnerabilities that could increase the frequency or impact of threat event(s) affecting the target environment.

Risk Evaluation. This task includes the evaluation of all collected information regarding threats, vulnerabilities, assets, and asset values in order to measure the associated chance of loss and the expected magnitude of loss for each of an array of threats that could occur. Results are usually expressed in monetary terms on an annualized basis (ALE) or graphically as a probabilistic "risk curve" for a quantitative risk assessment. For a qualitative risk assessment, results are usually expressed through a matrix of qualitative metrics such as ordinal ranking (low, medium, high or 1, 2, 3).

Interim Reports and Recommendations. These key reports are often issued during this process to document significant activity, decisions, and agreements related to the project:

- Project Sizing — This report presents the results of the project sizing task. The report is issued to senior management for their review and concurrence. This report, when accepted, assures that all parties understand and concur in the nature of the project before it is launched.
- Asset Identification and Valuation — This report may detail (or summarize) the results of the asset valuation task, as desired. It is issued to management for their review and concurrence. Such review helps prevent conflict about value later in the process. This report often provides management with their first insight into the value of the availability, confidentiality, or integrity of their information assets.
- Risk Evaluation — This report presents management with a documented assessment of risk in the current environment. Management may choose to accept that level of risk (a legitimate management decision) with no further action or to proceed with risk mitigation analysis.

Establish Risk Acceptance Criteria

With the results of the first risk assessment — through the risk evaluation task and associated reports (see below), management, with the interpretive help from the IRM leader, should establish the maximum acceptable financial risk, for example, "Do not accept more than a 1 in 100 chance of losing $1,000,000," in a given year. And, with that, and possibly additional risk acceptance criteria, such as "Do not accept an ALE greater than $500,000," proceed with the task of risk mitigation.

Mitigate Risk

The first step in this task is to complete the risk assessment with the risk mitigation, costing, and cost/benefit analysis. This task provides management with the decision support information necessary to plan for, budget, and execute actual risk mitigation measures. In other words, fix the financially unacceptable vulnerabilities. The following risk assessment tasks are discussed in further detail under the section "Tasks of Risk Assessment" later in this chapter.

Safeguard Selection and Risk Mitigation Analysis. This task includes the identification of risk-reducing safeguards that mitigate vulnerabilities and the degree to which selected safeguards can be expected to reduce threat frequency or impact. In other words, this task comprises the evaluation of risk regarding assets and threats before and after selected safeguards are applied.

Cost Benefit Analysis. This task includes the valuation of the degree of risk mitigation that is expected to be achieved by implementing the selected risk-mitigating safeguards. The gross benefit less the annualized cost for safeguards selected to achieve a reduced level of risk, yields the net benefit. Tools such as present value and return on investment are often applied to further analyze safeguard cost-effectiveness.

Final Report. This report includes the interim reports' results as well as details and recommendations from the safeguard selection and risk mitigation analysis, and supporting cost/benefit analysis tasks. This report, with approved recommendations, provides responsible management with a sound basis for subsequent risk management action and administration.

Monitor Information Risk Management Performance

Having established the IRM program, and gone this far — recommended risk mitigation measures have been acquired/developed and implemented — it is time to begin and maintain a process of monitoring IRM performance. This can be done by periodically reassessing risks to ensure that there is sustained adherence to good control or that failure to do so is

revealed, consequences considered, and improvement, as appropriate, duly implemented.

Strategic risk assessment plays a significant role in the risk mitigation process by helping to avoid uninformed risk acceptance and having, later, to retrofit (typically much more costly than built-in security or avoided risk) necessary information security measures.

There are numerous variations on this risk management process, based on the degree to which the technique applied is quantitative and how thoroughly all steps are executed. For example, the asset identification and valuation analysis could be performed independently. This task is often characterized as a business impact analysis. The vulnerability analysis could also be executed independently.

It is commonly but incorrectly assumed that information risk management is concerned only with catastrophic threats, that it is useful only to support contingency planning and related activities. A well-conceived and well-executed risk assessment can, and should, be used effectively to identify and quantify the consequences of a wide array of threats that can and do occur, often with significant frequency, as a result of ineffectively implemented or nonexistent IT management, administrative, and operational controls.

A well-run information risk management program — an integrated risk management program — can help management to significantly improve the cost-effective performance of its information technology environment, whether it is mainframe, client-server, internet, or any combination, and to ensure cost-effective compliance with applicable regulatory requirements.

The integrated risk management concept recognizes that many often uncoordinated units within an organization play an active role in managing the risks associated with the failure to assure the confidentiality, availability, and integrity of information. The following quote from FIPSPUB-73, published June 30, 1980, is a powerful reminder that information security was long ago recognized as a central, not marginal issue:

> "Security concerns should be an integral part of the entire planning, development, and operation of a computer application. Much of what needs to be done to improve security is not clearly separable from what is needed to improve the usefulness, reliability, effectiveness, and efficiency of the computer application."

Resistance and Benefits

"Why should I bother with doing risk assessment?!" "I already know what the risks are!" "I've got enough to worry about already!" "It hasn't happened yet..." Sound familiar? Most resistance to risk assessment boils down to one of three conditions:

- Ignorance,
- Arrogance, and
- Fear.

Management often is ignorant, except in the most superficial context, of the risk assessment process, the real nature of the risks, and the benefits of risk assessment. Risk assessment is not yet a broadly accepted element of the management toolkit, yet virtually every "Big 5" consultancy, and other major providers of information security services, offer risk assessment in some form.

Arrogance of the bottom line often drives an organization's attitude about information security, therefore about risk assessment. "Damn the torpedoes, full speed ahead!" becomes the marching order. If it can't readily be shown to improve profitability, don't do it. It is commendable that IT has become so reliable that management could maintain that attitude for more than a few giddy seconds. Despite the fact that a well-secured IT environment is also a well-controlled, efficient IT environment, management often has difficulty seeing how sound information security can and does affect the bottom line in a positive way.

This arrogance is often described euphemistically as an "entrepreneurial culture."

Finally, there is the fear factor — fear of discovering that the environment is not as well-managed as it could be — and having to take responsibility for that; fear of discovering, and having to address, risks not already known; and fear of being shown to be ignorant or arrogant.

While good information security may seem expensive, inadequate information security will be not just expensive, but, sooner or later, catastrophic.

Risk assessment, while still a young science, with a certain amount of craft involved, has proven itself to be very useful in helping management understand and cost-effectively address the risks to their information and IT environments.

Finally, with regard to resistance, when risk assessment had to be done manually, or could be done only qualitatively, the fact that the process could take many months to execute (and that it was not amenable to revision or "what if" assessment) was a credible obstacle to its successful use. But that is no longer the case.

Some specific benefits are described below:

- Risk assessment helps management understand:
  1. What is at risk?
  2. The value at risk — as associated with the identity of information assets and with the confidentiality, availability, and integrity of information assets.

3. The kinds of threats that could occur and their financial consequences annualized.
4. Risk mitigation analysis. What can be done to reduce risk to an acceptable level.
5. Risk mitigation costs (annualized) and associated cost/benefit analysis. Whether suggested risk mitigation activity is cost-effective.

- Risk assessment enables a strategic approach to information risk management. In other words, possible changes being considered for the IT environment can be assessed to identify the least risk alternative before funds are committed to any alternative. This information complements the standard business case for change and may produce critical decision support information that could otherwise be overlooked.
- "What if" analysis is supported. This is a variation on the strategic approach information to risk management. Alternative approaches can be considered and their associated level of risk compared in a matter of minutes.
- Information security professionals can present their recommendations with credible statistical and financial support.
- Management can make well-informed information risk management decisions.
- Management can justify, with credible quantitative tools, information security budgets/expenditures that are based on a reasonably objective risk assessment.
- Good information security, supported by quantitative risk assessment, will ensure an efficient, cost-effective IT environment.
- Management can avoid spending that is based solely on a perception of risk.
- An information risk management program based on the sound application of quantitative risk assessment can be expected to reduce liability exposure and insurance costs.

Qualitative vs. Quantitative Approaches

Background. As characterized briefly above, there are two fundamentally different metric schemes applied to the measurement of risk elements, qualitative and quantitative. The earliest efforts to develop an information risk assessment methodology were reflected originally in the National Bureau of Standards (now the National Institute of Standards & Technology [NIST] FIPSPUB-31 Automated Data Processing Physical Security and Risk Management, published in 1974. That idea was subsequently articulated in detail with the publication of FIPSPUB-65 Guidelines for Automated Data Processing Risk Assessment, published in August of 1979. This methodology provided the underpinnings for OMB A-71, a federal requirement for

conducting "quantitative risk assessment" in the federal government's information processing environments.

Early efforts to conduct quantitative risk assessments ran into considerable difficulty. First, because no initiative was executed to establish and maintain an independently verifiable and reliable set of risk metrics and statistics, everyone came up with their own approach; second, the process, while simple in concept, was complex in execution; and third, large amounts of data were collected that required substantial and complex mapping, pairing, and calculation to build representative risk models; fourth, with no software and desktop computers, the work was done manually — a very tedious and time-consuming process. Results varied significantly.

As a consequence, while some developers launched and continued efforts to develop credible and efficient automated quantitative risk assessment tools, others developed more expedient qualitative approaches that did not require independently objective metrics — and OMB A-130, an update to OMB A-71, was released, lifting the "quantitative" requirement for risk assessment in the federal government.

These qualitative approaches enabled a much more subjective approach to the valuation of information assets and the scaling of risk. In Exhibit 1, for example, the value of the availability of information and the associated risk were described as "low," "medium," or "high" in the opinion of knowledgeable management, as gained through interview or questionnaires.

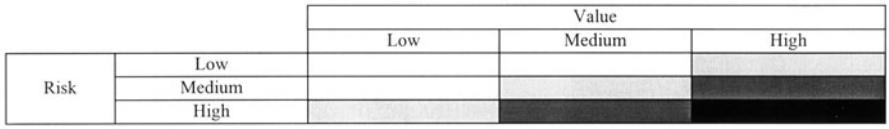


Exhibit 1. Value of the Availability of Information and the Associated Risk

Often, when this approach is taken, a strategy is defined wherein the highest risk exposures (darkest shaded areas) require prompt attention, the moderate risk exposures (lightly shaded areas) require plans for corrective attention, and the lowest risk exposures (unshaded areas) can be accepted.

Elements of Risk Metrics

There are six primitive elements of risk modeling to which some form of metric can be applied:

- Asset Value
- Threat Frequency
- Threat Exposure Factor
- Safeguard Effectiveness
- Safeguard Cost
- Uncertainty

To the extent that each of these elements is quantified in independently objective metrics such as the monetary replacement value for Asset Value or the Annualized Rate of Occurrence for Threat Frequency, the risk assessment is increasingly quantitative. If all six elements are quantified with independently objective metrics, the risk assessment is fully quantified, and the full range of statistical analyses is supported.

Exhibit 2 relates both the quantitative and qualitative metrics for these six elements.

Note: The Baseline approach makes no effort to scale risk or to value information assets. Rather, the Baseline approach seeks to identify in-place safeguards, compare those with what industry peers are doing to secure their information, then enhance security wherever it falls short of industry peer security. A further word of caution is appropriate here. The Baseline approach is founded on an interpretation of "due care" that is at odds with the well-established legal definition of due care. Organizations relying solely on the Baseline approach could find themselves at a liability risk with an inadequate legal defense should a threat event cause a loss that could have been prevented by available technology or practice that was not implemented because the Baseline approach was used.

The classic quantitative algorithm, as presented in FIPSPUB-65, that laid the foundation for information security risk assessment is simple:

$$(\text{Asset Value} \times \text{Exposure Factor} = \text{Single Loss Exposure})$$

$$\begin{array}{ll} \times & \underline{\text{Annualized Rate of Occurrence}} \\ = & \text{Annualized Loss Expectancy} \end{array}$$

For example, let's look at the risk of fire. Assume the Asset Value is \$1M, the exposure factor is 50%, and the Annualized Rate of Occurrence is 1/10 (once in ten years). Plugging these values into the algorithm yields the following:

$$(\$1\text{M} \times 50\% = \$500\text{K}) \times 1/10 = \$50\text{K}$$

Using conventional cost/benefit assessment, the \$50K ALE represents the cost/benefit break-even point for risk mitigation measures. In other words, the organization could justify spending up to \$50K per year to prevent the occurrence or reduce the impact of a fire.

	Quantitative Metrics				Qualitative Metrics			
Risk Element	Monetary Value	Percent Factors (%)	Annualized Rate of Occurrence	Bounded Distribution (Range)	Low, Medium & High	Ordinal Ranking	Vital, Critical, Important, etc.	Baseline
Asset Value	x			x	x	x	x	
Threat Frequency (Annualized)			x	x	x	x		
Threat Exposure Factor		x		x	x	x		
Recommended Safeguard Effectiveness		x		x	x	x		
Safeguard Cost (Annualized)	x			x	x	x		
Uncertainty (Confidence Factor)		x		x	x	x		

Exhibit 2. Quantitative and Qualitative Metrics for the Six Elements

It is true that the classic FIPSPUB-65 quantitative risk assessment took the first steps toward establishing a quantitative approach. However, in the effort to simplify fundamental statistical analysis processes so that everyone could readily understand, the algorithms developed went too far. The consequence was results that had little credibility for several reasons, three of which follow:

- The classic algorithm addresses all but two of the elements, recommended safeguard effectiveness, and uncertainty. Both of these must be addressed in some way, and uncertainty, the key risk factor, must be addressed explicitly.
- The algorithm cannot distinguish effectively between low frequency/high impact threats (such as "fire") and high frequency/low impact threats (such as "misuse of resources"). Therefore, associated risks can be significantly misrepresented.
- Each element is addressed as a discrete value, which, when considered with the failure to address uncertainty explicitly, makes it difficult to actually model risk and illustrate probabilistically the range of potential undesirable outcomes.

Yes, this primitive algorithm did have shortcomings, but advances in quantitative risk assessment technology and methodology to explicitly address uncertainty and support technically correct risk modeling have largely done away with those problems.

Pros and Cons of Qualitative and Quantitative Approaches

In this brief analysis, the features of specific tools and approaches will not be discussed. Rather, the pros and cons associated in general with qualitative and quantitative methodologies will be addressed.

Qualitative — Pros

- Calculations, if any, are simple and readily understood and executed.
- It is usually not necessary to determine the monetary value of information (its availability, confidentiality, and integrity).
- It is not necessary to determine quantitative threat frequency and impact data.
- It is not necessary to estimate the cost of recommended risk mitigation measures and calculate cost/benefit.
- A general indication of significant areas of risk that should be addressed is provided.

Qualitative — Cons

- The risk assessment and results are essentially subjective in both process and metrics. The use of independently objective metrics is eschewed.

- No effort is made to develop an objective monetary basis for the value of targeted information assets. Hence, the perception of value may not realistically reflect actual value at risk.
- No basis is provided for cost/benefit analysis of risk mitigation measures, only subjective indication of a problem.
- It is not possible to track risk management performance objectively when all measures are subjective.

Quantitative — Pros

- The assessment and results are based substantially on independently objective processes and metrics. Thus meaningful statistical analysis is supported.
- The value of information (availability, confidentiality, and integrity), as expressed in monetary terms with supporting rationale, is better understood. Thus, the basis for expected loss is better understood.
- A credible basis for cost/benefit assessment of risk mitigation measures is provided. Thus, information security budget decision-making is supported.
- Risk management performance can be tracked and evaluated.
- Risk assessment results are derived and expressed in management's language, monetary value, percentages, and probability annualized. Thus risk is better understood.

Quantitative — Cons

- Calculations are complex. If they are not understood or effectively explained, management may mistrust the results of "black box" calculations.
- It is not practical to attempt to execute a quantitative risk assessment without using a recognized automated tool and associated knowledge bases. A manual effort, even with the support of spread sheet and generic statistical software, can easily take ten to twenty times the work effort required with the support of a good automated risk assessment tool.
- A substantial amount of information about the target information and its IT environment must be gathered.
- As of this writing, there is not yet a standard, independently developed and maintained threat population and threat frequency knowledge base. Thus the users must rely on the credibility of the vendors who develop and support extant automated tools or do threat research on their own.

Business Impact Analysis vs. Risk Assessment

There is still confusion as to the difference between a Business Impact Analysis (BIA) and risk assessment. It is not unusual to hear the terms used

interchangeably. But that is not correct. A BIA, at the minimum, is the equivalent of one task of a risk assessment — Asset Valuation, a determination of the value of the target body of information and its supporting IT resources. At the most, the BIA will develop the equivalent of a Single Loss Exposure, with supporting details, of course, usually based on a worst case scenario. The results are most often used to convince management that they should fund development and maintenance of a contingency plan.

Information security is much more than contingency planning. A BIA often requires 75 to 100% or more of the work effort (and associated cost) of a risk assessment, while providing only a small fraction of the useful information provided by a risk assessment. A BIA includes little if any vulnerability assessment, and no sound basis for cost/benefit analysis.

Target Audience Concerns

Risk assessment continues to be viewed with skepticism by many in the ranks of management. Yet those for whom a well-executed risk assessment has been done have found the results to be among the most useful analyses ever executed for them.

To cite a few examples:

- In one case, involving an organization with multiple large IT facilities — one of which was particularly vulnerable — a well-executed risk assessment promptly secured the attention of the Executive Committee, which had resisted all previous initiatives to address the issue. Why? Because IT management could not previously supply justifying numbers to support its case. With the risk assessment in hand, IT management got the green light to consolidate IT activities from the highly vulnerable site to another facility with much better security. This was accomplished despite strong union and staff resistance. The move was executed by this highly regulated and bureaucratic organization within three months of the quantitative risk assessment's completion! The quantitative risk assessment provided what was needed, credible facts and numbers of their own.
- In another case, a financial services organization found, as a result of a quantitative risk assessment, that they were carrying four to five times the amount of insurance warranted by their level of exposure. They reduced coverage by half, still retaining a significant cushion, and have since saved hundreds of thousands of dollars in premiums.
- In yet another case, management of a relatively young but rapidly growing organization had maintained a rather "entrepreneurial" attitude toward IT in general, until presented with the results of a risk assessment that gave them a realistic sense of the risks inherent to that posture. Substantial policy changes were made on the spot, and

information security began receiving real consideration, not just lip service.
- Finally, an large energy industry organization was considering relocating its IT function from its original facility to a bunkered, tornado-proof facility across town that was being abandoned by a major insurance company. The energy company believed that they could reduce their IT related risk substantially. The total cost of the move would have run into the millions of dollars. Upon executing a strategic risk assessment for the alternatives, it was found that the old facility was sound and relocating would not significantly reduce their risk. In fact, it was found that the biggest risks were being taken in their failure to maintain good management practices.

Some specific areas of concern are addressed below.

Diversion of Resources. That organizational staff will have to spend some time providing information for the risk assessment is often a major concern. Regardless of the nature of the assessment, there are two key areas of information gathering that will require staff time and participation beyond that of the person(s) responsible for executing the risk assessment:

1. Valuing the intangible information asset's confidentiality, integrity, and availability, and
2. Conducting the vulnerability analysis.

These tasks will require input from two entirely different sets of people in most cases.

Valuing the Intangible Information Asset. There are a number of approaches to this task, and the amount of time it takes to execute will depend on the approach as well as whether it is qualitative or quantitative. As a general rule of thumb, however, one could expect all but the most cursory qualitative approach to require one to four hours of continuous time from two to five key knowledgeable staff for each intangible information asset valued.

Experience has shown that the Modified Delphi approach is the most efficient, useful, and credible. For detailed guidance, refer to the "Guideline for Information Valuation" (GIV) published by the Information System Security Association (ISSA). This approach will require (typically) the participation of three to five staff knowledgeable on various aspects of the target information asset. A Modified Delphi meeting routinely lasts 4 hours; so, for each target information asset, key staff time of 12 to 16 hours will be expended in addition to about 20 to 36 hours total for a meeting facilitator (4 hours) and a scribe (16 to 32 hours).

Providing this information has proven to be a valuable exercise for the source participants, and the organization, by giving them significant insight

into the real value of the target body of information and the consequences of losing its confidentiality, availability, or integrity. Still, this information alone should not be used to support risk mitigation cost/benefit analysis.

While this "Diversion of Resources" may be viewed initially by management with some trepidation, the results have invariably been judged more than adequately valuable to justify the effort.

Conducting the Vulnerability Analysis. This task, which consists of identifying vulnerabilities, can and should take no more than 5 work days (about 40 hours) of one-on-one meetings with staff responsible for managing or administering the controls and associated policy, e.g., logical access controls, contingency planning, change control, etc. The individual meetings — actually guided interviews, ideally held in the interviewees' workspace — should take no more than a couple of hours. Often, these interviews take as little as 5 minutes. Collectively, however, the interviewees' total diversion could add up to as much as 40 hours. The interviewer will, of course, spend matching time, hour for hour. This one-on-one approach minimizes disruption while maximizing the integrity of the vulnerability analysis by assuring a consistent level-setting with each interviewee.

Credibility of the Numbers. Twenty years ago, the task of coming up with "credible" numbers for information asset valuation, threat frequency and impact distributions, and other related risk factors was daunting. Since then, the GIV was published, and significant progress has been made by some automated tools' handling of the numbers and their associated knowledge bases. The knowledge bases that were developed on the basis of significant research to establish credible numbers. And, credible results are provided if proven algorithms with which to calculate illustrative risk models are used.

However, manual approaches or automated tools that require the users to develop the necessary quantitative data are susceptible to a much greater degree of subjectivity and poorly informed assumptions.

In the past couple of years, there have been some exploratory efforts to establish a Threat Research Center tasked with researching and establishing:

1. a standard Information security threat population,
2. associated threat frequency data, and
3. associated threat scenario and impact data;

and maintaining that information while assuring sanitized source channels that protect the providers of impact and scenario information from disclosure. As recognition of the need for strong information security and associated risk assessment continues to increase, the pressure to launch this function will eventually be successful.

Subjectivity. The ideal in any analysis or assessment is complete objectivity. Just as there is a complete spectrum from qualitative to quantitative, there is a spectrum from subjective to increasingly objective. As more of the elements of risk are expressed in independently objective terms, the degree of subjectivity is reduced accordingly, and the results have demonstrable credibility.

Conversely, to the extent a methodology depends on opinion, point of view, bias, or ignorance (subjectivity), the results will be of increasingly questionable utility. Management is loath to make budgetary decisions based on risk metrics that express value and risk in terms such as low, medium, and high.

There will always be some degree of subjectivity in assessing risks. However, to the extent that subjectivity is minimized by the use of independently objective metrics, and the biases of tool developers, analysts, and knowledgeable participants are screened, reasonably objective, credible risk modeling is achievable.

Utility of Results. Ultimately, each of the above factors (Diversion of Resources, Credibility of the Numbers, Subjectivity, and, in addition, Timeliness) plays a role in establishing the utility of the results. Utility is often a matter of perception. If management feels that the execution of a risk assessment is diverting resources from their primary mission inappropriately, if the numbers are not credible, if the level of subjectivity exceeds an often intangible cultural threshold for the organization, or if the project simply takes so long that the results are no longer timely, then the attention — and trust — of management will be lost or reduced along with the utility of the results.

A risk assessment executed with the support of contemporary automated tools can be completed in a matter of weeks, not months. Developers of the best automated tools have done significant research into the qualitative elements of good control, and their qualitative vulnerability assessment knowledge bases reflect that fact. The same is true with regard to their quantitative elements. Finally, in building these tools to support quantitative risk assessment, successful efforts have been made to minimize the work necessary to execute a quantitative risk assessment.

The bottom line is that it makes very little sense to execute a risk assessment manually or build one's own automated tool except in the most extraordinary circumstances. A risk assessment project that requires many work-months to complete manually (with virtually no practical "what-if" capability) can, with sound automated tools, be done in a matter of days, or weeks at worst, with credible, useful results.

TASKS OF RISK ASSESSMENT

In this section, we will explore the classic tasks of risk assessment and key issues associated with each task, regardless of the specific approach to be employed. The focus will, in general, be primarily on quantitative methodologies. However, wherever possible, related issues in qualitative methodologies will also be discussed.

Project Sizing

In virtually all project methodologies there are a number of elements to be addressed to ensure that all participants, and the target audience, understand and are in agreement about the project. These elements include:

- Background
- Purpose
- Scope
- Constraints
- Objective
- Responsibilities
- Approach

In most cases, it would not be necessary to discuss these individually, as most are well-understood elements of project methodology in general. In fact, they are mentioned here for the exclusive purpose of pointing out the importance of (1) ensuring that there is agreement between the target audience and those responsible for executing the risk assessment, and (2) describing the constraints on a risk assessment project. While a description of the scope, *what is included*, of a risk assessment project is important, it is equally important to describe specifically, in appropriate terms, *what is not included*. Typically, a risk assessment is focused on a subset of the organization's information assets and control functions. If what is not to be included is not identified, confusion and misunderstanding about the risk assessment's ramifications may result.

Again, the most important point about the project sizing task is to ensure that the project is clearly defined and that a clear understanding of the project by all parties is achieved.

Threat Analysis. In manual approaches and some automated tools, the analyst must determine what threats to consider in a particular risk assessment. Since there is not, at present, a standard threat population and readily available threat statistics, this task can require a considerable research effort. Of even greater concern is the possibility that a significant local threat could be overlooked and associated risks inadvertently

accepted. Worse, it is possible that a significant threat is intentionally disregarded.

The best automated tools currently available include a well-researched threat population and associated statistics. Using one of these tools virtually assures that no relevant threat is overlooked, and associated risks are accepted as a consequence.

If, however a determination has been made not to use one of these leading automated tools and instead to do the threat analysis independently, there are good sources for a number of threats, particularly for all natural disasters, fire, and crime (oddly enough, not so much for computer crime), even falling aircraft. Also, the console log is an excellent source for in-house experience of system development, maintenance, operations, and other events that can be converted into useful threat event statistics with a little tedious review. Finally, in-house physical and logical access logs (assuming such are maintained) can be a good source of related threat event data.

But, gathering this information independently, even for the experienced risk analyst, is no trivial task. Weeks, if not months, of research and calculation will be required, and, without validation, results may be less than credible.

For those determined to proceed independently, the following list of sources, in addition to in-house sources previously mentioned, will be useful:

- Fire — National Fire Protection Association (NFPA)
- Flood, all categories — National Oceanic and Atmospheric Administration (NOAA) and local Flood Control Districts
- Tornado — NOAA
- Hurricane — NOAA and local Flood Control Districts
- Windstorms — NOAA
- Snow — NOAA
- Icing — NOAA
- Earthquakes — U.S. Geological Survey (USGS) and local university geology departments
- Sinkholes — USGS and local university geology departments
- Crime — FBI and local law enforcement statistics, and your own in-house crime experience, if any
- Hardware failures — Vendor statistics and in-house records

Until an independent Threats Research Center is established, it will be necessary to rely on automated risk assessment tools, or vendors, or your own research for a good threat population and associated statistics.

Asset Identification and Valuation

While all assets may be valued qualitatively, such an approach is useless if there is a need to make well-founded budgetary decisions. Therefore, this discussion of Asset Identification and Valuation will assume a need for the application of monetary valuation.

There are two general categories of assets relevant to the assessment of risk in the IT environment:

- Tangible Assets, and
- Intangible Assets

Tangible Assets. The Tangible Assets include the IT facilities, hardware, media, supplies, documentation, and IT staff budgets that support the storage, processing, and delivery of information to the user community. The value of these assets is readily determined, typically, in terms of the cost of replacing them. If any of these are leased, of course, the replacement cost may be nil, depending on the terms of the lease.

Sources for establishing these values are readily found in the associated asset management groups, i.e., facilities management for replacement value of the facilities, hardware management for the replacement value for the hardware — from CPU's to controllers, routers and cabling, annual IT staff budgets for IT staff, etc.

Intangible Assets. The Intangible Assets, which might be better characterized as Information Assets, are comprised of two basic categories:

- Replacement costs for data and software, and
- The value of the confidentiality, integrity, and availability of information.

Replacement Costs. Developing replacement costs for data is not usually a complicated task unless source documents don't exist or are not backed up, reliably, at a secure off-site location. The bottom line is that "x" amount of data represents "y" key strokes — a time-consuming, but readily measurable manual key entry process.

Conceivably, source documents can now be electronically "scanned" to recover lost, electronically stored data. Clearly, scanning is a more efficient process, but it is still time-consuming. However, if neither source documents nor off-site backups exist, actual replacement may become virtually impossible, and the organization faces the question of whether such a condition can be tolerated. If, in the course of the assessment, this condition is found, the real issue is that the information is no longer available, and a determination must be made as to whether such a condition can be overcome without bankrupting the private sector organization or irrevocably compromising a government mission.

Value of Confidentiality, Integrity, and Availability. In recent years, a better understanding of the values of confidentiality, integrity, and availability and how to establish these values on a monetary basis with reasonable credibility has been achieved. That understanding is best reflected in the ISSA-published GIV referenced above. These values often represent the most significant "at risk" asset in IT environments. When an organization is deprived of one or more of these with regard to its business or mission information, depending on the nature of that business or mission, there is a very real chance that unacceptable loss will be incurred within a relatively short time.

For example, it is well-accepted that a bank that loses access to its business information (loss of availability) for more than a few days is very likely to go bankrupt.

A brief explanation of each of these three critical values for information is presented below.

- *Confidentiality* — Confidentiality is lost or compromised when information is disclosed to parties other than those authorized to have access to the information. In the complex world of IT today, there are many ways for a person to access information without proper authorization, if appropriate controls are not in place. Without appropriate controls, that access or theft of information could be accomplished without a trace. Of course, it still remains possible to simply pick up and walk away with confidential documents carelessly left lying about or displayed on an unattended, unsecured PC.
- *Integrity* — Integrity is the condition that information in or produced by the IT environment accurately reflects the source or process it represents. Integrity may be compromised in may ways, from data entry errors to software errors to intentional modification. Integrity may be thoroughly compromised, for example, by simply contaminating the account numbers of a bank's demand deposit records. Since the account numbers are a primary reference for all associated data, the information is effectively no longer available. There has been a great deal of discussion about the nature of integrity. Technically, if a single character is wrong in a file with millions of records, the file's integrity has been compromised.

  Realistically, however, some expected degree of integrity must be established. In an address file, 99% accuracy (only one out of 100 is wrong) may be acceptable. However, in the same file, if each record of 100 characters had only one character wrong — in the account number — the records would meet the poorly articulated 99% accuracy standard, but be completely compromised. In other words, the loss of integrity can have consequences that range from trivial to cata-

strophic. Of course, in a bank with one million clients, 99% accuracy means at best that the records of 10,000 clients are in error. In a hospital, even one such error could lead to loss of life!

- *Availability* — Availability, the condition that electronically stored information is where it needs to be, when it needs to be there, and in the form necessary, is closely related to the availability of the information processing technology. Whether because the process is unavailable, or the information itself is somehow unavailable, makes no difference to the organization dependent on the information to conduct its business or mission. The value of the information's availability is reflected in the costs incurred, over time, by the organization, because the information was not available, regardless of cause. A useful tool (from the Modified Delphi method) for capturing the value of availability, and articulating uncertainty, is illustrated in Exhibit 3. This chart represents the cumulative cost, over time, of the best case and worst case scenarios, with confidence factors, for the loss of availability of a specific information asset.

INTERVAL	LO$	HI$	CF %	INTERVAL	LO$	HI$	CF %
0-1 HR				4 DAYS			
2 HR				8 DAYS			
4 HR				16 DAYS			
8 HR				1 MONTH			
16 HR				2 MONTHS			
1 DAY				3 MONTHS			
2 DAY				6 MONTHS			

Exhibit 3. Capturing the Value of Availability (Modified Delphi Method)

Vulnerability Analysis

This task consists of the identification of vulnerabilities that would allow threats to occur with greater frequency, greater impact, or both. For maximum utility, this task is best conducted as a series of one-on-one interviews with individual staff members responsible for developing or implementing organizational policy through the management and administration of controls. To maximize consistency and thoroughness, and to minimize subjectivity, the vulnerability analysis should be conducted by an interviewer who guides each interviewee through a well-researched series of questions designed to ferret out all potentially significant vulnerabilities.

It should be noted that establishment and global acceptance of Generally Accepted System Security Principles (GASSP), as recommended in the National Research Council report "Computers at Risk" (12/90), the National

Information Infrastructure Task Force (NIITF) findings, the Presidential National Security and Telecommunications Advisory Council (NSTAC) report (12/96), and the President's Commission on Critical Infrastructure Protection (PCCIP) report (10/97), all of which were populated with a strong private sector representation, will go far in establishing a globally accepted knowledge base for this task. The "Treadwell Commission" report published by the American Institute of Certified Public Accountants (AICPA) Committee of Sponsoring Organizations (COSO)in 1994, "Internal Control, Integrated Framework" now, beginning in 1997, specifically requires that auditors verify that subject organizations assess and manage the risks associated with IT and other significant organizational resources. The guiding model characterized in the requirement represents quantitative risk assessment. Failure to have effectively implemented such a risk management mechanism now results in a derogatory audit finding.

Threat/Vulnerability/Asset Mapping

Without connecting — mapping — threats to vulnerabilities and vulnerabilities to assets and establishing a consistent way of measuring the consequences of their interrelationships, it becomes nearly impossible to establish the ramifications of vulnerabilities in a useful manner. Of course, intuition and common sense are useful, but how does one measure the risk and support good budgetary management and cost/benefit analysis when the rationale is so abstract?

For example, it is only good common sense to have logical access control, but how does one justify the expense? I am reminded of a major bank whose management, in a cost-cutting frenzy, came very close to terminating its entire logical access control program! With risk assessment, one can show the expected risk and annualized asset loss/probability coordinates that reflect the ramifications of a wide array of vulnerabilities. Exhibit 4 carries the illustration further with two basic vulnerabilities.

Applying some simple logic at this point will give the reader some insight into the relationships between vulnerabilities, threats, and potentially affected assets.

No Logical Access Control. Not having logical access control means that anyone can sign on the system, get to any information they wish, and do anything they wish with the information. Most tangible assets are not at risk. However, if IT staff productivity is regarded as an asset, as reflected by their annual budget, that asset could suffer a loss (of productivity) while the staff strives to reconstruct or replace damaged software or data. Also, if confidentiality is compromised by the disclosure of sensitive information (competitive strategies or client information), substantial competitive advantage and associated revenues could be lost, or liability suits for

VULNERABILITY	MAPPED THREAT(S)	AFFECTED ASSETS (At minimum)[a]
No Logical Access Control	Sabotage of Software	Software Goodwill
	Sabotage of Data/Information	Information Integrity Goodwill
	Theft of Software	Software Goodwill
	Theft of Data/Information	Information Confidentiality Goodwill
	Destruction of Software	Software Goodwill
	Destruction of Data/Information	Information Availability Goodwill
No Contingency Plan	Fire Hurricane Earthquake Flood Terrorist Attack	Facilities Hardware Media and Supplies IT Staff Budgets Software Information Availability Goodwill
	Toxic Contamination[b]	IT Staff Budgets Software Information Availability Goodwill

[a] In each case it is assumed that the indicated vulnerability is the only vulnerability, thus any impact on other information assets is expected to be insignificant. Otherwise, without current backups, for example, virtually every threat on this chart could have a significant impact on information availability

[b] Tangible assets are not shown as being impacted by a toxic contamination, aside from the IT staff budgets, because it is assumed that the toxic contamination can be cleaned up and the facilities and equipment restored to productive use.

Exhibit 4. Two Basic Vulnerabilities

disclosure of private information could be very costly. Both could cause company goodwill to suffer a loss.

Since the only indicated vulnerability is not having logical access, it is reasonable to assume monetary loss resulting from damage to the integrity of the information or the temporary loss of availability of the information is limited to the time and resources needed to recover with well-secured, off-site backups.

Therefore, it is reasonable to conclude, all other safeguards being effectively in place, that the greatest exposure resulting from not having logical access control is the damage that may result from a loss of confidentiality for a single event. But, without logical access control, there could be many such events!

What if there was another vulnerability? What if the information was not being backed up effectively? What if there were no useable backups? The loss of availability — for a single event — could become overwhelmingly expensive, forcing the organization into bankruptcy or compromising a government mission.

No Contingency Plan. Not having an effective contingency plan means that the response to any natural or man-made disaster will be without prior planning or arrangements. Thus, the expense associated with the event is not assuredly contained to a previously established maximum acceptable loss. The event may very well bankrupt the organization or compromise a government mission. This is without considering the losses associated with the Tangible Assets! Studies have found that organizations hit by a disaster and not having a good contingency plan are likely (4 out of 5) to be out of business within two years of the disaster event.

What if there were no useable backups — another vulnerability? The consequences of the loss of information availability would almost certainly be made much worse, and recovery, if possible, would be much more costly. The probability of being forced into bankruptcy is much higher.

By mapping vulnerabilities to threats to assets, we can see the interplay among them and understand a fundamental concept of risk assessment:

> *Vulnerabilities allow threats to occur with greater frequency or greater impact. Intuitively, it can be seen that the more vulnerabilities there are, the greater is the risk of loss.*

Risk Metrics/Modeling. There are a number of ways to portray risk, some qualitative, some quantitative, and some more effective than others.

In general, the objective of risk modeling is to convey to decision-makers a credible, useable portrayal of the risks associated with the IT environment, answering (again) these questions:

- What could happen (threat event)?
- How bad would it be (impact)?
- How often might it occur (frequency)?
- How certain are the answers to the first three questions (uncertainty)?

With such risk modeling, decision makers are on their way to making well-informed decisions — either to accept, mitigate, or transfer associated risk.

The following brief discussion of the two general categories of approach to these questions, qualitative and quantitative, will give the reader a degree of insight into the ramifications of using one or the other approach:

Qualitative. The definitive characteristic of the qualitative approach is the use of metrics that are subjective, such as ordinal ranking — low, medium, high, etc. (see Exhibit 5). In other words, independently objective values such as objectively established monetary value, and recorded history of threat event occurrence (frequency) are not used.

		Value		
		Low	Medium	High
Risk	Low			
	Medium			
	High			

Exhibit 5. Value of the Availability of Information and the Associated Risk

Quantitative. The definitive characteristic of quantitative approaches is the use of independently objective metrics and significant consideration given to minimizing the subjectivity that is inherent in any risk assessment. Exhibit 6 was produced from a leading automated tool, BDSS™, and illustrates quantitative risk modeling.

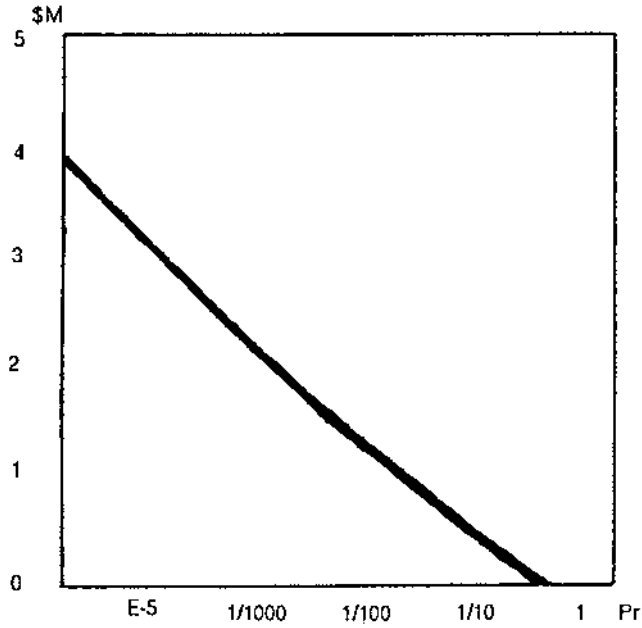


Exhibit 6. Results of Risk Evaluation in BDSS™ Before Any Risk Mitigation

The graph shown in Exhibit 6 reflects the integrated "all threats" risk that is generated to illustrate the results of Risk Evaluation in BDSS™ before any risk mitigation. The combined value of the tangible and intangible assets at risk is represented on the "Y" axis, and the probability of financial loss is represented on the "X" axis. Thus, reading this graphic model, there is a 1/10 chance of losing about $0.5M over a one year period.

The graph shown in Exhibit 7 reflects the same environment after risk mitigation and associated cost/benefit analysis. The original risk curve (Exhibit 6) is shown in Exhibit 7 with the reduced risk curve and associated average annual cost of all recommended safeguards superimposed on it, so the viewer can see the risk before risk mitigation, the expected reduction in risk, and the cost to achieve it. In Exhibit 7, the risk at 1/10 and 1/100 chance of loss is now minimal, and the risk at 1/1000 chance of loss has been reduced from about $2.0M to about $0.3M. The suggested safeguards are thus shown to be well justified.

Management Involvement and Guidance. Organizational culture plays a key role in determining, first, whether to assess risk, and second, whether to use qualitative or quantitative approaches. Many firms' management

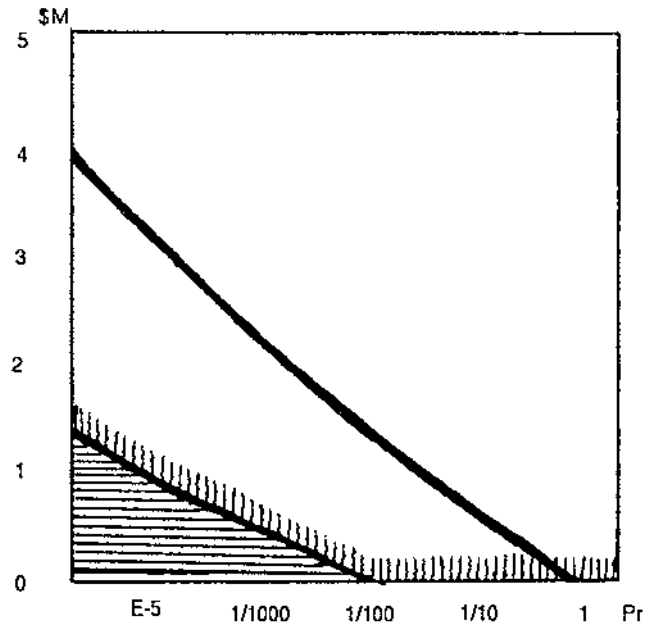


Exhibit 7. Results of Risk Evaluation After Risk Mitigation and Associated Cost/Benefit Analysis

organizations see themselves as "entrepreneurial" and have an aggressive bottom line culture. Their basic attitude is to minimize all costs, take the chance that nothing horrendous happens, and assume they can deal with it if it does happen.

Other firms, particularly larger, more mature organizations, will be more interested in a replicable process that puts results in management language such as monetary terms, cost/benefit assessment, and expected loss. Terms that are understood by business management will facilitate the creation of effective communication channels and support sound budgetary planning for information risk management.

It is very useful to understand the organizational culture when attempting to plan for a risk assessment and get necessary management support. While a quantitative approach will provide, generally speaking, much more useful information, the culture may not be ready to assess risk in significant depth.

In any case, with the involvement, support and guidance of management, more utility will be gained from the risk assessment, regardless of its qualitative or quantitative nature. And, as management gains understanding of the concepts and issues of risk assessment and begins to realize the value to be gained, reservations about quantitative approaches will diminish, and they will increasingly look toward those quantitative approaches to provide more credible, defensible budgetary support.

Risk Mitigation Analysis

With the completion of the risk modeling and associated report on the observed status of information security and related issues, management will almost certainly find some areas of risk that they are unwilling to accept and for which they wish to see proposed risk mitigation analysis. In other words, they will want answers to the last three questions for those unacceptable risks:

- What can be done?
- How much will it cost?
- Is it cost effective?

There are three steps in this process:

1. Safeguard Analysis and Expected Risk Mitigation
2. Safeguard Costing
3. Safeguard Cost/Benefit Analysis

Safeguard Analysis and Expected Risk Mitigation. With guidance from the results of the Risk Evaluation, included modeling and associated data collection tasks, and reflecting management concerns, the analyst will seek to

identify and apply safeguards that could be expected to mitigate the vulnerabilities of greatest concern to management. Management will, of course, be most concerned about those vulnerabilities that could allow the greatest loss expectancies for one or more threats, or those subject to regulatory or contractual compliance. The analyst, to do this step manually, must first select appropriate safeguards for each targeted vulnerability; second, map or confirm mapping, safeguard/vulnerability pairs to all related threats; and third, determine, for each threat, the extent of asset risk mitigation to be achieved by applying the safeguard. In other words, for each affected threat, determine whether the selected safeguard(s) will reduce threat frequency, reduce threat exposure factors, or both, and to what degree.

Done manually, this step will consume many days or weeks of tedious work effort. Any "What if" assessment will be very time-consuming as well. When this step is executed with the support of a knowledge-based expert automated tool, however, only a few hours to a couple of days are expended, at most.

Safeguard Costing. In order to perform useful cost/benefit analysis, estimated costs for all suggested safeguards must be developed. While these cost estimates should be reasonably accurate, it is not necessary that they be precise. However, if one is to err at this point, it is better to overstate costs. Then, as bids or detailed cost proposals come in, it is more likely that cost/benefit analysis results, as shown below, will not overstate the benefit.

There are two basic categories of costing for safeguards:

- Cost per square foot, installed, and
- Time and materials

In both cases, the expected life and annual maintenance costs must be included to get the average annual cost over the life of the safeguard. An example of each is provided in Exhibits 8 and 9.

Cost per square foot	$165.00	
Total Square feet	50,000	
Total		$8,250,000
Safeguard Life expectancy	10 years	
Annualized cost (8,250,000/10)		$825,000
Annual Maintenance	$250,000	
Average Annual Cost		$1,075,000

Exhibit 8. Cost Per Square Foot, Installed, For a Robust New IT Facility

Cost per labor hour	$65.00	
Labor hours	480	
Implementation cost, labor		$31,200
Purchase/materials for an automated DRP tool	$29,000	
Total acquisition and implementation cost		$70,200
Safeguard life expectancy	8 years	
Annualized acquisition and implementation cost ($70,200/8)		$8,775
Annual maintenance:	$4,350	
DRP license maintenance	$32,500	
DRP staff, .5 work year (65,000 x .5)		$36,850
Average Annual Cost		$45,625

Exhibit 9. Time and Materials for Acquiring and Implementing a Disaster Recovery Plan (DRP

These Average Annual Costs represent the break-even point for safeguard cost/benefit assessment for each safeguard. In these examples, discrete, single-point values have been used to simplify the illustration. At least one of the leading automated risk assessment tools, BDSS™, allows the analyst to input bounded distributions with associated confidence factors to articulate explicitly the uncertainty of the values for these preliminary cost estimates. These bounded distributions with confidence factors facilitate the best use of optimal probabilistic analysis algorithms.

Safeguard Cost/Benefit Analysis. The risk assessment is now almost complete, though this final set of calculations is, once again, not trivial. In previous steps, the expected value of risk mitigation — the Annualized Loss Expectancy (ALE) before safeguards are applied, less the ALE after safeguards are applied, less the average annual costs of the applied safeguards — is conservatively represented individually, safeguard by safeguard, and collectively. The collective safeguard cost/benefit is represented first, threat by threat with applicable selected safeguards; and, second, showing the overall integrated risk for all threats with all selected safeguards applied. This may be illustrated as follows:

Safeguard 1 → Vulnerability 1→ n → Threat 1→ n

One safeguard may mitigate one or more vulnerabilities to one or more threats. A generalization of each of the three levels of calculation is represented below.

For the Single Safeguard. A single safeguard may act to mitigate risk for a number of threats. For example, a contingency plan will contain the loss for disasters by facilitating a timely recovery. The necessary calculation includes the integration of all affected threats' risk models before the safeguard is applied, less their integration after the safeguard is applied to define the gross risk reduction benefit Finally, subtract the safeguard's average annual cost to derive the net annual benefit.

RB(T)1 RA(T)1

[( ) – ( ) = GRRB] — SGAAC = NRRB

RB(T)n RA(T)n

Where:

RB(T) = the risk model for threats1-n *before* the safeguard is applied.
RA(T) = the risk model for threats1-n *after* the safeguard is applied.
GRRB = Gross Risk Reduction Benefit
NRRB = Net Risk Reduction Benefit
SGAAC = Safeguard Average Annual Cost

This information is useful in determining whether individual safeguards are cost effective. If the net risk reduction (mitigation) benefit is negative, the benefit is negative, i.e., not cost effective.

For the Single Threat. Any number of safeguards may act to mitigate risk for any number of threats. It is useful to determine, for each threat, how much the risk for that threat was mitigated by the collective population of safeguards selected that act to mitigate the risk for the threat. Recognize at the same time that one or more of these safeguards may act as well to mitigate the risk for one or more other threats.

$$[(\text{AALEB} - \text{AALEA} = \text{GRRB}) - \text{SGAACSG1–n}] = \text{NRRB}$$

Where:

AALEB = Average Annual loss Expectancy *before* safeguards

AALEA = Average Annual Loss Expectancy *after* safeguards

In this case, NRRB refers to the combined benefit of the collective population of safeguards selected for a specific threat. This process should be executed for each threat addressed. Still, these two processes alone should not be regarded as definitive decision support information. There remains the very real condition that the collective population of safeguards could mitigate risk very effectively for one major threat while having only minor risk mitigating effect for a number of other threats relative to their collective SGAAC.

In other words, if looked at out of context, the selected safeguards could appear, for those marginally affected risks, to be cost prohibitive — their costs may exceed their benefit for those threats. Therefore, the next process is essential to an objective assessment of the selected safeguards overall benefits:

For All Threats. The integration of all individual threat risk models for before selected safeguards are applied and for after selected safeguards are applied shows the gross risk reduction benefit for the collective population of selected safeguards as a whole. Subtract the average annual cost of the selected safeguards, and the net risk reduction benefit as a whole is established.

This calculation will generate a single risk model that accurately represents the combined effect of all selected safeguards in mitigating risk for the array of affected threats. In other words, an executive summary of the expected results of proposed risk mitigating measures is generated.

Final Recommendations. After the risk assessment is complete, final recommendations should be prepared on two levels; (1) A categorical set of recommendations in an executive summary, and (2) detailed recommendations

in the body of the risk assessment report. The executive summary recommendations are supported by the integrated risk model reflecting all threats risks before and after selected safeguards are applied, the average annual cost of the selected safeguards, and their expected risk mitigation benefit.

The detailed recommendations should include a description of each selected safeguard and its supporting cost benefit analysis. Detailed recommendations may also include an implementation plan. However, in most cases, implementation plans are not developed as part of the risk assessment report. Implementation plans are typically developed upon executive endorsement of specific recommendations.

Automated Tools

The following products represent a broad spectrum of automated risk assessment tools ranging from the comprehensive, knowledge based expert system BDSS™, to RiskCalc, a simple risk assessment shell with provision for user-generated algorithms and a framework for data collection and mapping.

- ARES, Air Force Communications and Computer Security Management Office. Kelly AFB, TX
- @RISK. Palisade Corp. Newfield, NY
- Bayesian Decision Support System (BDSS™). OPA, Inc. — The Integrated Risk Management Group, Petaluma, CA
- Control Matrix Methodology for Microcomputers. Jerry FitzGerald & Associates. Redwood City, CA
- COSSAC. Computer Protection Systems Inc. Plymouth, MI
- CRITI-CALC. International Security Technology. Reston, VA
- CRAMM. Executive Resources Association. Arlington, VA
- GRA/SYS. Nander Brown & Co. Reston, VA
- IST/RAMP. International Security Technology. Reston, VA
- JANBER. Eagon. McAllister Associates Inc. Lexington Park, MD
- LAVA. Los Alamos National Laboratory. Los Alamos, NM
- LRAM. Livermore National Laboratory. Livermore, CA
- MARION. Coopers & Lybrand (UK-based). London, England
- Micro Secure Self Assessment. Boden Associates. East Williston, NY
- Predictor. Concorde Group International. Westport, CT
- PRISM. Palisade Corp. Newfield, NY
- QuikRisk. Basic Data Systems. Rockville, MD
- RA/SYS. Nander Brown & Co. Reston, VA
- RANK-IT. Jerry FitzGerald & Associates. Redwood City, CA
- RISKCALC. Hoffman Business Associates Inc. Bethesda, MD
- RISKPAC. Profile Assessment Corp. Ridgefield, CT

- RISKWATCH. Expert Systems Software Inc. Long Beach, Ca
- The Buddy System Risk Assessment and Management System for Microcomputers. Countermeasures, Inc. Hollywood, MD

SUMMARY

While the dialogue on risk assessment continues, management increasingly is finding utility in the technology of risk assessment. Readers should, if possible, given the culture of their organization, make every effort to assess the risks in the subject IT environments using automated, quantitatively oriented tools. If there is strong resistance to using quantitative tools, then proceed with an initial approach using a qualitative tool. But do start the risk assessment process!

Work on automated tools continues to improve their utility and credibility. More and more of the "Big Accounting Firms" and other major consultancies, including those in the insurance industry, are offering risk assessment services using, or planning to use, quantitative tools. Managing risk is the central issue of information security. Risk assessment with automated tools provides organizational management with sound insight on their risks and how best to manage them and reduce liability cost effectively.

SERVER SECURITY POLICIES

Jon David

INSIDE

Server Functions, Access Control, Encryption, Logging, Disk Utilization, Backup, Communications, Server Access and Control, General (Node) Access Control, Passwords, Physical Security, Legal Considerations, Higher-Level Security

INTRODUCTION

Local area networks (LANs) have become the repository of mission-critical information at many major organizations, the information-processing backbone at most large organizations, and the sole implementation avenue for Internet protocol (IP) efforts in smaller concerns. The growing importance of LANs — the integrity and confidentiality of data and programs on them, their availability for use — demands proper security, but LANs have historically been designed to facilitate sharing and access, not for security. There is a growing pattern of interconnecting these networks, further increasing their vulnerabilities.

The Internet has similarly become an integral part of day-to-day operations for many users, to a point that business cards and letterheads often contain E-mail addresses, and a large number of organizations have their own Internet domain, organization-name.com. The World Wide Web (WWW) is an extension of the Internet, actually an additional set of functions the Internet makes readily available. It is gaining in popularity at a very fast rate, such that it is now common to see even TV advertisements cite Web addresses for additional information or points of contact (e.g., www.news-show.com, www.product-name.com, etc.). Today, even with the Web still in its infancy, there is much Web commerce, e.g., the display and purchase of merchandise. Although LANs come from a background where relatively little attention was devoted to security, the RFCs (Requests for Comment, i.e., the specifications to which the Internet conforms) specifically state

PAYOFF IDEA

By far the key element in server security is the server administrator. Regardless of what products are employed to execute server strategy policies, the quality of security correlates most highly with the abilities and the efforts of the server administrator.

that security is not provided and is therefore the sole responsibility of users. The Internet is rife with vulnerabilities, and the Web adds a further level of risks to those of the Internet.

Although servers are integral parts of various types of networks, this article will deal with LANs, not the Internet or the Web, or any other type of network. The Internet and the Web are individually and together important, and servers are particularly critical components (with PCs through mainframes being used as Web servers), but it is felt that most readers will be LAN oriented. The exposures of both the Internet and the Web differ significantly from LAN vulnerabilities in many areas, and deserve separate (and extensive) treatment on their own.

THE NEED FOR SERVER SECURITY

For a long time, information — and its processing — has been a major asset, if not *the* major asset, of large organizations. The importance of information is even reflected in the language used to refer to what originally was a simple and straightforward function: What was once known as computing became electronic data processing (EDP) and is now information processing; when expert guidance is needed in this field, people skilled in information technology (IT) are sought; in the contemporary electronic world, both the military and commercial segments fear information warfare (IW).

The information that we enter into, store on, and transmit via our computers is critical to our organizations, and in many cases it is critical not just for efficiency, profit, and the like, but to the very existence of the organization. We *must* keep prying eyes from seeing information they should not see, we *must* make sure that information is correct, we *must* have that information available to us when needed. Privacy, integrity, and availability are the functions of security.

LANs are a key part of critical information processing; servers are the heart of LANs. The need for proper server security is (or at least certainly should be) obvious.

Server/NOS vendors do not help the situation. As delivered, servers are at the mercy of the "deadly defaults." Because security tends to be intrusive and/or constraining in various ways, servers "from the box" tend to have security settings at the most permissive levels to make their networks perform most impressively.

THE NEED FOR SERVER SECURITY POLICIES

The media have been very helpful in recent years in highlighting the importance of proper information security. Although they have certainly not been on a crusade to make large operations more secure, the many security breaches experienced have made good copy and have been well publicized. Because pain is an excellent teacher, and successful or-

ganizations endeavor to learn from the pain of others, the publicizing of various breaches of information security has made everyone aware of its importance.

Successful organizations endeavor to remain successful. If they recognize a need (versus merely a nicety), they endeavor to treat it. "Go out and buy us some," and "What will it it cost?" are frequently heard once the need for security is recognized. Unfortunately, security is not something you go out and buy, it is something you plan and something you work on — when planning it, when creating it, when living with it.

Security policies are a prerequisite to proper security. They provide direction, they treat all areas necessary for proper security, and, possibly most important, because it is so rarely recognized, they provide a means for consistency. Without direction, completeness, and consistency, security can always be trivially breached. If your security efforts concentrate on securing your servers, yet you do not tell users not to have stick-on notes with their passwords on their monitors, your security policies are deficient; if you require server changes be made only at the server console, yet allow anyone other than duly authorized administrators to make such changes, you have again missed the boat in terms of security policy. And, when networks that are 100% secure in and of themselves can each compromise the others via inconsistencies in their respective security types if they are interconnected (and interconnection has been a hot item for some time), having components with proper security is no longer enough; you must have consistent security set forth in your policies.

Warning: Your policies should fit *your* operational environment and requirements. It is unlikely that the policies of even a similar organization will be best for you in every area. This does not mean that looking at the policies of other organizations cannot be of help to you — if they are good policies, of course — in terms of suggesting things like the types of areas to be treated, but you need to do what is right for you, not what may or may not have been right for somebody else.

POLICIES

Servers are parts of networks, networks are parts of information-processing structures, and information-processing structures are parts of organizational operations. Although this article deals only with server security policies, all other security areas must be dealt with in an organization's full security policies statement. A single security breach of any type can, and often does, compromise all operations. (If, for example, visitors were allowed to enter a facility unchallenged, and if nodes were left operational but unattended — during lunch periods or whatever — the best server security policies in the world would readily be defeated.)

The statements of policy set forth below are generic in nature. Not all will apply — even in modified form — to all servers, and many, if not

most, will have to be adapted to specific operations. They are, however, most likely better than those you are likely to get from friends, and should serve as a good start for, and basis of, proper server security policies for your particular situation. For convenience, they are grouped in functional areas.

One area, and possibly the most critical one, will not be covered: the LAN security administrator. Your security cannot be any better than your administrators make and maintain it. You require the best possible personnel, and they must be given the authority, and not just the responsibility, to do whatever is necessary to provide the proper server — and network — security. Too often we see "the Charlie Syndrome": LANs come in, administrators are needed, Charlie is free, so Charlie is made the system administrator. Why is Charlie free? Well, in all honesty, it is because Charlie is not good enough to be given anything worthwhile to do. What this means is that rather than having the best people as system administrators, the worst are too frequently in that position — system administration should not be a part-time assignment for a secretary!

SERVER FUNCTIONS

Access Control

- The server shall be able to require user identification and authentication at times other than log-on.
- Reauthentication shall be required prior to access to critical resources.
- File and directory access rights and privileges should be set in keeping with the sensitivity and uses of the files and directories.
- Users should be granted rights and privileges only on a need-to-know/use basis (and not be given everything except the ones they are known not to need, as is very commonly done).

Encryption

- Sensitive files should be maintained in encrypted form. This includes password files, key files, audit files, confidential data files, etc. Suitable encryption algorithms should be available, and encryption should be able to be designated as automatic, if appropriate.
- For any and every encryption process, cleartext versions of files encrypted must be overwritten immediately after the encryption is complete. This should be made automatic, effectively making it the final step of encryption.

Logging

- Audit logs should be kept of unsuccessful log-on attempts, unauthorized access/operation attempts, suspends and accidental or deliber-

ate disconnects, software and security assignment changes, log-ons/log-offs, other designated activities (e.g., accesses to sensitive files), and, optionally, all activity.
- Audit log entries should consist of at least resource, action, user, date and time, and, optionally, workstation ID and connecting point.
- There should be an automatic audit log review function to examine all postings by posting type (illegal access attempt, access of sensitive data, etc.), and for each posting type. If a transaction threshold (set by the LAN administrator) for any designated operation exception is exceeded, an alarm should be issued and an entry made in an action-item report file.
- The audit file should be maintained in encrypted format.
- There should be reporting functions to provide user profiles and access rules readily and clearly, as well as reports on audit log data.

Disk Utilization

- As appropriate to their sensitivity, ownership, licensing agreements, and other considerations, all programs should be read-only or execute-only, and/or should be kept in read-only or execute-only directories. This should also apply to macro libraries.
- Users should be provided with private directories for storage of their nonsystem files. (These include files that are shared with other users.)
- There should be no uploads of programs to public areas; the same is true for macros and macro libraries.

Backup

- The availability of the LAN should be maintained by the server scheduling and performing regular backups. These backups should provide automatic verification (read-after-write), and should be of both the full and partial (changed items only) varieties. All security features, including encryption, should be in full effect during backups.
- Both backups and the restore/recovery functions should be regularly tested.
- Backups should be kept off premises.
- Automatic recovery of the full LAN and of all and individual servers (and workstations) must be available.

Communications

- Communications (i.e., off-LAN) access should be restricted to specific users, programs, data, transaction types, days/dates, and times.
- An extra layer of identification/authentication protocol should be in effect (by challenge-response, additional passwords, etc.) for communications access.

- All communications access should be logged.
- All communications access messages should be authenticated, using message authentication codes (MACs), digital signatures, etc.
- The password change interval should be shorter for communications access users.
- Stronger encryption algorithms should be used for communications access users.
- Any and all confidential information — passwords, data, whatever — should be encrypted during transmission in either or both directions for all communications access activities.
- Encryption capabilities for communications should include both end-to-end and link encryption.

Server Access and Control

- There shall be no remote, i.e., from other than the console, control of any kind of the server, and there shall similarly be no remote execution of server functions.
- All server functions must be done from the console. This specifically excludes access via dial-in, gateways, bridges, routers, protocol converters, PADs, micro-to-mainframe connections, local workstations other than the server console, and the like.
- All administrator operations (e.g., security changes) shall be done from the console.
- Supervisor-level log-on shall not be done at any device other than the console.
- If supervisor-level use of a device other than the console becomes necessary, it shall be done only after a boot/restart using a write-protected boot diskette is certified as "clean" (this implies that such diskettes are readily available, as they should be for even stand-alone PCs), or from tape.
- There shall be no user programs executed at the server by user (i.e., remote or local workstation) initiation.
- There shall be no immediate workstation access to the server or to any server resources following a diskette boot at the server.
- All communication among and between nodes must be done through the server. There shall be no peer-to-peer direct communication.
- There shall be no multiple user IDs (UIDs)/passwords logged on (i.e., the same user on the system more than once at a given time). There should also be the ability to suspend the active user session and/or issue alarms should this situation occur.

General (Node) Access Control

- Both a user ID and a password shall be required by servers for a user as part of logging on.

- The server should be able to identify both the workstation and workstation connection point at log-on.
- All files (programs and data) and other resources (peripheral equipment, system capabilities) should be able to be protected.
- All resource access should be only on a need-to-know/need-to-use basis.
- File access control should be at file, directory, and subdirectory levels.
- File access privileges should include read, read-only, write (with separate add and update levels), execute, execute-only, create, rename, delete, change access, none.
- Resource access should be assignable on an individual, group, or public basis.

Passwords

- There should be appropriate minimum (6 is the least, 8 is recommended, more is better) and maximum (at least 64, more is better) lengths. (Longer "passwords" are usually "pass-phrases," e.g., "Four score and 7 years ago.")
- Passwords should be case sensitive.
- There should be a requirement for at least one uppercase character, one lowercase character, one numeric, and one alphabetic character to be used in user-selected passwords. For high-security access, this should be extended to include one nonprint (and nonspace) character.
- There should be computer-controlled lists of prescribed passwords to include common words and standard names, and employee/company information as available (name, address, social security number, license plate number, date of birth, family member names, company departments, divisions, projects, locations, etc.). There should also be algorithms (letter and number sequences, character repetition, initials, etc.) to determine password weakness.
- Passwords should be changed frequently; quarterly is a minimum — monthly is better. High security access should have weekly change.
- There should be reuse restrictions so that no user can reuse any of the more recent passwords previously used. The minimum should be 5, but more is better, and 8 is a suggested minimum.
- There should be no visual indication of password entry, or password entry requirements. This obviously prohibits the password characters from echoing on the screen, but also includes echoing of some dummy character (normally an asterisk) on a per character basis, or used to designate maximum field length.
- New passwords should always be entered twice for verification.
- LAN administrators, in addition to their passwords with associated supervisory privileges, should have an additional password for "normal" system use without supervisory privileges.

Note: There are password test programs to allow automatic review and acceptance/rejection of passwords. These are usually written in an easily ported language, typically C, and can be readily structured to implement whatever rules the security administrator feels are appropriate. They are used between the password entry function and the password acceptance function already in place, so only proper passwords get used by the system.

Physical Security

- All servers should be as secured as possible in keeping with their sensitivity.
- Servers should be physically secured in locked rooms.
- Access to servers should be restricted to authorized personnel.
- Access to the server area should be automatically logged via use of an electronic lock or other such mechanism as appropriate.
- The room in which the server is kept should be waterproof and fireproof.
- Walls should extend above the ceiling to the floor above.
- Water sprinklers and other potentially destructive (to computers) devices should not be allowed in the server room.
- The server console should be kept with the server.
- Servers should have key locks.
- Connection points to servers should be secured (and software-disabled when not in use) and regularly inspected.
- All cabling to servers should be concealed whenever possible. Access to cabling should be only by nonpublic avenues.
- All "good" media practices — encryption of sensitive information, storage in secure locations, wiping/overwriting when finished, etc. — should be in full effect.

Legal Considerations

- Programs that by license cannot be copied should be stored in execute-only or, if this is not possible, read-only directories, and should be specifically designated as execute-only or read-only.
- Concurrent use count should be maintained and reviewed for programs licensed for a specific number of concurrent users. There should be a usage threshold above which additional concurrent access is prohibited.
- Access rules should be reviewed for all programs licensed for specific numbers of concurrent users.
- Appropriate banner warnings should be displayed as part of the logon process prior to making a LAN available for use.
- Appropriate warning screens should be displayed on access attempts to sensitive areas and/or items.

Other

- There shall be no unauthorized or unsupervised use of traffic monitors/recorders, routers, etc.
- There should be a complete formal and tested disaster recovery plan in place for all servers. This should include communications equipment and capabilities in addition to computer hardware and software. (This is, of course, true for full LANs, and for the entire IP operations.)
- There shall be no sensitive information ever sent over lines of any sort in cleartext format.
- Servers should require workstations that can also function as standalone PCs to have higher levels of PC security than those PCs that are not connected to a LAN. Workstations that operate in unattended modes, have auto-answer abilities, are external to the LAN location (even if only on another floor), and/or are multiuser should have the highest level of PC security.
- Workstation sessions should be suspended after a period of inactivity (determined by the LAN administrator), and terminated after a further determined period of time has elapsed.
- Explicit session (memory) cleanup activities should be performed after session disconnect, whether the session disconnect was by workstation request (log-off), by server initiative (such as due to inactivity), or accidental (even if only temporary, as might be the case with a line drop).
- In cases where session slippage tends to occur (such as line drops), or in instances where service requests require significant changes of access level privileges, reauthentication should be required.
- Unused user IDs and passwords should be suspended after a period of time specified by the LAN administrator.
- Successful log-ons should display date and time of last log-on and log-off.
- There should be the ability to disable keyboard activity during specified operations.
- The integrity of data should be maintained by utilization of transaction locks on all shared data — both data files and databases.
- The integrity of data and the availability of data and the entire LAN should be maintained by specific protections against viruses and other malicious code.
- All security functions and software changes/additions should be made only from the server and only by the LAN administrator.

HIGHER-LEVEL SECURITY

Although the preceding capabilities will be significantly more than most LAN servers would find appropriate, there are still more sophisticated se-

curity features that are appropriate to LANs with high-risk/high-loss profiles. For the sake of completeness, major ones are set forth in the following:

- Access to critical resources should require reauthenticaton.
- Access to critical resources should not only authenticate the user, but further verify the correctness of the workstation in use, the connection point of that workstation, and the correctness of the day/date/time of the access.
- Message sequence keys should be used to detect missing or misordered messages.
- After a failed log-on attempt, the server should generate an alarm, and be able to simulate a proper log-on for the failed user (to keep this user connected while personnel go to the offending workstation).
- After excessive access violations, the server should generate an alarm, and be able to simulate a continuing session (with dummy data, etc.) for the failed user (to keep this user connected while personnel go to the offending workstation).
- Traffic padding — the filling in of unused transmission bandwidth with dummy pseudo-traffic — should be used to prevent transmission patterns from being readily detected (thereby making it easier to "trap" valid information).
- Multiple — at least two — LAN administrators should be required for all potentially critical server changes. (These might be adding a new user, altering an existing user's rights and privileges, changing or adding software, and the like.) For example, one administrator could add a user, but only from a list internal to the computer that a second administrator created. This means that any deliberate breach of security by an administrator would require collusion to be effective.
- LAN administrators should have separate passwords for each individual server function they perform, the rights and privileges associated with that password being the minimum necessary to do the specific job for which it is being used.
- The server should be fully compatible with tokens, biometric devices, and other such higher-security access control products and technologies.
- The server should be able to do automatic callback for any and all communications access.
- To improve the availability of the LAN, it should be fault tolerant. Multiple (shadow) servers, disk mirroring, and the like should be in place.
- There should be a file/system integrity product in regular and automatic use to alert the administrator to any and all server changes.

- Sophisticated authentication methodologies should be in place to assure not only the contents of a message/request, but also the source. MACs and digital signatures are viable means to certify contents, and public key/private key (commonly known as RSA-type) encryption provides acceptable source verification.
- Backups should be made to an off-LAN facility. This could be an organizational mainframe, a service bureau, or whatever. With this "store and forward backup," recovery media is immediately away from the server.
- Servers should be compatible with biometric devices (fingerprint, retinal scan, palm print, voice, etc.) for user verification.

CAVEATS

Seat belts, air bags, and other automotive safety devices merely make it less likely that you will be seriously injured in an accident, and certainly do not guarantee your not being involved in one. By the same token, computer security merely lessens the chances your systems will be misused, lessens the likelihood of damages associated with certain common incidents, makes it more likely to discover and limit any misuse and/or damages promptly, and makes it easier to recover from various types of both accident and misuse.

No realistic computer security "can't be beaten," and this certainly includes server security. Proper server security will make networks much more difficult to compromise, and can make it not worth an intruder's time (in terms of anticipated cost to break in versus expected return as a result of a break-in) to even attempt to break into a properly secured network.

With servers viewed as being in the hands of "experts" (which they often are, of course), many, if not most, users rely exclusively on server security for total protection, and do not practice proper security as part of their operations. Server security, and even full network security, is not a substitute for other types of security; your security policies must reflect this.

TEETH

The best policies in the world will fail if they are not enforced. ("Thou shalt not print your password on a stick-on note and post it to your monitor" sounds good, but people still tend to do it — If you don't make sure that they don't, or take proper corrective actions if they do, your policies are little more than a waste of paper.) Your policies should have teeth in them: as appropriate, server, as well as all other security policies, should contain monitoring and enforcement sections.

Because operational environments are often in a virtually continuous state of change — new equipment and users, changing capabilities,

rights and privileges, etc. — you should regularly review your server (and full) security to make sure it continues to be in agreement with your server security policies.

Similarly, untested server security may only be security on paper. Because even the most qualified personnel can make mistakes in creating server security policies and/or in implementing them, your security should be tested to see that it really works as intended and conforms to your server security policies. This should obviously be done when you design/develop/install your security, but should also be done on a reasonable periodic basis (quarterly, yearly, whatever). Such tests are usually done best by outsiders, because truly capable personnel often are not available on staff, and employees often have friends to protect and personal interests in particular operations.

CONCLUSION

LANs have become critical processing elements of many, if not most, organizations of all sizes, and servers are the hearts of LANs. As the frequent repository of highly sensitive, often mission-critical information, proper security is of prime importance. Without proper security policies, security is unlikely to succeed, and policies have to be in place to allow the appropriate security to be designed and installed. Adequate security can be obtained by companies willing to work at it, and the work must start with proper security policies and must continue by seeing that security continues to conform to existing security policies. The key element by far in LAN security is the LAN administrator; for all purposes, and in spite of whatever products you may purchase, the quality of security will be in one-to-one correspondence with the abilities and efforts of this person.

Jon David is an independent consultant with more than 20 years experience in system and network security. He is an expert in Internet and WWW security. His clients include major financial organizations, top Fortune companies, and key government agencies.

78

Toward Enforcing Security Policy: Encouraging Personal Accountability for Corporate Information Security Policy

John O. Wylder, CISSP

Information security professionals through the years have long sought support in enforcing the information security policies of their companies. The support they have received has usually come from internal or external audit and has had limited success in influencing the individuals who make up the bulk of the user community. Internal and external auditors have their own agendas and do not usually consider themselves prime candidates for the enforcement role.

Other attempts to achieve policy enforcement have included rounding up the usual suspects of senior management and executive management memoranda and security awareness campaigns. In general, none of these programs were felt to be successful, as evidenced by routine tests of information security policy compliance. This chapter discusses a new approach to policy enforcement. The proposal is to encourage the support for these policies by incorporating compliance activities with an individual's annual personnel performance evaluation.

Background

The successful implementation of an information security program derives from a combination of technical and nontechnical efforts. The process starts with the development of a comprehensive plan that assesses the risks and threats to an individual firm and then moves to the development of a set of policies and strategies to mitigate those risks. These policies are often a mix of technical and nontechnical items that require routine testing or measurement to ensure that the desired level of compliance is maintained over time. In most cases, the technical policies are the initial focus of a security program and are done in cooperation with information technology (IT) staff. This is the traditional home of information security practitioners.

The Problem

Most security practitioners are aware that the bulk of their problems are internal rather than external. Whatever their level in the organization and regardless of the degree of support they feel they have or do not have within

EXHIBIT 78.1 PricewaterhouseCoopers Survey

There was a recent survey by PricewaterhouseCoopers of 1000 companies in the United Kingdom. The survey found the majority of companies spent, on average, less than 1 percent of their total IT budget on information security while an average of 3 to 4 percent was recommended.
Paradoxically, it said that 73 percent of senior managers interviewed believed that IT security was a top priority.
Potter said: "The board of most companies want to do something about security but it does not know how much money it should spend on it." The survey was commissioned by the Department of Trade and Industry.

the organization, it has become clear over time that Pareto's law applies here: 80 percent of the problems are caused by 20 percent of the people.

Pentasafe Security Technologies recently conducted a survey among companies and found that nine out of ten employees were likely to open and execute an e-mail attachment without questioning its source or authenticity. This leads, of course, to virus and worm infections on the corporate e-mail server. Why do people do this despite the widespread publicity that such infections have received? Is it the lack of awareness, as some might say, or is it the lack of understanding the consequences of failing to comply with security policy?

Companies have tried a variety of means to ensure that their employees have received at least minimal training in information security policies. Here is a list of some of those approaches:

- Inclusion of security policies in employee handbooks
- Requirement to take a self-study course prior to initial issuance of user credentials
- Annual testing of security awareness
- PR campaigns using posters and Web and e-mail reminders

All of these are valid approaches and should be considered as a part of the security program for any company. Yet despite these types of programs, security practitioners still find that users fail in routine functions of security and still choose passwords, for example, that are easily guessed or even shared. Raising the bar on having complex passwords that must be changed frequently usually results in passwords that are written on notepads and left underneath the keyboard.

When employees are interviewed about their lack of compliance, they often cite the pressure to be productive and that they see the incremental security policy as counter to their productivity. When it comes to complying with security and trying to be productive, most users err on the side of productivity rather than security. This leads to the question of how you make employees personally accountable for their role in compliance with information security policy.

Some security professionals say that the problem starts at the top with a lack of awareness and support by the executive team. There is some truth to that, as the budget and resource allocation starts at the top and if there is no money, there is little chance that the security program will succeed (see Exhibit 79.1).

In some companies, a new approach emerged in the late 1980s, that is, the creation of a "C"-level position for security, that of the Chief Information Security Officer. The thinking was that by elevating the position to a peer with the other "C"-level positions, it would be easier for those people to gain compliance with their policies. By giving them a seat at the table, they would be in a better position to ensure that their policies are ones that have the full support of the management team.

The Role of the Chief Information Security Officer (CISO)

Recently, there has been a resurgence in the movement to create the position of Chief Information Security Officer (CISO) that reports to the CIO or at least to the CTO. Another recent innovation is to create a Chief Privacy Officer (CPO), either in addition to or instead of a CISO. All too often, this has been done due to poor results shown in audits of the compliance with the existing policies. The higher-level reporting structure is seen as a way to better ensure that information security receives the proper level of management attention. Creation of the new position alone, however, has not been shown to be the way to ensure policy compliance across the enterprise.

Many companies today have some form of matrix management in place. In one company this author recently worked with, the Chief Security Office had responsibility for security policy from both a creation and an enforcement standpoint, but only had dotted-line responsibility for the tactical side of information security. In that company, the technical policies were done first by and for the IT department and then rolled out into

either the employee manual or into the company's corporate-wide compliance manual. It is this set of policies that became the more difficult ones to assess and to ensure compliance, despite its corporate-wide distribution.

This split is not atypical today. The responsibility for administering passwords and user credentials is often part of the technology area. In some cases, these responsibilities may even go to a network help desk for administration. There may be nothing wrong with this approach but the measurement of compliance with policy is often overlooked in this case. The security administrator is measured by things like password resets and log-in failures, but who is measuring why those passwords need to be reset and who is responding to any audits of the strength and quality of the passwords?

Security Policy and Enforcement

One of the standard descriptions of information security programs is that they are about "people, policies, and procedures." In developing the policies for a company, this is taken to the next level down and the process is then about creating a risk profile and developing the appropriate policies to reduce risk. Once the policies are created, the appropriate implementation mechanisms are put in place and then come the controls that allow the measurement and enforcement of those policies.

Technology-Based Enforcement

For example, the risk profile of a company with product trade secrets will logically be different from the risk profile of a company that is in the services business. The company with the trade secrets has high-risk information that needs to be kept secure and it may have a detailed information classification policy as part of its Information Security Policy manual. Along with information classification, it may also have role-based access controls that allow it to implement the classification policy. This then may lead it to the implementation of certain technologies that allow automated controls and enforcement of the information classification and access control policy. This can then be described as technology-based enforcement. The access control system, once properly implemented, allows or prevents access to information and enforces the policy.

There are many good examples of this approach in the marketplace today. This approach sometimes comes under the title of "Identity Management." It addresses a broad spectrum of controls, including authentication and authorization systems. Included here are such technologies as biometrics, smart cards, and more traditional access control systems. Enforcement is achieved through approval or denial of access and reporting of policy violations through error or audit logs.

Executive Enforcement

Frequently cited in articles on the creation of an effective information security program is the need for support by executive management. This is sometimes seen as the route to enforcement of policy. Comments heard from many information security professionals include, "I need the president of the company to come out in favor of our policies, then I can get people to comply with them." There is a fallacy here because executive management is too far removed from the day-to-day operations of a company to become enmeshed in the enforcement of any given policy or policies. It is unlikely that the president of a large or even a medium-sized company can be brought into the discussion of the virtues of maintaining role-based access controls as opposed to broad-based access. This type of discussion is usually left to the operational areas to work out among them.

It is possible to get the support of the executive team to send the message to all employees about their support for the information security program. That executive support can, in fact, be essential to the information security department as it goes out and spreads its message. It is very difficult, on the other hand, to translate that support into direct action on the enforcement of specific policies.

Audit as Enforcement

The auditing department of a company is often seen as part of the enforcement mechanism and sometimes may be seen as the primary enforcement tool. Most auditors disagree that they should play an operational role and try to keep their "enforcement" role to a minimum. This is often done by auditing the existence of policy, measuring the effectiveness of the policy, and leaving the role of enforcement to others. For example, auditors would look at whether or not there were policies governing the role-based access to classified information.

They then may drill down and test the effectiveness of the administration of such policies. Their finding would be one of fact: "We tested the authorization policies of the XYZ department. We found that ZZ members of the department had complete read, write, and update authority to the system. This appears to be inappropriate based on the job description of those people. We recommend that management review the access list and reduce it to the minimum number of people necessary to perform those critical job functions and that access be granted based on the job description on file with the HRMS department."

This type of finding is typical of most auditors' roles and does not lend itself to assisting with the enforcement of policy. For example, in the above case, there is neither a finding that indicates who created the violations, nor is there a finding of what actions should be taken to ensure that that person is admonished for creating the violations.

Traditional Management Enforcement

The remaining place in an organization that most people look to for enforcement of policy is to the individuals managing the various corporate departments. Enforcement of information security policies here comes under the broad heading of enforcement of all corporate-level policies. Managers, like their employees, have to juggle the sometimes-conflicting need to enforce policies while maintaining productivity. Sometimes, employees see the need to have access beyond their normal approved level as a means to improve their job performance. In other cases, there may be conflicting messages sent by management about which company goals have priority. In any case, this model is one of distributed enforcement, which can lead to uneven application of policy and compliance.

All of the above methods have been tried through the years with varying degrees of success. Few people active today in the information security field have great confidence that their enforcement mechanisms are working to their satisfaction.

Policy Compliance and the Human Resources Department

In asking a security manager if it would make any difference if security compliance were to become part of the employee annual performance assessment process, the response was that "it would make all the difference in the world." During the same engagement, the human resources (HR) manager was asked if his department could help enforce information security policies; his response was, "No way!"

The HR manager explained that policies to them were a zero-sum game; if a new policy were to be added, they needed to consider which policy would be dropped. They understood that compliance could become one of their responsibilities and then said that they already had to measure compliance with policies covering attendance, hiring practices, privacy, pay equity, and a host of others. Which policy should they drop to help with the compliance to security policy?

They had a good point, but I then asked what would happen if we added it as a job-performance criterion. Suddenly there was a change in attitude and an understanding that perhaps a middle ground could be found where compliance could be brought into existing policies and procedures.

The problem then is how to accomplish this and how to maintain the support of the human resources professionals. The remainder of this chapter explores this idea and proposes a possible means to accomplish this through the development of an annual personal information security plan by each employee.

The Personal Security Plan

The HR people in that engagement gave a glimmer of hope that security could become part of performance appraisals and therefore compliance with policies could not only be measured but could be enforced at some level. Most employees understand the old adage that what is measured gets done. If the company provides a way to report on employee compliance with any policy and links it to performance measurement and compensation, then company employees are more likely to comply with that policy.

Personal Accountability

A new term has popped up recently in the press with respect to IT practices — and that is *accountability.* This has come up with some of the recent legal actions where victims of poor IT practices are filing suits against

companies that may not be the perpetrator, but whose own practices may be part of the problem. There was a recent action in which a denial-of-service (DoS) attack occurred and a lawsuit was filed against an Internet service provider (ISP) whose network was used by the perpetrators to launch a zombie DoS attack. This case is still moving through the court system and the outcome at this point is undetermined, but the net effect is to try to shift the burden of blame to people who fail to practice safe computing. This philosophy can then be used in another way to help shift the focus of enforcement of policy from management, audit, or technology to the individual.

This idea recently received a boost with the backing of professionals in the U.S. Government:

> "Federal agencies must raise staff accountability for breaches and demand security become standard in all network and computing products. Otherwise, enterprises won't be able to improve cyber attack response and prevention, according to highlights of a recent conference sponsored by the National High Performance Computing and Communications Council.
>
> Rather than emphasizing technology's role in information security, several speakers urged stronger user awareness programs and more involvement of top management."
>
> "You can't hold firewalls and intrusion detection systems accountable. **You can only hold people accountable**," said Daryl White, chief information officer for the U.S. Department of the Interior, in a published report (emphasis added).

The Personal Security Plan: Phase One

Using this approach, the proposal being made here is the creation of a personal security plan and the incorporation of that plan into an employee's annual performance appraisal.

Exhibit 79.2 shows an example of such a plan. This is a simple document that addresses the basic but core issues of security. It is neither highly technical nor does it require the company to invest money in any large-scale implementation of technical solutions such as biometrics, Public Key Infrastructure (PKI), or any other simple or even exotic technologies. The emphasis here is on the routine things an employee does that can create risk to the company.

However, the items to be measured include the need to track compliance at a technical level. It is not practical to just rely on the employee writing a plan and taking a pledge of compliance. It is important that the technical approaches to compliance be used and the results included in the evaluation of the effectiveness of the plan. These should not come as any surprise to a security professional, and the tools should be part of their arsenal:

- *Password cracking programs:* measuring the strength of the passwords used by the employees
- *Log-in tracking reports:* recording the number of times the user tried to log in remotely and succeeded or failed
- *Network security audits*: tracking the use of dial-up lines or DSL access

All of these would produce data that would then be sent to the employee's supervisor for use in the annual performance appraisal.

The idea here is to broaden the focus on information security policies in the mind of the employee. By making each individual employee accountable for making and executing a Personal Security Plan, each employee then has a stake in the process of practicing safe computing at his or her company. Employees also have to become more knowledgeable about the effects of their actions on the state of security as a whole.

How the Plan Would Work

Prior to his or her annual performance review each year, each employee would be required to complete a Personal Security Plan. The plan would be designed in conjunction with the company's Information Security Policies, which would dictate key items such as remote access policies, password policies, and secure computing standards. The individual's plan would consist of his own usage profile plus his written plans for the year to use corporate computing resources in compliance with the published Information Security Policies.

For example, people who work from home using dial-up lines might be required to use a smart card or other two-factor authentication scheme as part of their access methodology. This may be combined with the use of a personal firewall and installation of anti-virus software. Employees would then use this form to describe

EXHIBIT 78.2 Personal Information Security Plan

XXX Company
Personal Information Security Plan
Date:

Plan period — From:_____ **To:**_____

Employee Name:_____

Network user ID:_____

Home computer profile:_____

Computer make, type:_____

Home ISP: AOL ☐ WorldNet ☐ CompuServe ☐ Earth link ☐ Other ☐

Access type: Dial-up ☐ DSL ☐ Cable modem ☐

Number of times a week used for work:

Home network (if applicable): Ethernet ____ Token ring ___ Wireless ____

Home protection profile (please describe methodologies or technology used at home to protect computers and networks):

Anti-virus software (vendor, version): _____

Personal firewall (vendor, version):_____

Other:_____

_____ _____
Employee signature Manager's Signature

This section to be completed by supervisor:

From annual security audit describe any security violations or compliance issues:

_____
_____
_____

their remote access profiles and how they are going to comply with corporate-wide policies. Another aspect of the plan would be for the employees to sign a notice that they understand and comply with the corporate Information Security Plan. This annual certification can become important if the employee is ever investigated for a violation.

Once this form is completed, the employees would give it to their supervisors for approval. The supervisors would be required to review the plan to ensure that it complies with corporate standards. Once approved, a copy of the plan is given back to the employees for their files and the original is kept on file with other vital employee records. The plans would be useful to the Chief Information Security Officer to use to check for overall compliance at the department and division levels.

Enforcement of the Personal Security Plan

Enforcement of the approach would be similar to the managerial approach but much more focused and specific. All employees would have to have a plan, and the effectiveness of both individual plans and the process as a whole could be measured and managed. Employees would know that their job performance and compensation would now be linked to their individual plan. HRMS should be satisfied with this approach because it is not the enforcer of the Information Security Plan, merely of the compliance mechanism. Audit likewise would be satisfied with this approach because it is measurable and has clear lines of accountability that can be measured.

EXHIBIT 78.3 Seven Simple Computer Security Tips for Small Business and Home Computer Users

Consult www.nipc.gov for more information.

- **Use strong passwords.** Choose passwords that are difficult or impossible to guess. Give different passwords to all accounts.
- **Make regular backups of critical data.** Backups must be made at least once each day. Larger organizations should perform a full backup weekly and incremental backups every day. At least once a month, the backup media should be verified.
- **Use virus protection software.** That means three things: having it on your computer in the first place, checking daily for new virus signature updates, and then actually scanning all the files on your computer periodically.
- **Use a firewall as a gatekeeper between your computer and the Internet.** Firewalls are usually software products. They are essential for those who keep their computers online through the popular DSL and cable modem connections but they are also valuable for those who still dial in.
- **Do not keep computers online when not in use.** Either shut them off or physically disconnect them from Internet connection.
- **Do not open e-mail attachments from strangers,** regardless of how enticing the Subject Line or attachment may be. **Be suspicious of any *unexpected* e-mail attachment from someone you *do* know** because it may have been sent without that person's knowledge from an infected machine.
- Regularly download security patches from your software vendors.

Finally, information security professionals should be the happiest of all because they will now have a way to bring the entire organization into the process of Information Security Policy compliance and enforcement.

Each company using this approach is responsible for matching the results to any actions taken with respect to the employee's performance appraisal. The weight that the Personal Security Plan carries for appraisal purposes will vary from company to company. In cases where there is a high-risk profile, the plan will logically carry more weight than in low-risk profile positions. Failure to complete the plan or failure to execute the plan then becomes the negative side of enforcement, requiring disciplinary action to be taken on the part of the responsible manager.

This alone will not end all risk to the company, nor can it be a substitute for technical approaches to solving technology problems. What this can do is move the responsibility to the point closest to compliance — that is, the actual employee required to comply with the policy.

Support for This Idea

The National Infrastructure Protection Center (NIPC) recently published some simple security tips (see Exhibit 79.3) that fit this strategy.

These tips could become the basis of any company's personal strategy to be used to educate employees on their responsibilities. They then become the core elements to be used in the creation of that company's version of a Personal Security Plan.

These plans would need to be updated on an annual basis and the various items in the plan would be updated as both the employees' usage changes and as technology changes. But once the process begins, the changes become a routine part of the employee's duties.

The Personal Security Plan: Phase 2

This program could be expanded in a second phase to take into account actual job-performance-related criteria. The first phase concentrates on the employee's personal computer usage and extends to any off-site access of the company network. In the next phase you could add details about the employee's current usage of information and computers while at work.

The following elements could be added to the plan in this phase:

- Access level (public, confidential, private, secret)
- Authorization level (read, write, update)
- System level access, if any (supervisor, operator, analyst)

This would make an excellent tie-in to the company's identity management program, whereby the access rules are provisioned based on job profile. The security plan for the individual would then have components that describe the access rules, authorization levels, and a record of compliance with those rules. This would be much more specific and would require more time on the part of the supervisor. The supervisor would be required to review violation and audit logs and track any violations that occurred during the planning period

The advantage of this approach is that it would bring employees full circle in their understanding of their roles and rights for information access to their actual experiences and performances. This is again aimed at getting individual accountability and making that the key element of the enforcement process.

Conclusion

The title of this chapter is "Toward Enforcing Information Security Policy." In no way is this approach intended to be the endpoint of the journey to getting full enforcement of an information security policy. This approach gives the security professional a practical way to move enforcement of security policy further along in an organization. It also moves enforcement from a top-down model to a bottom-up model and takes into account individual accountability for policy compliance.

By going beyond awareness and enlisting the assistance of other areas such as Human Resources, security policy becomes a routine part of the job rather than the exception. By making it routine and including it in the measurement of compliance with other more traditional policies, it becomes more feasible to expect that the goal of compliance will be achieved. After all, the goal is compliance, and enforcement is only the mechanism.

79

The Common Criteria for IT Security Evaluation

Debra S. Herrmann

This chapter introduces the Common Criteria (CC) by:

- Describing the historical events that led to their development
- Delineating the purpose and intended use of the CC and, conversely, situations not covered by the CC
- Explaining the major concepts and components of the CC methodology and how they work
- Discussing the CC user community and stakeholders
- Looking at the future of the CC

History

The Common Criteria, referred to as "the standard for information security,"[1] represent the culmination of a 30-year saga involving multiple organizations from around the world. The major events are discussed below and summarized in Exhibit 79.1.

A common misperception is that computer and network security began with the Internet. In fact, the need for and interest in computer security or COMPUSEC have been around as long as computers. Likewise, the *Orange Book* is often cited as the progenitor of the CC; actually, the foundation for the CC was laid a decade earlier. One of the first COMPUSEC standards, DoD 5200.28-M,[2] *Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource-Sharing ADP Systems*, was issued in January 1973. An amended version was issued June 1979.[3] DoD 5200.28-M defined the purpose of security testing and evaluation as:[2]

- To develop and acquire methodologies, techniques, and standards for the analysis, testing, and evaluation of the security features of ADP systems
- To assist in the analysis, testing, and evaluation of the security features of ADP systems by developing factors for the Designated Approval Authority concerning the effectiveness of measures used to secure the ADP system in accordance with Section VI of DoD Directive 5200.28 and the provisions of this Manual
- To minimize duplication and overlapping effort, improve the effectiveness and economy of security operations, and provide for the approval and joint use of security testing and evaluation tools and equipment

As shown in the next section, these goals are quite similar to those of the Common Criteria.

The standard stated that the security testing and evaluation procedures "will be published following additional testing and coordination."[2] The result was the publication of CSC-STD-001–83, the *Trusted Computer*

EXHIBIT 79.1 Timeline of Events Leading to the Development of the CC

Year	Lead Organization	Standard/Project	Short Name
1/73	U.S. DoD	DoD 5200.28M, ADP Computer Security Manual — Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource Sharing ADP Systems	—
6/79	U.S. DoD	DoD 5200.28M, ADP Computer Security Manual — Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource Sharing ADP Systems, with 1st Amendment	—
8/83	U.S. DoD	CSC-STD-001–83, Trusted Computer System Evaluation Criteria, National Computer Security Center	TCSEC or *Orange Book*
12/85	U.S. DoD	DoD 5200.28-STD, Trusted Computer System Evaluation Criteria, National Computer Security Center	TCSEC or *Orange Book*
7/87	U.S. DoD	NCSC-TG-005, Version 1, Trusted Network Interpretation of the TCSEC, National Computer Security Center	TNI, part of Rainbow Series
8/90	U.S. DoD	NCSC-TG-011, Version 1, Trusted Network Interpretation of the TCSEC, National Computer Security Center	TNI, part of Rainbow Series
1990	ISO/IEC	JTC1 SC27 WG3 formed	—
3/91	U.K. CESG	UKSP01, UK IT Security Evaluation Scheme: Description of the Scheme, Communications–Electronics Security Group	—
4/91	U.S. DoD	NCSC-TG-021, Version 1, Trusted DBMS Interpretation of the TCSEC, National Computer Security Center	Part of Rainbow Series
6/91	European Communities	Information Technology Security Evaluation Criteria (ITSEC), Version 1.2, Office for Official Publications of the European Communities	ITSEC
11/92	OECD	Guidelines for the Security of Information Systems, Organization for Economic Cooperation and Development	—
12/92	U.S. NIST and NSA	Federal Criteria for Information Technology Security, Version 1.0, Volumes I and II	Federal criteria
1/93	Canadian CSE	The Canadian Trusted Computer Product Evaluation Criteria (CTCPEC), Canadian System Security Centre, Communications Security Establishment, Version 3.oe	CTCPEC
6/93	CC Sponsoring Organizations	CC Editing Board established	CCEB

12/93	ECMA	Secure Information Processing versus the Concept of Product Evaluation, Technical Report ECMA TR/64, European Computer Manufacturers' Association	ECMA TR/64
1/96	CCEB	Committee draft 1.0 released	CC
1/96 to 10/97	—	Public review, trial evaluations	—
10/97	CCIMB	Committee draft 2.0 beta released	CC
11/97	CEMEB	CEM-97/017, Common Methodology for Information Technology Security, Part 1: Introduction and General Model, Version 0.6	CEM Part 1
10/97 to 12/99	CCIMB with ISO/IEC JTC1 SC27 WG3	Formal comment resolution and balloting	CC
8/99	CEMEB	CEM-99/045, Common Methodology for Information Technology Security Evaluation, Part 2: Evaluation Methodology, v1.0	CEM Part 2
12/99	ISO/IEC	ISO/IEC 15408, Information technology — Security techniques — Evaluation criteria for IT security, Parts 1–3 released	CC Parts 1–3
12/99 forward	CCIMB	Respond to requests for interpretations (RIs), issue final interpretations, incorporate final interpretations	—
5/00	Multiple	Common Criteria Recognition Agreement signed	CCRA
8/01	CEMEB	CEM-2001/0015, Common Methodology for Information Technology Security Evaluation, Part 2: Evaluation Methodology, Supplement: ALC_FLR — Flaw Remediation, v1.0	CEM Part 2 supplement

EXHIBIT 79.2 Summary of *Orange Book* Trusted Computer System Evaluation Criteria (TCSEC) Divisions

Evaluation Division	Evaluation Class	Degree of Trust
A — Verified protection	A1 — Verified design	Highest
B — Mandatory protection	B3 — Security domains	
	B2 — Structured protection	
	B1 — Labeled security protection	
C — Discretionary protection	C2 — Controlled access protection	
	C1 — Discretionary security protection	
D — Minimal protection	D1 — Minimal protection	Lowest

System Evaluation Criteria (TCSEC),[4] commonly known as the *Orange Book*, in 1983. A second version of this standard was issued in 1985.[5]

The *Orange Book* proposed a layered approach for rating the strength of COMPUSEC features, similar to the layered approach used by the Software Engineering Institute (SEI) Capability Maturity Model (CMM) to rate the robustness of software engineering processes. As shown in Exhibit 79.2, four evaluation divisions composed of seven classes were defined. Division A class A1 was the highest rating, while division D class D1 was the lowest. The divisions measured the extent of security protection provided, with each class and division building upon and strengthening the provisions of its predecessors. Twenty-seven specific criteria were evaluated. These criteria were grouped into four categories: security policy, accountability, assurance, and documentation. The *Orange Book* also introduced the concepts of a reference monitor, formal security policy model, trusted computing base, and assurance.

The *Orange Book* was oriented toward custom software, particularly defense and intelligence applications, operating on a mainframe computer that was the predominant technology of the time. Guidance documents were issued; however, it was difficult to interpret or apply the *Orange Book* to networks or database management systems. When distributed processing became the norm, additional standards were issued to supplement the *Orange Book,* such as the Trusted Network Interpretation and the Trusted Database Management System Interpretation. Each standard had a different color cover, and collectively they became known as the Rainbow Series. In addition, the Federal Criteria for Information Technology Security was issued by NIST and NSA in December 1992, but it was short-lived.

At the same time, similar developments were proceeding outside the United States. Between 1990 and 1993, the Commission of the European Communities, the European Computer Manufacturers Association (ECMA), the Organization for Economic Cooperation and Development (OECD), the U.K. Communications–Electronics Security Group, and the Canadian Communication Security Establishment (CSE) all issued computer security standards or technical reports. These efforts and the evolution of the Rainbow Series were driven by three main factors:[6]

1. The rapid change in technology, which led to the need to merge communications security (COMSEC) and computer security (COMPUSEC)
2. The more universal use of information technology (IT) outside the defense and intelligence communities
3. The desire to foster a cost-effective commercial approach to developing and evaluating IT security that would be applicable to multiple industrial sectors

These organizations decided to pool their resources to meet the evolving security challenge. ISO/IEC Joint Technical Committee One (JTC1) Subcommittee 27 (SC27) Working Group Three (WG3) was formed in 1990. Canada, France, Germany, the Netherlands, the United Kingdom, and the United States, which collectively became known as the CC Sponsoring Organizations, initiated the CC Project in 1993, while maintaining a close liaison with ISO/IEC JTC1 SC27 WG3. The CC Editing Board (CCEB), with the approval of ISO/IEC JTC1 SC27 WG3, released the first committee draft of the CC for public comment and review in 1996. The CC Implementation Management Board (CCIMB), again with the approval of ISO/IEC JTC1 SC27 WG3, incorporated the comments and observations gained from the first draft to create the second committee draft.

It was released for public comment and review in 1997. Following a formal comment resolution and balloting period, the CC were issued as ISO/IEC 15408 in three parts:

- ISO/IEC 15408-1(1999-12-01), Information technology — Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model
- ISO/IEC 15408-2(1999-12-01), Information technology — Security techniques — Evaluation criteria for IT security — Part 2: Security functional requirements
- ISO/IEC 15408-3(1999-12-01), Information technology — Security techniques — Evaluation criteria for IT security — Part 3: Security assurance requirements

Parallel to this effort was the development and release of the Common Evaluation Methodology, referred to as the CEM or CM, by the Common Evaluation Methodology Editing Board (CEMEB):

- CEM-97/017, Common Methodology for Information Technology Security Evaluation, Part 1: Introduction and General Model, v0.6, November 1997
- CEM-99/045, Common Methodology for Information Technology Security Evaluation, Part 2: Evaluation Methodology, v1.0, August 1999
- CEM-2001/0015, Common Methodology for Information Technology Security Evaluation, Part 2: Evaluation Methodology, Supplement: ALC_FLR — Flaw Remediation, v1.0, August 2001

As the CEM becomes more mature, it too will become an ISO/IEC standard.

Purpose and Intended Use

The goal of the CC project was to develop a standardized methodology for specifying, designing, and evaluating IT products that perform security functions which would be widely recognized and yield consistent, repeatable results. In other words, the goal was to develop a full life-cycle, consensus-based security engineering standard. Once this was achieved, it was thought, organizations could turn to commercial vendors for their security needs rather than having to rely solely on custom products that had lengthy development and evaluation cycles with unpredictable results. The quantity, quality, and cost effectiveness of commercially available IT security products would increase; and the time to evaluate them would decrease, especially given the emergence of the global economy.

There has been some confusion that the term *IT product* only refers to plug-and-play commercial off-the-shelf (COTS) products. In fact, the CC interprets the term *IT product* quite broadly, to include a single product or multiple IT products configured as an IT system or network.

The standard lists several items that are not covered and considered out of scope:[7]

- Administrative security measures and procedural controls
- Physical security
- Personnel security
- Use of evaluation results within a wider system assessment, such as certification and accreditation (C&A)
- Qualities of specific cryptographic algorithms

Administrative security measures and procedural controls generally associated with operational security (OPSEC) are not addressed by the CC/CEM. Likewise, the CC/CEM does not define how risk assessments should be conducted, even though the results of a risk assessment are required as an input to a PP.[7] Physical security is addressed in a very limited context — that of restrictions on unauthorized physical access to security equipment and prevention of and resistance to unauthorized physical modification or substitution of such equipment.[6] Personnel security issues are not covered at all; instead, they are generally handled by assumptions made in the PP. The CC/CEM does not address C&A processes or criteria. This was specifically left to each country and/or government agency to define. However, it is expected that CC/CEM evaluation results will be used as input to C&A. The robustness of cryptographic algorithms, or even which algorithms are acceptable, is not discussed in the CC/CEM. Rather, the CC/CEM limits itself to defining requirements for key management and cryptographic operation. Many issues not handled by the CC/CEM are covered by other national and international standards.

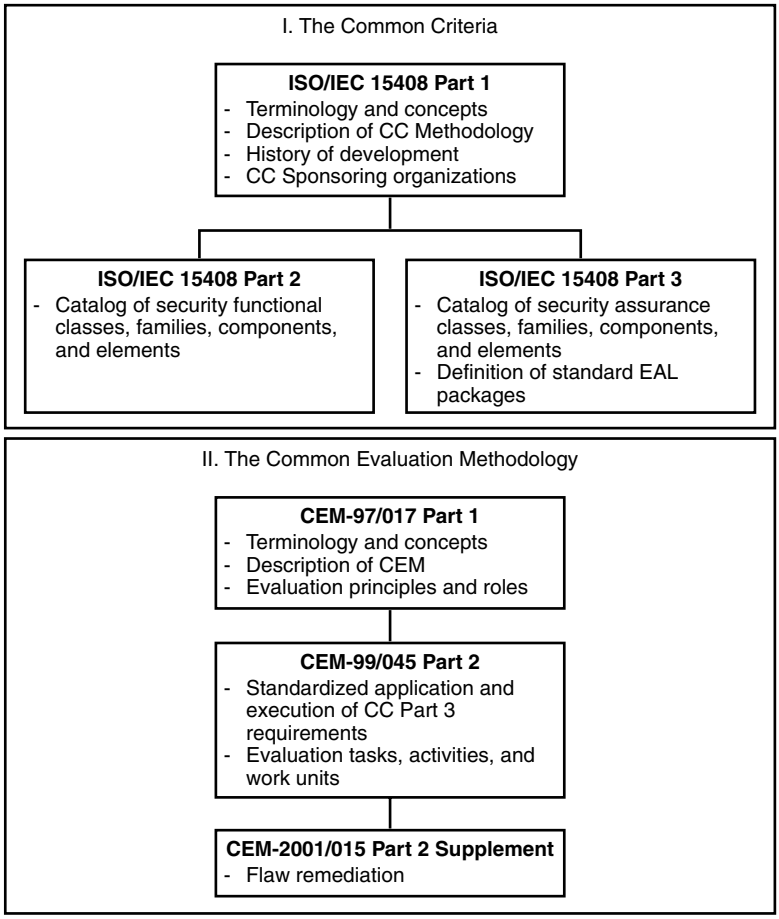


EXHIBIT 79.3 Major components of the CC CEM.

Major Components of the Methodology and How They Work

The three-part CC standard (ISO/IEC 15408) and the CEM are the two major components of the CC methodology, as shown in Exhibit 79.3.

The CC

Part 1 of ISO/IEC 15408 provides a brief history of the development of the CC and identifies the CC sponsoring organizations. Basic concepts and terminology are introduced. The CC methodology and how it corresponds to a generic system development life cycle are described. This information forms the foundation necessary for understanding and applying Parts 2 and 3. Four key concepts are presented in Part 1:

- Protection Profiles (PPs)
- Security Targets (STs)
- Targets of Evaluation (TOEs)
- Packages

A Protection Profile, or PP, is a formal document that expresses an *implementation-independent* set of security requirements, both functional and assurance, for an IT product that meets specific consumer needs.[7] The process of developing a PP helps a consumer to elucidate, define, and validate their security requirements, the end

result of which is used to (1) communicate these requirements to potential developers and (2) provide a foundation from which a security target can be developed and an evaluation conducted.

A Security Target, or ST, is an *implementation-dependent* response to a PP that is used as the basis for developing a TOE. In other words, the PP specifies security functional and assurance requirements, while an ST provides a design that incorporates security mechanisms, features, and functions to fulfill these requirements.

A Target of Evaluation, or TOE, is an IT product, system, or network and its associated administrator and user guidance documentation that is the subject of an evaluation.[7-9] A TOE is the physical implementation of an ST. There are three types of TOEs: monolithic, component, and composite. A monolithic TOE is self-contained; it has no higher or lower divisions. A component TOE is the lowest-level TOE in an IT product or system; it forms part of a composite TOE. In contrast, a composite TOE is the highest-level TOE in an IT product or system; it is composed of multiple component TOEs.

A package is a set of components that are combined together to satisfy a subset of identified security objectives.[7] Packages are used to build PPs and STs. Packages can be a collection of functional or assurance requirements. Because they are a collection of low-level requirements or a subset of the total requirements for an IT product or system, packages are intended to be reusable. Evaluation assurance levels (EALs) are examples of predefined packages.

Part 2 of ISO/IEC 15408 is a catalog of standardized security functional requirements, or SFRs. SFRs serve many purposes. They[7-9] (1) describe the security behavior expected of a TOE, (2) meet the security objectives stated in a PP or ST, (3) specify security properties that users can detect by direct interaction with the TOE or by the TOE's response to stimulus, (4) counter threats in the intended operational environment of the TOE, and (5) cover any identified organizational security policies and assumptions.

The CC organizes SFRs in a hierarchical structure of security functionality:

- Classes
- Families
- Components
- Elements

Eleven security functional classes, 67 security functional families, 138 security functional components, and 250 security functional elements are defined in Part 2. Exhibit 79.4 illustrates the relationship between classes, families, components, and elements.

A class is a grouping of security requirements that share a common focus; members of a class are referred to as families.[7] Each functional class is assigned a long name and a short three-character mnemonic beginning with an "F." The purpose of the functional class is described and a structure diagram is provided that depicts the family members. ISO/IEC 15408-2 defines 11 security functional classes. These classes are lateral toone

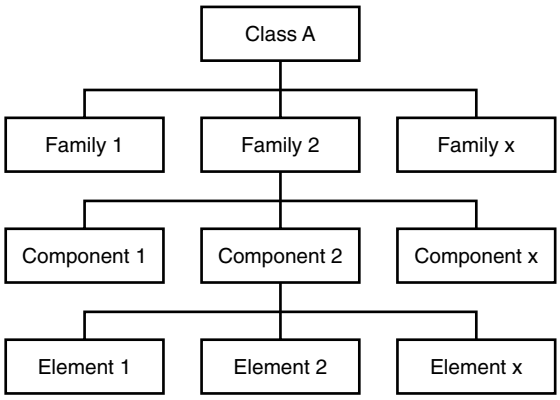


EXHIBIT 79.4 Relationship between classes, families, components, and elements.

EXHIBIT 79.5 Functional Security Classes

Short Name	Long Name	Purpose[8]
FAU	Security audit	Monitor, capture, store, analyze, and report information related to security events
FCO	Communication	Assure the identity of originators and recipients of transmitted information; non-repudiation
FCS	Cryptographic support	Management and operational use of cryptographic keys
FDP	User data protection	Protect (1) user data and the associated security attributes within a TOE and (2) data that is imported, exported, and stored
FIA	Identification and authentication	Ensure unambiguous identification of authorized users and the correct association of security attributes with users and subjects
FMT	Security management	Management of security attributes, data, and functions and definition of security roles
FPR	Privacy	Protect users against discovery and misuse of their identity
FPT	Protection of the TSF	Maintain the integrity of the TSF management functions and data
FRU	Resource utilization	Ensure availability of system resources through fault tolerance and the allocation of services by priority
FTA	TOE access	Controlling user session establishment
FTP	Trusted path/channels	Provide a trusted communication path between users and the TSF and between the TSF and other trusted IT products

another; there is no hierarchical relationship among them. Accordingly, the standard presents the classes in alphabetical order. Classes represent the broadest spectrum of potential security functions that a consumer may need in an IT product. Classes are the highest-level entity from which a consumer begins to select security functional requirements. It is not expected that a single IT product will contain SFRs from all classes. Exhibit 79.5 lists the security functional classes.

A functional family is a grouping of SFRs that share security objectives but may differ in emphasis or rigor. The members of a family are referred to as components.[7] Each functional family is assigned a long name and a three-character mnemonic that is appended to the functional class mnemonic. Family behavior is described. Hierarchics or ordering, if any, between family members is explained. Suggestions are made about potential OPSEC management activities and security events that are candidates to be audited.

Components are a specific set of security requirements that are constructed from elements; they are the smallest selectable set of elements that can be included in a Protection Profile, Security Target, or a package.[7] Components are assigned a long name and described. Hierarchical relationships between one component and another are identified. The short name for components consists of the class mnemonic, the family mnemonic, and a unique number.

An element is an indivisible security requirement that can be verified by an evaluation, and it is the lowest-level security requirement from which components are constructed.[7] One or more elements are stated verbatim for each component. Each element has a unique number that is appended to the component identifier. If a component has more than one element, all of them must be used. Dependencies between elements are listed. Elements are the building blocks from which functional security requirements are specified in a protection profile. Exhibit 79.6 illustrates the standard CC notation for security functional classes, families, components, and elements.

Part 3 of ISO/IEC 15408 is a catalog of standardized security assurance requirements or SARs. SARs define the criteria for evaluating PPs, STs, and TOEs and the security assurance responsibilities and activities of developers and evaluators. The CC organize SARs in a hierarchical structure of security assurance classes, families, components, and elements. Ten security assurance classes, 42 security assurance families, and 93 security assurance components are defined in Part 3.

A class is a grouping of security requirements that share a common focus; members of a class are referred to as families.[7] Each assurance class is assigned a long name and a short three-character mnemonic beginning

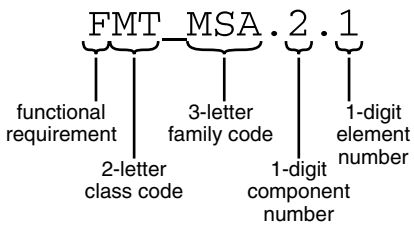


EXHIBIT 79.6 Standard notation for classes, families, components, and elements.

with an "A." The purpose of the assurance class is described and a structure diagram is provided that depicts the family members. There are three types of assurance classes: (1) those that are used for Protection Profile or Security Target validation, (2) those that are used for TOE conformance evaluation, and (3) those that are used to maintain security assurance after certification. ISO/IEC 15408-3 defines ten security assurance classes. Two classes, APE and ASE, evaluate PPs and STs, respectively. Seven classes verify that a TOE conforms to its PP and ST. One class, AMA, verifies that security assurance is maintained between certification cycles. These classes are lateral to one another; there is no hierarchical relationship among them. Accordingly, the standard presents the classes in alphabetical order. Classes represent the broadest spectrum of potential security assurance measures that a consumer may need to verify the integrity of the security functions performed by an IT product. Classes are the highest-level entity from which a consumer begins to select security assurance requirements. Exhibit 79.7 lists the security assurance classes in alphabetical order and indicates their type.

EXHIBIT 79.7 Security Assurance Classes

Short Name	Long Name	Type	Purpose
APE	Protection profile evaluation	PP/ ST	Demonstrate that the PP is complete, consistent, and technically sound
ASE	Security target evaluation	PP/ ST	Demonstrate that the ST is complete, consistent, technically sound, and suitable for use as the basis for a TOE evaluation
ACM	Configuration management	TOE	Control the process by which a TOE and its related documentation is developed, refined, and modified
ADO	Delivery and operation	TOE	Ensure correct delivery, installation, generation, and initialization of the TOE
ADV	Development	TOE	Ensure that the development process is methodical by requiring various levels of specification and design and evaluating the consistency between them
AGD	Guidance documents	TOE	Ensure that all relevant aspects of the secure operation and use of the TOE are documented in user and administrator guidance
ALC	Lifecycle support	TOE	Ensure that methodical processes are followed during the operations and maintenance phase so that security integrity is not disrupted
ATE	Tests	TOE	Ensure adequate test coverage, test depth, functional and independent testing
AVA	Vulnerability assessment	TOE	Analyze the existence of latent vulnerabilities, such as exploitable covert channels, misuse or incorrect configuration of the TOE, the ability to defeat, bypass, or compromise security credentials
AMA	Maintenance of assurance	AMA	Essure that the TOE will continue to meet its security target as changes are made to the TOE or its environment

PP/ST — Protection Profile or Security Target evaluation.
TOE — TOE conformance evaluation.
AMA — Maintenance of assurance after certification.

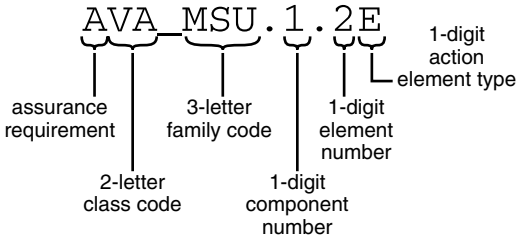


EXHIBIT 79.8 Standard notation for assurance classes, families, components, and elements.

An assurance family is a grouping of SARs that share security objectives. The members of a family are referred to as components.[7] Each assurance family is assigned a long name and a three-character mnemonic that is appended to the assurance class mnemonic. Family behavior is described. Unlike functional families, the members of an assurance family only exhibit linear hierarchical relationships, with an increasing emphasis on scope, depth, and rigor. Some families contain application notes that provide additional background information and considerations concerning the use of a family or the information it generates during evaluation activities.

Components are a specific set of security requirements that are constructed from elements; they are the smallest selectable set of elements that can be included in a Protection Profile, Security Target, or a package.[7] Components are assigned a long name and described. Hierarchical relationships between one component and another are identified. The short name for components consists of the class mnemonic, the family mnemonic, and a unique number. Again, application notes may be included to convey additional background information and considerations.

An element is an indivisible security requirement that can be verified by an evaluation, and it is the lowest-level security requirement from which components are constructed.[7] One or more elements are stated verbatim for each component. If a component has more than one element, all of them must be used. Dependencies between elements are listed. Elements are the building blocks from which a PP or ST is created. Each assurance element has a unique number that is appended to the component identifier and a one-character code. A "D" indicates assurance actions to be taken by the TOE developer. A "C" explains the content and presentation criteria for assurance evidence, that is, what must be demonstrated.[7] An "E" identifies actions to be taken or analyses to be performed by the evaluator to confirm that evidence requirements have been met. Exhibit 79.8 illustrates the standard notation for assurance classes, families, components, and elements.

Part 3 of ISO/IEC 15408 also defines seven hierarchical evaluation assurance levels, or EALs. An EAL is a grouping of assurance components that represents a point on the predefined assurance scale.[7] In short, an EAL is an assurance package. The intent is to ensure that a TOE is not over- or underprotected by balancing the level of assurance against cost, schedule, technical, and mission constraints. Each EAL has a long name and a short name, which consists of "EAL" and a number from 1 to 7. The seven EALs add new and higher assurance components as security objectives become more rigorous. Application notes discuss limitations on evaluator actions and/or the use of information generated. Exhibit 79.9 cites the seven standard EALs.

EXHIBIT 79.9 Standard EAL Packages

Short Name	Long Name	Level of Confidence
EAL 1	Functionally tested	Lowest
EAL 2	Structurally tested	
EAL 3	Methodically tested and checked	
EAL 4	Methodically designed, tested, and reviewed	Medium
EAL 5	Semi-formally designed and tested	
EAL 6	Semi-formally verified design and tested	
EAL 7	Formally verified design and tested	Highest

The CEM

The Common Methodology for Information Technology Security Evaluation, known as the CEM (or CM), was created to provide concrete guidance to evaluators on how to apply and interpret SARs and their developer, content and presentation, and evaluator actions, so that evaluations are consistent and repeatable. To date the CEM consists of two parts and a supplement. Part 1 of the CEM defines the underlying principles of evaluations and delineates the roles of sponsors, developers, evaluators, and national evaluation authorities. Part 2 of the CEM specifies the evaluation methodology in terms of evaluator tasks, subtasks, activities, subactivities, actions, and work units, all of which tie back to the assurance classes. A supplement was issued to Part 2 in 2001 that provides evaluation guidance for the ALC_FLR family. Like the CC, the CEM will become an ISO/IEC standard in the near future.

CC User Community and Stakeholders

The CC user community and stakeholders can be viewed from two different constructs: (1) generic groups of users, and (2) formal organizational entities that are responsible for overseeing and implementing the CC/CEM worldwide. (See Exhibit 79.10.)

ISO/IEC 15408-1 defines the CC/CEM generic user community to consist of:

- Consumers
- Developers
- Evaluators

Consumers are those organizations and individuals who are interested in acquiring a security solution that meets their specific needs. Consumers state their security functional and assurance requirements in a PP. This mechanism is used to communicate with potential developers by conveying requirements in an implementation-independent manner and information about how a product will be evaluated.

Developers are organizations and individuals who design, build, and sell IT security products. Developers respond to a consumer's PP with an implementation-dependent detailed design in the form of an ST. In addition, developers prove through the ST that all requirements from the PP have been satisfied, including the specific activities levied on developers by SARs.

Evaluators perform independent evaluations of PPs, STs, and TOEs using the CC/CEM, specifically the evaluator activities stated in SARs. The results are formally documented and distributed to the appropriate entities. Consequently, consumers do not have to rely only on a developer's claims — they are privy to independent assessments from which they can evaluate and compare IT security products. As the standard[7] states:

> The CC is written to ensure that evaluations fulfill the needs of consumers — this is the fundamental purpose and justification for the evaluation process.

The Common Criteria Recognition Agreement (CCRA),[10] signed by 15 countries to date, formally assigns roles and responsibilities to specific organizations:

- Customers or end users
- IT product vendors
- Sponsors
- Common Criteria Testing Laboratories (CCTLs)
- National Evaluation Authorities
- Common Criteria Implementation Management Board (CCIMB)

Customers or end users perform the same role as consumers in the generic model. They specify their security functional and assurance requirements in a PP. By defining an assurance package, they inform developers how the IT product will be evaluated. Finally, they use PP, ST, and TOE evaluation results to compare IT products and determine which best meets their specific needs and will work best in their particular operational environment.

IT product vendors perform the same role as developers in the generic model. They respond to customer requirements by developing an ST and corresponding TOE. In addition, they provide proof that all security

EXHIBIT 79.10 Roles and Responsibilities of CC/CEM Stakeholders

Category	Roles and Responsibilities
I. Generic Users[a]	
Consumers	Specify requirements Inform developers how IT product will be evaluated Use PP, ST, and TOE evaluation results to compare products
Developers	Respond to consumer's requirements Prove that all requirements have been met
Evaluators	Conduct independent evaluations using standardized criteria
II. Specific Organizations[b]	
Customer or end user	Specify requirements Inform vendors how IT product will be evaluated Use PP, ST, and TOE evaluation results to compare IT products
IT product vendor	Respond to customer's requirements Prove that all requirements have been met Deliver evidence to sponsor
Sponsor	Contract with CCTL for IT product to be evaluated Deliver evidence to CCTL
Common Criteria Testing Laboratory (CCTL)	Request accreditation from National Evaluation Authority Receive evidence from sponsor Conduct evaluations according to CC/CEM Produce Evaluation Technical Reports Make certification recommendation to National Evaluation Authority
National Evaluation Authority	Define and manage national evaluation scheme Accredit CCTLs Monitor CCTL evaluations Issue guidance to CCTLs Issue and recognize CC certificates Maintain Evaluated Products Lists and PP Registry
Common Criteria Implementation Management Board (CCIMB)	Facilitate consistent interpretation and application of the CC/CEM Oversee National Evaluation Authorities Render decisions in response to Requests for Interpretations (RIs) Maintain the CC/CEM Coordinate with ISO/IEC JTC1 SC27 WG3 and CEMEB

[a] ISO/IEC 15408-1(1999-12-01), Information technology — Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model; Part 2: Security functional requirements; Part 3: Security assurance requirements.

[b] Arrangement on the Recognition of Common Criteria Certificates in the Field of Information Technology Security, May 23, 2000.

functional and assurance requirements specified in the PP have been satisfied by their ST and TOE. This proof and related development documentation is delivered to the Sponsor.

A new role introduced by the CCRA is that of the Sponsor. A Sponsor locates an appropriate CCTL and makes contractual arrangements with them to conduct an evaluation of an IT product. They are responsible for delivering the PP, ST, or TOE and related documentation to the CCTL and coordinating any pre-evaluation activities. A Sponsor may represent the customer or the IT product vendor, or be a neutral third party such as a system integrator.

The CCRA divides the generic evaluator role into three hierarchical functions: Common Criteria Testing Laboratories (CCTLs), National Evaluation Authorities, and the Common Criteria Implementation Management Board (CCIMB).

CCTLs must meet accreditation standards and are subject to regular audit and oversight activities to ensure that their evaluations conform to the CC/CEM. CCTLs receive the PP, ST, or TOE and the associated documentation from the Sponsor. They conduct a formal evaluation of the PP, ST or TOE according to the CC/CEM and the assurance package specified in the PP. If missing, ambiguous, or incorrect information is uncovered during

the course of an evaluation, the CCTL issues an Observation Report (OR) to the sponsor requesting clarification. The results are documented in an Evaluation Technical Report (ETR), which is sent to the National Evaluation Authority along with a recommendation that the IT product be certified (or not).

Each country that is a signatory to the CCRA has a National Evaluation Authority. The National Evaluation Authority is the focal point for CC activities within its jurisdiction. A National Evaluation Authority may take one of two forms — that of a Certificate Consuming Participant or that of a Certificate Authorizing Participant. A Certificate Consuming Participant recognizes CC certificates issued by other entities but, at present, does not issue any certificates itself. It is not uncommon for a country to sign on to the CCRA as a Certificate Consuming Participant, then switch to a Certificate Authorizing Participant later, after it has established a national evaluation scheme and accredited some CCTLs.

A Certificate Authorizing Participant is responsible for defining and managing the evaluation scheme within its jurisdiction. This is the administrative and regulatory framework by which CCTLs are initially accredited and subsequently maintain their accreditation. The National Evaluation Authority issues guidance to CCTLs about standard practices and procedures and monitors evaluation results to ensure their objectivity, repeatability, and conformance to the CC/CEM. The National Evaluation Authority issues official CC certificates, if they agree with the CCTL recommendation, and recognizes CC certificates issued by other National Evaluation Authorities. In addition, the National Evaluation Authority maintains the Evaluated Products List and PP Registry for its jurisdiction.

The Common Criteria Implementation Management Board (CCIMB) is composed of representatives from each country that is a party to the CCRA. The CCIMB has the ultimate responsibility for facilitating the consistent interpretation and application of the CC/CEM across all CCTLs and National Evaluation Authorities. Accordingly, the CCIMB monitors and oversees the National Evaluation Authorities. The CCIMB renders decisions in response to Requests for Interpretations (RIs). Finally, the CCIMB maintains the current version of the CC/CEM and coordinates with ISO/IEC JTC1 SC27 WG3 and the CEMEB concerning new releases of the CC/CEM and related standards.

Future of the CC

As mentioned earlier, the CC/CEM is the result of a 30-year evolutionary process. The CC/CEM and the processes governing it have been designed so that CC/CEM will continue to evolve and not become obsolete when technology changes, like the *Orange Book* did. Given that and the fact that 15 countries have signed the CC Recognition Agreement (CCRA), the CC/CEM will be with us for the long term. Two near-term events to watch for are the issuance of both the CEM and the SSE-CMM as ISO/IEC standards.

The CCIMB has set in place a process to ensure consistent interpretations of the CC/CEM and to capture any needed corrections or enhancements to the methodology. Both situations are dealt with through what is known as the Request for Interpretation (RI) process. The first step in this process is for a developer, sponsor, or CCTL to formulate a question. This question or RI may be triggered by four different scenarios. The organization submitting the RI:[10]

1. Perceives an error in the CC or CEM
2. Perceives the need for additional material in the CC or CEM
3. Proposes a new application of the CC or CEM and wants this new approach to be validated
4. Requests help in understanding part of the CC or CEM

The RI cites the relevant CC or CEM reference and states the problem or question.

The ISO/IEC has a five-year reaffirm, update, or withdrawal cycle for standards. This means that the next version of ISO/IEC 15408, which will include all of the final interpretations in effect at that time, should be released near the end of 2004. The CCIMB has indicated that it may issue an interim version of the CC or CEM, prior to the release of the new ISO/IEC 15408 version, if the volume and magnitude of final interpretations warrant such an action. However, the CCIMB makes it clear that it remains dedicated to support the ISO/IEC process.[1]

Acronyms

ADP — Automatic Data Processing equipment
C&A — Certification and Accreditation

CC — Common Criteria
CCEB — Common Criteria Editing Board
CCIMB — Common Criteria Implementation Board
CCRA — Common Criteria Recognition Agreement
CCTL — accredited CC Testing Laboratory
CEM — Common Evaluation Methodology
CESG — U.K. Communication Electronics Security Group
CMM — Capability Maturity Model
COMSEC — Communications Security
COMPUSEC — Computer Security
CSE — Canadian Computer Security Establishment
DoD — U.S. Department of Defense
EAL — Evaluation Assurance Level
ECMA — European Computer Manufacturers Association
ETR — Evaluation Technical Report
IEC — International Electrotechnical Commission
ISO — International Organization for Standardization
JTC — ISO/IEC Joint Technical Committee
NASA — U.S. National Aeronautics and Space Administration
NIST — U.S. National Institute of Standards and Technology
NSA — U.S. National Security Agency
OECD — Organization for Economic Cooperation and Development
OPSEC — Operational Security
OR — Observation Report
PP — Protection Profile
RI — Request for Interpretation
SAR — Security Assurance Requirement
SEI — Software Engineering Institute at Carnegie Mellon University
SFR — Security Functional Requirement
SSE-CMM — System Security Engineering CMM
ST — Security Target
TCSEC — Trusted Computer Security Evaluation Criteria
TOE — Target of Evaluation

References

1. www.commoncriteria.org; centralized resource for current information about the Common Criteria standards, members, and events.
2. DoD 5200.28M, *ADP Computer Security Manual — Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource-Sharing ADP Systems*, U.S. Department of Defense, January 1973.
3. DoD 5200.28M, *ADP Computer Security Manual — Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource-Sharing ADP Systems*, with 1st Amendment, U.S. Department of Defense, June 25, 1979.
4. CSC-STD-001-83, Trusted Computer System Evaluation Criteria (TCSEC), National Computer Security Center, U.S. Department of Defense, August 15, 1983.
5. DoD 5200.28-STD, Trusted Computer System Evaluation Criteria (TCSEC), National Computer Security Center, U.S. Department of Defense, December 1985.
6. Herrmann, D., *A Practical Guide to Security Engineering and Information Assurance*, Auerbach Publications, Boca Raton, FL, 2001.
7. ISO/IEC 15408-1(1999-12-01), Information technology — Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model.

8. ISO/IEC 15408-2(1999-12-01), Information technology — Security techniques — Evaluation criteria for IT security — Part 2: Security functional requirements.
9. ISO/IEC 15408-3(1999-12-01), Information technology — Security techniques — Evaluation criteria for IT security — Part 3: Security assurance requirements.
10. Arrangement on the Recognition of Common Criteria Certificates in the Field of Information Technology Security, May 23, 2000.

80

A Look at the Common Criteria

Ben Rothke, CISSP

Until recently, information security was something that only the military and some financial services took seriously. But in the post-September 11 era, all of that has radically changed. As this chapter is being written, American troops are in Iraq, and with that, information security has become even more critical.

While a major story was Microsoft's Trustworthy Computing Initiative of 2002, much of the momentum for information security started years earlier. And one of the prime forces has been the Common Criteria.

The need for a common information security standard is obvious. Security means many different things to different people and organizations. But this subjective level of security cannot be objectively evaluated. So, a common criterion was needed to evaluate the security of an information technology product.

The need for common agreement is clear. When you buy a DVD, put gas in your car, or make an online purchase from an E-commerce site, all of these function due to the simple fact that they operate in agreement with a common set of standards and guidelines.

And that is precisely what the Common Criteria is meant to be, a global security standard. This ensures that there is a common mechanism for evaluating the security of technology products and systems. By providing a common set of requirements for comparing the security functions of software and hardware products, the Common Criteria enables users to have an objective yardstick in which to evaluate the security of the respective product.

With that, Common Criteria certification is slowly but increasingly being used as a criterion for many Requests for Proposals, primarily in the government sector. By offering a consistent, rigorous, and independently verifiable set of evaluation requirements to hardware and software, the Common Criteria is attempting to be the Good Housekeeping™ seal of approval for the information security sector.

But what is especially historic about the Common Criteria is that it is the first time governments around the world have united in support of an information security evaluation program.

Origins of the Common Criteria

In the United States, the Common Criteria has its roots in the Trusted Computer System Evaluation Criteria (TCSEC). The most notable aspect of the TCSEC was the *Orange Book*. But by the early 1990s, it was clear that the TCSEC was not viable for the new world of client/server computing. The main problem with the TCSEC was that it was not accommodating to new computing paradigms.

In Europe, the Information Technology Security Evaluation Criteria (ITSEC), already in development in the early 1990s, was published in 1991 by the European Commission. This was a joint effort with representatives from France, Germany, the Netherlands, and the United Kingdom contributing.

Simultaneously, the Canadian government created the Canadian Trusted Computer Product Evaluation Criteria as an amalgamation of the ITSEC and TCSEC approaches. In the United States, the draft Federal Criteria for Information Technology Security was published in 1993 in an attempt to combine the various methods for evaluation criteria.

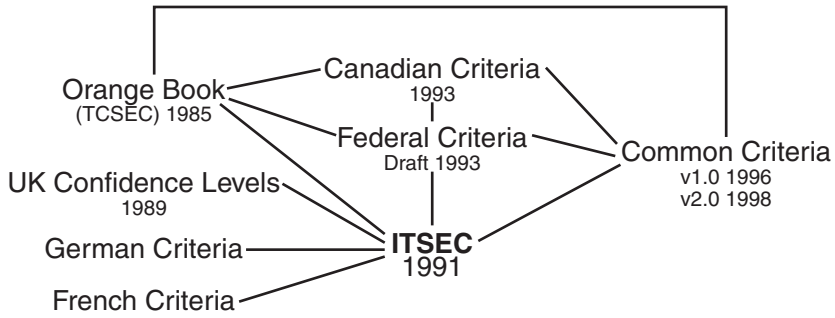


EXHIBIT 80.1 The Common Criteria.

With so many different approaches going on at once, there was consensus to create a common approach. At that point, the International Organization for Standardization (ISO) began to develop a new set of standard evaluation criteria for general use that could be used internationally. The new methodology is what later became the Common Criteria.

The goal was to unite the various international and diverse standards into a new set of criteria for the evaluation of information technology products. This effort ultimately led to the development of the Common Criteria, which is now an international standard in ISO 15408:1999.[1]

Exhibit 80.1 illustrates the development of the Common Criteria.

The specific international organizations that are representatives to the Common Criteria include:

- NIST (United States)
- NSA (United States)
- SCSSI (France)
- NLNCSA (the Netherlands)
- CSE (Canada)
- CESG (United Kingdom)

The international recognition of the Common Criteria comes via the signing of a Mutual Recognition Arrangement (MRA) between the various countries. The MRA enables products that have earned Common Criteria certification to be used in different jurisdictions without the need for them to be reevaluated and recertified each time. The recognition of the results of the single evaluations means that products evaluated in one MRA member nation can be accepted in the other member nations.

Common Criteria Sections

Common Criteria version 2.1 is the current version[2] of the Common Criteria. Version 2.1 is a set of three distinct but related parts that are individual documents. The three parts of the Common Criteria are:

- Part 1 (61 pages) is the introduction to the Common Criteria. It defines the general concepts and principles of information technology security evaluation and presents a general model of evaluation. Part 1 also presents the constructs for expressing information technology security objectives, for selecting and defining information technology security requirements, and for writing high-level specifications for products and systems. In addition, the usefulness of each part of the Common Criteria is described in terms of each of the target audiences.
- Part 2 (362 pages) details the specific security functional requirements and details a criterion for expressing the security functional requirements for Targets of Evaluation (TOEs).
- Part 3 (216 pages) details the security assurance requirements and defines a set of assurance components as a standard way of expressing the assurance requirements for TOEs. Part 3 lists the set of assurance components, families, and classes, defines evaluation criteria for Protection Profiles[3] (PPs[4]) and Security Targets (STs[5]), and presents evaluation assurance levels that define the predefined Common Criteria scale for rating assurance for TOEs, namely the Evaluation Assurance Levels (EAL).

Protection Profiles and Security Targets

Protection Profiles (PPs) and Security Targets (STs) are two building blocks of the Common Criteria.

A PP defines a standard set of security requirements for a specific type of product (e.g., operating systems, databases, firewalls, etc.). These profiles form the basis of the Common Criteria evaluation. By listing required security features for product families, the Common Criteria allows products to state conformity to a relevant protection profile. During Common Criteria evaluation, the product is tested against a specific PP, providing reliable verification of the security capabilities of the product.

The overall purpose of Common Criteria product certification is to provide end users with a significant level of trust. Before a product can be submitted for certification, the vendor must first specify an ST. The ST description includes an overview of the product, potential security threats, detailed information on the implementation of all security features included in the product, and any claims of conformity against a PP at a specified EAL (Evaluation Assurance Level).

The vendor must submit the ST to an accredited testing laboratory for evaluation. The laboratory then tests the product to verify the described security features and evaluate the product against the claimed PP. The end result of a successful evaluation includes official certification of the product against a specific PP at a specified EAL. Exhibit 80.2 shows the required contents of a PP.

Examples of various protection profiles can be found at:

- NSA PP for firewalls and a peripheral sharing switch: www.radium.ncsc.mil/tpep/library/protection_profiles/index.html
- IATF PP for firewalls, VPN, peripheral sharing switch, remote access, multiple domain solutions, mobile code, operating systems, tokens, secured messaging, PKI and KMI, and IDS: www.nsff.org/protection_profiles/profiles.cfm
- NIST PP for smart cards, an operating system, role-based access control, and firewalls: http://niap.nist.gov/cc-scheme/PPRegistry.html

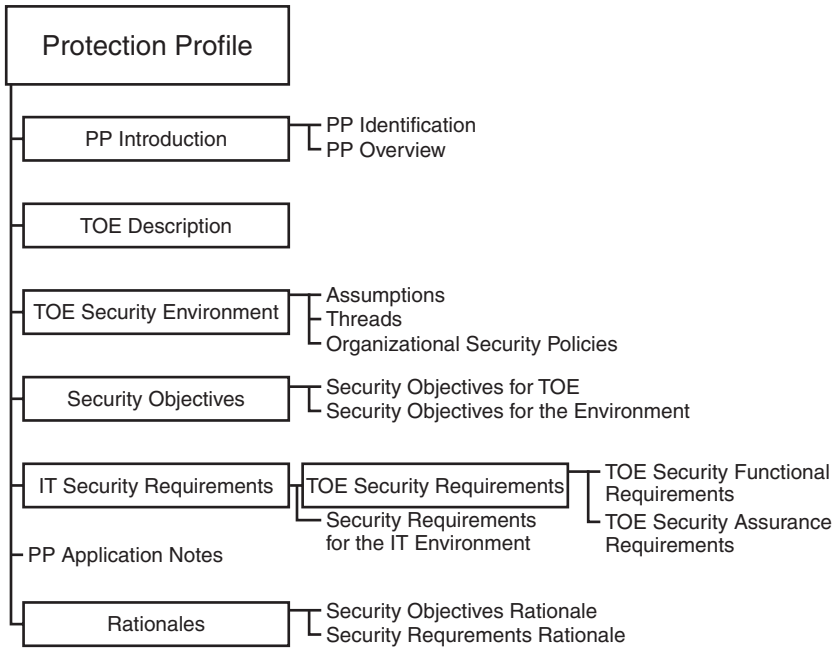


EXHIBIT 80.2 Protection Profile.

Security Requirements

Security guru Bruce Schneier has made a mantra out of his proclamation that "security is a process, not a product." With that in mind, the Common Criteria defines a number of security processes and functional requirements. These are the highest-level categories and are known as *classes* in Common Criteria vernacular. There are 11 Common Criteria classes, namely:

1. Audit
2. Cryptographic Support
3. Communications
4. User Data Protection
5. Identification and Authentication
6. Security Management
7. Privacy
8. Protection of the TOE Security Functions
9. Resource Utilization
10. TOE Access
11. Trusted Path/Channels

Each of these classes contains a subset number of families. The requirements within each family share a common security objective, but often fluctuate to the specific level of risk.

Common Criteria Security Assurance Classes

Part 3 of the Common Criteria lists eight assurance classes, namely:

1. Configuration Management
2. Delivery and Operation
3. Development
4. Guidance Documents
5. Life Cycle Support
6. Tests
7. Vulnerability Assessment
8. Assurance Maintenance

Also, the Common Criteria has seven assurance rankings, called Evaluation Assurance Levels (EALs); namely:

1. EAL1: functionally tested
2. EAL2: structurally tested
3. EAL3: methodically tested and checked
4. EAL4: methodically designed, tested, and reviewed
5. EAL5: semiformally designed and tested
6. EAL6: semiformally verified design and tested
7. EAL7: formally verified design and tested

EAL1 is the lowest ranking. Products certified to EAL4 and above can only achieve certification if they were originally designed with a very strong level of security engineering. EAL7, the highest level, offers extremely high assurances of security, but is often far too expensive to develop for general consumer use.

Many people are familiar with the TCSEC levels, which made C2 quite famous. Exhibit 80.3 compares the Common Criteria to TCSEC levels.

Evaluation Assurance Levels

The specifics of each evaluation assurance level are as follows.[6]

EXHIBIT 80.3 Common Criteria Compared to TCSEC Levels

Common Criteria	U.S. TCSEC
N/A	D: Minimal Protection
EAL 1	
EAL 2	C1: Discretionary Security
EAL 3	C2: Controlled Access
EAL 4	B1: Labeled Security
EAL 5	B2: Structured Protection
EAL 6	B3: Security Domains
EAL 7	A1: Verified Design

EAL1: Functionally Tested

EAL1 is applicable where there is some level of confidence in the correct level of operation required, but the threats to security are not viewed as serious. It will be of value where independent assurance is required to support contention that due care has been exercised with respect to the protection of personal or similar information.

This level provides an evaluation of the TOE as made available to the consumer, including independent testing against a specification, and an examination of the guidance documentation is provided. For the most part, almost any product can gain EAL1, which makes this level insignificant for any type of effective information security assistance.

Once again, EAL1 should be viewed as the most basic level of security. For those organizations that require more significant levels of assurance, EAL1 would clearly not be appropriate.

EAL2: Structurally Tested

EAL2 requires greater assistance with the applications developer in terms of the delivery of design information and test results, but should not demand more effort on the part of the developer than what best practices would dictate.

EAL2 is applicable in those circumstances where developers or users require a low to moderate level of independently assured security in the absence of ready availability of the complete development record. Such a situation may arise when securing legacy systems, or where access to the developer may be limited.

EAL3: Methodically Tested and Checked

EAL3 permits a developer to gain maximum assurance from positive security engineering at the design stage without substantial alteration of existing best development practices. It is applicable in those circumstances where developers or users require a moderate level of independently assured security, and require a thorough investigation of the TOE and its development without incurring substantial reengineering costs.

An EAL3 evaluation provides an analysis supported by *gray-box testing* (see Exhibit 80.4), selective confirmation of the developer test results, and evidence of a developer search for obvious vulnerabilities. Development of environmental controls and TOE configuration management are also required.

EAL4: Methodically Designed, Tested, and Reviewed

EAL4 permits a developer to maximize assurance gained from positive security engineering based on good commercial development practices. Although rigorous, these practices do not require substantial specialist knowledge, skills, or other resources. EAL4 is the highest level at which it is likely to be economically feasible to retrofit an existing product line. It is applicable in those circumstances where developers or users require a moderate-to-high level of independently assured security in conventional commodity TOEs, and are prepared to incur additional security-specific engineering costs.

An EAL4 evaluation provides an analysis supported by the low-level design of the modules of the TOE and a subset of the implementation. Testing is supported by an independent search for vulnerabilities. Development

EXHIBIT 80.4 Of White-Box, Black-Box, and Gray-Box Testing

A large part of the Common Criteria evaluation includes the TOE testing. There are different methods of testing a piece of hardware or software: white-box, black-box, and gray-box testing.

White-Box Testing

White-box testing is also known as open-box testing. This is a software testing technique in which the tester has explicit knowledge of the internal workings of the item being tested. In addition, the white-box tester is able to select the test data. A caveat of white-box testing is that the testing can only be meaningful if the person carrying out the testing knows what the software or hardware is supposed to do. This is often much more difficult than it sounds. In addition, actual review of the code is performed.

Black-Box Testing

Black-box testing is a technique in which the tester does not know the internal workings of the item being tested. In a black-box test, the tester only knows the inputs and what the expected outcomes should be but not how the program will arrive at those outputs. In black-box testing, the tester does not examine the software code itself.

Black-box testing advantages include (from www.webopedia.com/TERM/B/Black_Box_Testing.html):

- Unbiased because the designer and the tester are independent of each other
- Tester does not need knowledge of any specific programming languages
- Test is done from the point of view of the user, not the designer
- Test cases can be designed as soon as the specifications are complete

Black-box testing disadvantages include:

- Test can be redundant if the software designer has already run a test case.
- Test cases are difficult to design.
- Testing every possible input stream is unrealistic because it would take an inordinate amount of time; therefore, many program paths will go untested.

Gray-Box Testing

For a complete software examination, both white-box and black-box tests are required. With that, a combination of different methods — so that they are not hindered by the limitations of a particular one — is used. This is called gray-box testing

controls are supported by a life-cycle model, identification of tools, and automated configuration management. EAL4 is becoming a popular evaluation target, akin to what TCSEC C2 was.[7]

EAL5: Semiformally Designed and Tested

EAL5 is where things get interesting and the real security efficacy of the Common Criteria can be seen. EAL5 permits a developer to gain maximum assurance from security engineering, based upon rigorous commercial development practices, supported by moderate application of specialist security engineering techniques. Such a TOE will probably be designed and developed with the intent of achieving EAL5 assurance. It is likely that the additional costs attributable to the EAL5 requirements, relative to rigorous development without the application of specialized techniques, will not be large.

EAL5 is therefore applicable in those circumstances where developers or users require a high level of independently assured security in a planned development and require a rigorous development approach without incurring unreasonable costs attributable to specialist security techniques.

An EAL5 evaluation provides an analysis that includes all of the implementation. Assurance is supplemented by a formal model and a semiformal presentation of the functional specification and high-level design, and a semiformal demonstration of correspondence. The search for vulnerabilities must ensure resistance to attackers with a moderate attack potential. Covert channel analysis and design are also required. As can be seen, an EAL5 evaluation can become quite costly.

EAL6: Semiformally Verified Design and Tested

EAL6 permits developers to gain high assurance from the application of security engineering techniques to a rigorous development environment in order to produce a premium TOE for protecting high-value assets against significant risks.

EAL6 is, therefore, applicable to the development of security TOE for application in high-risk situations where the value of the protected assets justifies the additional cost.

An EAL6 evaluation provides an analysis that is supported by a modular and layered approach to design, and a structured presentation of the implementation. The independent search for vulnerabilities must ensure resistance to attackers with a high attack potential. The search for covert channels must be systematic. Development environment and configuration management controls are further strengthened.

EAL7: Formally Verified Design and Tested

EAL7 is applicable to the development of security TOE for application in extremely high-risk situations and/or where the high value of the assets justifies the higher costs. Practical application of EAL7 is currently limited to TOEs with tightly focused security functionality that is amenable to extensive formal analysis.

For an EAL7 evaluation, the formal model is supplemented by a formal presentation of the functional specification and high-level design, showing correspondence. Evidence of developer "white-box" testing (see Exhibit 80.4) and complete, independent confirmation of the developer test results is required. Complexity of the design must be minimized.

A list of certified products is available at www.commoncriteria.org/epl. Of the 85 products listed,[8] only one is at EAL5 and the remainder is certified to EAL4 and below.

The actual evaluation for Common Criteria certification is not done by any governing body, but rather by independent evaluation laboratories. The official list of Common Criteria evaluation laboratories is found at www.commoncriteria.org/services/LabCountry.htm. In the United States, there are just seven Common Criteria evaluation laboratories.

Commercial laboratories can evaluate only EAL1 through EAL 4; EAL5 through EAL7 must be done by official bodies. In the United States, the National Security Agency (NSA) performs these tests.

Government and Commercial Use of Common Criteria

The U.S. Department of Defense directive NSTISSP #11 (National Security Telecommunications and Information Systems Security Policy), which became effective in July 2002, requires any product acquired for national security systems to achieve EAL3 certification for non-cryptographic module products. This includes all commercial-off-the-shelf (COTS) or government-off-the-shelf (GOTS) information assurance (IA) or IA-enabled information technology products that are to be used as part of a solution for systems entering, processing, storing, displaying, or transmitting national security information.

Within the commercial sector, Microsoft has used the Common Criteria as a selling point for its operating systems. In October 2002, Windows 2000 received Common Criteria EAL4 certification.[9] The actual certification (or, in Common Criteria vernacular, conformance claim) was EAL 4 Augmented (Flaw Remediation[10]) and was for Windows 2000 Professional, Server, and Advanced Server with Service Pack 3 and hotfix Q326886. A dissenting look at the aspect of certifying Windows is detailed in *Understanding the Windows EAL4 Evaluation.*[11]

Sun Microsystems has also entered the Common Criteria arena. In fact, Trusted Solaris 8 received its EAL4 conformance claim before that of Windows 2000. The only difference between the two was that Windows 2000 was performed by a U.S.-based testing laboratory (SAIC), while Solaris testing was done by CESG[12] in the United Kingdom.

Problems with the Common Criteria

While there are huge benefits to the Common Criteria, there are also problems. The point of this chapter is not to detail those problems, but in a nutshell, some of the main issues are:

- *Administrative.* The overhead involved with gaining certification takes a huge amount of time and resources.
- *Expensive.* Gaining certification is extremely expensive. Getting quotes from Common Criteria Testing Laboratories is understandably infeasible, given the many variables involved. It is estimated that Microsoft spent millions of dollars in getting Windows 2000 certified.
- *Labor-intensive.* The certification process takes many, many weeks and months.
- *Requires skilled and experienced analysts.* The number of information security professionals with the required experience is still lacking.
- *Various interpretations.* The Common Criteria leaves room for various interpretations of what it is attempting to achieve.
- Limited number of Common Criteria Testing Laboratories. There are only seven laboratories in the United States.
- Becoming a Common Criteria Testing Laboratory takes a long time. Even for those organizations that are interested in becoming certified, that process in and of itself takes quite a while.

Conclusion

The Common Criteria is indeed historic in that it is the first time governments around the world have united in support of an information security evaluation program. Yet while they may be in agreement about the need for an information security evaluation program, industry as a whole has not jumped on the Common Criteria bandwagon, especially in the United States.

In fact, many have questioned the efficacy of the Common Criteria, especially after Windows 2000 still continues to be plagued by security holes.

Nonetheless, the Common Criteria should be seen as the beginning of an effective and comprehensive information security evaluation program — not as the ultimate example of one.

Notes

1. The official name of the standard is the International Common Criteria for Information Technology Security Evaluation.
2. As of May 2003.
3. www.commoncriteria.org/protection_profiles.
4. A protection profile is a set of security requirements for a category of TOE.
5. Security targets are the set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.
6. From www.commoncriteria.org/docs/EALs.html.
7. See "The Case against C2," *Windows NT Magazine,* May 1997.
8. As of May 2003.
9. http://niap.nist.gov/cc-scheme/CCEVS_VID402-VR.pdf.
10. To meet the Flaw Remediation requirement over and above EAL 4, as Windows 2000 did, the developer/vendor must establish flaw remediation procedures that describe the tracking of security flaws, the identification of corrective actions, and the distribution of corrective action information to customers. The Microsoft Security Response Center fulfills these roles for Windows 2000. See www.microsoft.com/technet/security/issues/W2KCCWP.asp.
11. http://eros.cs.jhu.edu/~shap/NT-EAL4.html.
12. CESG is the U.K. Government's National Technical Authority for Information Assurance.

Links

1. National Information Assurance Partnership (NIAP) home page: http://niap.nist.gov
2. NIAP Common Criteria Scheme home page: http://niap.nist.gov/cc-scheme
3. International Common Criteria information portal: www.commoncriteria.org
4. Common Criteria Overview: www.commoncriteria.org/introductory_overviews/CCIntroduction.pdf

5. Canadian Common Criteria Evaluation and Certification Scheme: www.cse-cst.gc.ca/en/services/common_criteria/common_criteria.html
6. British Common Criteria Evaluation and Certification Scheme: www.cesg.gov.uk/site/iacs/index.cfm?menuSelected = 1&displayPage = 1
7. International Common Criteria Conference: www.iccconference.com
8. Automating the Common Criteria Evaluation Process: www.cisc.jmu.edu/research/prietodiaz2.html
9. Exploring Visual Impact Analysis Approaches for Common Criteria Security Evaluations: www.cisc.jmu.edu/news_events/presentations/Bohner/Bohner1.pdf
10. Common Criteria Tools: http://niap.nist.gov/tools/cctool.html

81

The Security Policy Life Cycle: Functions and Responsibilities

Patrick D. Howard, CISSP

Most information security practitioners normally think of security policy development in fairly narrow terms. Use of the term *policy development* usually connotes writing a policy on a particular topic and putting it into effect. If practitioners happen to have recent, hands-on experience in developing information security policies, they may also include in their working definition the staffing and coordination of the policy, security awareness tasks, and perhaps policy compliance oversight. But is this an adequate inventory of the functions that must be performed in the development of an effective security policy? Unfortunately, many security policies are ineffective because of a failure to acknowledge all that is actually required in developing policies. Limiting the way security policy development is defined also limits the effectiveness of policies resulting from this flawed definition.

Security policy development goes beyond simple policy writing and implementation. It is also much more than activities related to staffing a newly created policy, making employees aware of it, and ensuring that they comply with its provisions. A security policy has an entire life cycle that it must pass through during its useful lifetime. This life cycle includes research, getting policies down in writing, getting management buy-in, getting them approved, getting them disseminated across the enterprise, keeping users aware of them, getting them enforced, tracking them and ensuring that they are kept current, getting rid of old policies, and other similar tasks. Unless an organization recognizes the various functions involved in the policy development task, it runs the risk of developing policies that are poorly thought out, incomplete, redundant, not fully supported by users or management, superfluous, or irrelevant.

Use of the *security policy life cycle* approach to policy development can ensure that the process is comprehensive of all functions necessary for effective policies. It leads to a greater understanding of the policy development process through the definition of discrete roles and responsibilities, through enhanced visibility of the steps necessary in developing effective policies, and through the integration of disparate tasks into a cohesive process that aims to generate, implement, and maintain policies.

Policy Definitions

It is important to be clear on terms at the beginning. What do we mean when we say *policy*, or *standard*, or *baseline*, or *guideline*, or *procedure*? These are terms information security practitioners hear and use every day in the performance of their security duties. Sometimes they are used correctly, and sometimes they are not. For the purpose of this discussion, these terms are defined in Exhibit 81.1.

Exhibit 81.1 provides generally accepted definitions for a security policy hierarchy. A *policy* is defined as a broad statement of principle that presents management's position for a defined control area. A *standard* is defined as a rule that specifies a particular course of action or response to a given situation and is a mandatory directive for carrying out policies. *Baselines* establish how security controls are to be implemented on specific

EXHIBIT 81.1 Definition of Terms

Policy: A broad statement of principle that presents management's position for a defined control area. Policies are intended to be long-term and guide the development of more specific rules to address specific situations. Policies are interpreted and supported by standards, baselines, procedures, and guidelines. Policies should be relatively few in number, should be approved and supported by executive management, and should provide overall direction to the organization. Policies are mandatory in nature, and an inability to comply with a policy should require approval of an exception.

Standard: A rule that specifies a particular course of action or response to a given situation. Standards are mandatory directives to carry out management's policies and are used to measure compliance with policies. Standards serve as specifications for the implementation of policies. Standards are designed to promote implementation of high-level organization policy rather than to create new policy in themselves.

Baseline: A baseline is a platform-specific security rule that is accepted across the industry as providing the most effective approach to a specific security implementation. Baselines are established to ensure that the security features of commonly used systems are configured and administered uniformly so that a consistent level of security can be achieved throughout the organization.

Procedure: Procedures define specifically how policies, standards, baselines, and guidelines will be implemented in a given situation. Procedures are either technology or process dependent and refer to specific platforms, applications, or processes. They are used to outline steps that must be taken by an organizational element to implement security related to these discrete systems and processes. Procedures are normally developed, implemented, and enforced by the organization owning the process or system. Procedures support organization policies, standards, baselines, and guidelines as closely as possible, while addressing specific technical or procedural requirements within the local organization to which they apply.

Guideline: A guideline is a general statement used to recommend or suggest an approach to implementation of policies, standards, and baselines. Guidelines are essentially recommendations to consider when implementing security. While they are not mandatory in nature, they are to be followed unless there is a documented and approved reason not to.

technologies. *Procedures* define specifically how policies and standards will be implemented in a given situation. *Guidelines* provide recommendations on how other requirements are to be met. An example of interrelated security requirements at each level might be an electronic mail security policy for the entire organization at the highest policy level. This would be supported by various standards, including perhaps a requirement that e-mail messages be routinely purged 90 days following their creation. A baseline in this example would relate to how security controls for the e-mail service will be configured on a specific type of system (e.g., ACF2, VAX VMS, UNIX, etc.). Continuing the example, procedures would be specific requirements for how the e-mail security policy and its supporting standards are to be applied in a given business unit. Finally, guidelines in this example would include guidance to users on best practices for securing information sent or received via electronic mail.

It should be noted that many times the term *policy* is used in a generic sense to apply to security requirements of all types. When used in this fashion it is meant to comprehensively include policies, standards, baselines, guidelines, and procedures. In this document, the reader is reminded to consider the context of the word's use to determine if it is used in a general way to refer to policies of all types or to specific policies at one level of the hierarchy.

Policy Functions

There are 11 functions that must be performed throughout the life of security policy documentation, from cradle to grave. These can be categorized in four fairly distinct phases of a policy's life. During its development a policy is created, reviewed, and approved. This is followed by an implementation phase where the policy is communicated and either complied with or given an exception. Then, during the maintenance phase, the policy must be kept up-to-date, awareness of it must be maintained, and compliance with it must be monitored and enforced. Finally, during the disposal phase, the policy is retired when it is no longer required.

Exhibit 81.2 shows all of these security policy development functions by phase and their relationships through the flow of when they are performed chronologically in the life cycle. The following paragraphs expand on each of these policy functions within these four phases.

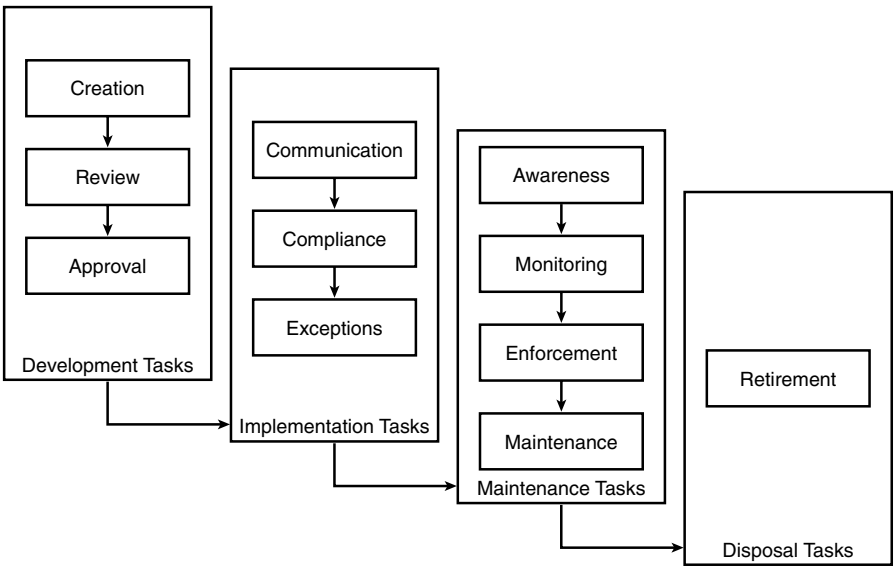


EXHIBIT 81.2 Policy functions.

Creation: Plan, Research, Document, and Coordinate the Policy

The first step in the policy development phase is the planning for, research, and writing of the policy — or, taken together, the *creation* function. The policy creation function includes identifying why there is a need for the policy (for example, the regulatory, legal, contractual, or operational requirement for the policy); determining the scope and applicability of the policy; roles and responsibilities inherent in implementing the policy; and assessing the feasibility of implementing it. This function also includes conducting research to determine organizational requirements for developing policies, (i.e., approval authorities, coordination requirements, and style or formatting standards), and researching industry-standard best practices for their applicability to the current organizational policy need. This function results in the documentation of the policy in accordance with organization standards and procedures, as well as coordination as necessary with internal and external organizations that it affects to obtain input and buy-in from these elements. Overall, policy creation is probably the most easily understood function in the policy development life cycle because it is the one that is most often encountered and which normally requires the readily identifiable milestones.

Review: Get an Independent Assessment of the Policy

Policy *review* is the second function in the development phase of the life cycle. Once the policy document has been created and initial coordination has been effected, it must be submitted to an independent individual or group for assessment prior to its final approval. There are several benefits of an independent review: a more viable policy through the scrutiny of individuals who have a different or wider perspective than the writer of the policy; broadened support for the policy through an increase in the number of stakeholders; and increased policy credibility through the input of a variety of specialists on the review team. Inherent to this function is the presentation of the policy to the reviewer(s) either formally or informally; addressing any issues that may arise during the review; explaining the objective, context, and potential benefits of the policy; and providing justification for why the policy is needed. As part of this function, the creator of the policy is expected to address comments and recommendations for changes to the policy, and to make all necessary adjustments and revisions resulting in a final policy ready for management approval.

Approval: Obtain Management Approval of the Policy

The final step in the policy development phase is the *approval* function. The intent of this function is to obtain management support for the policy and endorsement of the policy by a company official in a position of

authority through their signature. Approval permits and hopefully launches the implementation of the policy. The approval function requires the policy creator to make a reasoned determination as to the appropriate approval authority; coordination with that official; presentation of the recommendations stemming from the policy review; and then a diligent effort to obtain broader management buy-in to the policy. Also, should the approving authority hesitate to grant full approval of the policy, the policy creator must address issues regarding interim or temporary approval as part of this function.

Communication: Disseminate the Policy

Once the policy has been formally approved, it passes into the implementation phase of the policy life cycle. *Communication* of the policy is the first function to be performed in this phase. The policy must be initially disseminated to organization employees or others who are affected by the policy (e.g., contractors, partners, customers, etc.). This function entails determining the extent and the method of the initial distribution of the policy, addressing issues of geography, language, and culture; prevention of unauthorized disclosure; and the extent to which the supervisory chain will be used in communicating the policy. This is most effectively completed through the development of a policy communication, implementation, or rollout plan, which addresses these issues as well as resources required for implementation, resource dependencies, documenting employee acknowledgment of the policy, and approaches for enhancing visibility of the policy.

Compliance: Implement the Policy

Compliance encompasses activities related to the initial execution of the policy to comply with its requirements. This includes working with organizational personnel and staff to interpret how the policy can best be implemented in various situations and organizational elements; ensuring that the policy is understood by those required to implement, monitor, and enforce the policy; monitoring, tracking, and reporting on the pace, extent, and effectiveness of implementation activities; and measuring the policy's immediate impact on operations. This function also includes keeping management apprised of the status of the policy's implementation.

Exceptions: Manage Situations where Implementation Is Not Possible

Because of timing, personnel shortages, and other operational requirements, not every policy can be complied with as originally intended. Therefore, *exceptions* to the policy will probably need to be granted to organizational elements that cannot fully meet the requirements of the policy. There must be a process in place to ensure that requests for exception are recorded, tracked, evaluated, submitted for approval/disapproval to the appropriate authority, documented, and monitored throughout the approved period of noncompliance. The process must also accommodate permanent exceptions to the policy as well as temporary waivers of requirements based on short-term obstacles.

Awareness: Assure Continued Policy Awareness

Following implementation of the policy, the maintenance phase of the policy development life cycle begins. The *awareness* function of the maintenance phase comprises continuing efforts to ensure that personnel are aware of the policy in order to facilitate their compliance with its requirements. This is done by defining the awareness needs of various audience groups within the organization (executives, line managers, users, etc.); determining the most effective awareness methods for each audience group (i.e., briefings, training, messages); and developing and disseminating awareness materials (presentations, posters, mailings, etc.) regarding the need for adherence to the policy. The awareness function also includes efforts to integrate up-to-date policy compliance and enforcement feedback as well as current threat information to make awareness information as topical and realistic as possible. The final task is measuring the awareness of employees with the policy and adjusting awareness efforts based on the results of measurement activities.

Monitoring: Track and Report Policy Compliance

During the maintenance phase, the *monitoring* function is performed to track and report on the effectiveness of efforts to comply with the policy. This information results from observations of employees and supervisors; from formal audits, assessments, inspections, and reviews; and from violation reports and incident response

activities. This function includes continuing activities to monitor compliance or noncompliance with the policy through both formal and informal methods, and the reporting of these deficiencies to appropriate management authorities for action.

Enforcement: Deal with Policy Violations

The compliance muscle behind the policy is effective *enforcement*. The enforcement function comprises management's response to acts or omissions that result in violations of the policy with the purpose of preventing or deterring their recurrence. This means that once a violation is identified, appropriate corrective action must be determined and applied to the people (disciplinary action), processes (revision), and technologies (upgrade) affected by the violation to lessen the likelihood of it happening again. As stated previously, inclusion of information on these corrective actions in the awareness efforts can be highly effective.

Maintenance: Ensure the Policy Is Current

Maintenance addresses the process of ensuring the currency and integrity of the policy. This includes tracking drivers for change (i.e., changes in technology, processes, people, organization, business focus, etc.) that may affect the policy; recommending and coordinating policy modifications resulting from these changes; and documenting policy changes and recording change activities. This function also ensures the continued availability of the policy to all parties affected by it, as well as maintaining the integrity of the policy through effective version control. When changes to the policy are required, several previously performed functions need to be revisited — review, approval, communication, and compliance in particular.

Retirement: Dispense with the Policy when No Longer Needed

After the policy has served its useful purpose (e.g., the company no longer uses the technology for which it applies, or it has been superseded by another policy), then it must be retired. The *retirement* function makes up the disposal phase of the life cycle, and is the final function in the policy development life cycle. This function entails removing a superfluous policy from the inventory of active policies to avoid confusion, archiving it for future reference, and documenting information about the decision to retire the policy (i.e., justification, authority, date, etc.).

These four life-cycle phases comprising 11 distinct functions must be performed in their entirety over the complete life cycle of a given policy. One cannot rule out the possibility of combining certain functions to suit current operational requirements. Nevertheless, regardless of the manner in which they are grouped, or the degree to which they are abbreviated by immediate circumstances, each function needs to be performed. In the development phase, organizations often attempt to develop policy without an independent review, resulting in policies that are not well conceived or well received. Shortsighted managers often fail to appropriately address the exception function from the implementation phase, mistakenly thinking there can be no circumstances for noncompliance. Many organizations fail to continually evaluate the need for their established policies during the maintenance phase, discounting the importance of maintaining the integrity and availability of the policies. One often finds inactive policies on the books of major organizations, indicating that the disposal function is not being applied. Not only do all the functions need to be performed, several of them must be done iteratively. In particular, maintenance, awareness, compliance monitoring, and enforcement must be continually exercised over the full life of the policy.

Policy Responsibilities

In most cases the organization's information security function — either a group or an individual — performs the vast majority of the functions in the policy life cycle and acts as the proponent for most policy documentation related to the protection of information assets. By design, the information security function exercises both long-term responsibility and day-to-day tasks for securing information resources and, as such, should *own* and exercise centralized control over security-related policies, standards, baselines, procedures, and guidelines. This is not to say, however, that the information security function and its staff should be the proponent for all security-related policies or perform all policy development functions. For instance, owners of information systems should have responsibility for establishing requirements necessary to implement organization policies for

their own systems. While requirements such as these must comport with higher-level policy directives, their proponent should be the organizational element that has the greatest interest in ensuring the effectiveness of the policy.

While the proponent or owner of a policy exercises continuous responsibility for the policy over its entire life cycle, there are several factors that have a significant bearing on deciding what individual or element should have direct responsibility for performing specific policy functions in an organization. These factors include the following:

- The principle of *separation of duties* should be applied in determining responsibility for a particular policy function to ensure that necessary checks and balances are applied. To provide a different or broader perspective, an official or group that is independent of the proponent should review the policy, and an official who is senior to the proponent should be charged with approving the policy. Or, to lessen the potential for conflicts of interest, the audit function as an independent element within an organization should be tasked with monitoring compliance with the policy, while external audit groups or organizations should be relied upon to provide an independent assessment of policy compliance to be consistent with this principle.
- Additionally, for reasons of *efficiency*, organizational elements other than the proponent may need to be assigned responsibility for certain security policy development life-cycle functions. For instance, dissemination and communication of the policy is best carried out by the organizational element normally charged with performing these functions for the entire organization, (i.e., knowledge management, corporate communications, etc.). On the other hand, awareness efforts are often assigned to the organization training function on the basis of efficiency, even though the training staff is not particularly well suited to perform the policy awareness function. While the training department may render valuable support during the initial dissemination of the policy and in measuring the effectiveness of awareness efforts, the organization's information security function is better suited to perform continuing awareness efforts because it is well positioned to monitor policy compliance and enforcement activities and to identify requirements for updating the program, each of which is an essential ingredient in effective employee awareness of the policy.
- Limits on *span of control* that the proponent exercises have an impact on who should be the proponent for a given policy function. Normally, the proponent can play only a limited role in compliance monitoring and enforcement of the policy because the proponent cannot be in all places where the policy has been implemented at all times. Line managers, because of their close proximity to the employees who are affected by security policies, are in a much better position to effectively monitor and enforce them and should therefore assume responsibility for these functions. These managers can provide the policy owner assurance that the policy is being adhered to and can ensure that violations are dealt with effectively.
- Limits on the *authority* that an individual or element exercises may determine the ability to successfully perform a policy function. The effectiveness of a policy may often be judged by its visibility and the emphasis that organizational management places on it. The effectiveness of a policy in many cases depends on the authority on which the policy rests. For a policy to have organization-wide support, the official who approves it must have some recognized degree of authority over a substantial part of the organization. Normally, the organization's information security function does not enjoy that level of recognition across an entire organization and requires the support of upper-level management in accomplishing its mission. Consequently, acceptance of and compliance with information security policies is more likely when based on the authority of executive management.
- The proponent's placement in the organization may cause a lack of *knowledge* of the environment in which the policy will be implemented, thus hindering its effectiveness. Employment of a policy evaluation committee can provide a broader understanding of operations that will be affected by the policy. A body of this type can help ensure that the policy is written so as to promote its acceptance and successful implementation, and it can be used to forecast implementation problems and to effectively assess situations where exceptions to the policy may be warranted.
- Finally, the *applicability* of the policy also affects the responsibility for policy life-cycle functions. What portion of the organization is affected by the policy? Does it apply to a single business unit, all users of a particular technology, or the entire global enterprise? This distinction can be significant. If the

applicability of a policy is limited to a single organizational element, then management of that element should own the policy. However, if the policy is applicable to the entire organization, then a higher-level entity should exercise ownership responsibilities for the policy.

The Policy Life-Cycle Model

To ensure that all functions in the policy life cycle are appropriately performed and that responsibilities for their execution are adequately assigned for each function, organizations should establish a framework that facilitates ready understanding, promotes consistent application, establishes a hierarchical structure of mutually supporting policy levels, and effectively accommodates frequent technological and organizational change. Exhibit 81.3 provides a reference for assigning responsibilities for each policy development function according to policy level.

In general, this model proposes that responsibilities for functions related to security policies, standards, baselines, and guidelines are similar in many respects. As the element charged with managing the organization's overall information security program, the information security function should normally serve as the proponent for most related policies, standards, baselines, and guidelines related to the security of the organization's information resources. In this capacity, the information security function should perform the creation, awareness, maintenance, and retirement functions for security policies at these levels. There are exceptions to this general principle, however. For instance, even though it has a substantial impact on the security of information resources, it is more efficient for the human resources department to serve as the proponent for employee hiring policy and standards.

Responsibilities for functions related to security procedures, on the other hand, are distinctly different than those for policies, standards, baselines, and guidelines. Exhibit 81.3 shows that proponents for procedures rest outside the organization information security function and are decentralized based on the limited applicability by organizational element. Although procedures are created and implemented (among other functions) on a decentralized basis, they must be consistent with higher organization security policy; therefore, they should be reviewed by the organization information security function as well as the next-higher official in the proponent element's management chain. Additionally, the security and audit functions should provide feedback to the proponent on compliance with procedures when conducting reviews and audits.

The specific rationale for the assignment of responsibilities shown in the model is best understood through an exploration of the model according to life-cycle functions as noted below.

- *Creation.* In most organizations the information security function should serve as the proponent for all security-related policies that extend across the entire enterprise; and should be responsible for creating these policies, standards, baselines, and guidelines. However, security procedures necessary to implement higher-level security requirements and guidelines should be created by each proponent element to which they apply because they must be specific to the element's operations and structure.
- *Review.* The establishment of a policy evaluation committee provides a broad-based forum for reviewing and assessing the viability of security policies, standards, baselines, and guidelines that affect the entire organization. The policy evaluation committee should be chartered as a group of policy stakeholders drawn from across the organization who are responsible for ensuring that security policies, standards, baselines, and guidelines are well written and understandable, are fully coordinated, and are feasible in terms of the people, processes, and technologies that they affect. Because of their volume, and the number of organizational elements involved, it will probably not be feasible for the central policy evaluation committee to review all procedures developed by proponent elements. However, security procedures require a similar review, and the proponent should seek to establish a peer review or management review process to accomplish this or request review by the information security function within its capability.
- *Approval.* The most significant differences between the responsibilities for policies vis-à-vis standards, baselines, and guidelines are the level of approval required for each and the extent of the implementation. Security policies affecting the entire organization should be signed by the chief executive officer to provide the necessary level of emphasis and visibility to this most important type of policy. Because information security standards, baselines, and guidelines are designed to elaborate on specific policies, this level of policy should be approved with the signature of the executive official subordinate to the CEO who has overall responsibility for the implementation of the policy. The chief information officer

EXHIBIT 81.3 Policy Function–Responsibility Model

	Responsibility			
Function	Policies	Standards and Baselines	Guidelines	Procedures
Creation	Information security function	Information security function	Information security function	Proponent element
Review	Policy evaluation committee	Policy evaluation committee	Policy evaluation committee	Information security function and proponent management
Approval	Chief executive officer	Chief information officer	Chief information officer	Department vice president
Communication	Communications department	Communications department	Communications Department	Proponent element
Compliance	Managers and employees organization-wide	Managers and employees organization-wide	Managers and employees organization-wide	Managers and employees of proponent element
Exceptions	Policy evaluation committee	Policy evaluation committee	Not applicable	Department vice president
Awareness	Information security function	Information security function	Information security function	Proponent management
Monitoring	Managers and employees, information security function, and audit function	Managers and employees, information security function, and audit function	Managers and employees, information security function, and audit function	Managers and employees assigned to proponent element, information security function, and audit function
Enforcement	Managers	Managers	Not applicable	Managers assigned to proponent element
Maintenance	Information security function	Information security function	Information security function	Proponent element
Retirement	Information security function	Information security function	Information security function	Proponent element

will normally be responsible for approving these types of policies. Similarly, security procedures should bear the approval of the official exercising direct management responsibility for the element to which the procedures apply. The department vice president or department chief will normally serve in this capacity.

- *Communication.* Because it has the apparatus to efficiently disseminate information across the entire organization, the communications department should exercise the policy communication responsibility for enterprisewide policies. The proponent should assume the responsibility for communicating security procedures, but as much as possible should seek the assistance of the communications department in executing this function.
- *Compliance.* Managers and employees to whom security policies are applicable play the primary role in implementing and ensuring initial compliance with newly published policies. In the case of organization-wide policies, standards, baselines, and guidelines, this responsibility extends to all managers and employees to whom they apply. As for security procedures, this responsibility will be limited to managers and employees of the organizational element to which the procedures apply.
- *Exceptions.* At all levels of an organization, there is the potential for situations that prevent full compliance with the policy. It is important that the proponent of the policy or an individual or group with equal or higher authority review exceptions. The policy evaluation committee can be effective in screening requests for exceptions received from elements that cannot comply with policies, standards, and baselines. Because guidelines are, by definition, recommendations or suggestions and are not mandatory, formal requests for exceptions to them are not necessary. In the case of security procedures, the lower-level official who approves the procedures should also serve as the authority for approving exceptions to them. The department vice president typically performs this function.
- *Awareness.* For most organizations, the information security function is ideally positioned to manage the security awareness program and should therefore have the responsibility for this function in the case of security policies, standards, baselines, and guidelines that are applicable to the entire organization. However, the information security function should perform this function in coordination with the organization's training department to ensure unity of effort and optimum use of resources. Proponent management should exercise responsibility for employee awareness of security procedures that it owns. Within capability, this can be accomplished with the advice and assistance of the information security function.
- *Monitoring.* The responsibility for monitoring compliance with security policies, standards, baselines, and guidelines that are applicable to the entire organization is shared among employees, managers, the audit function, and the information security function. Every employee who is subject to security requirements should assist in monitoring compliance by reporting deviations that they observe. Although they should not be involved in enforcing security policies, the information security functions and organization audit function can play a significant role in monitoring compliance. This includes monitoring compliance with security procedures owned by lower-level organizational elements by reporting deviations to the proponent for appropriate enforcement action.
- *Enforcement.* The primary responsibility for enforcing security requirements of all types falls on managers of employees affected by the policy. Of course, this does not apply to guidelines, which by design are not enforceable in strict disciplinary terms. Managers assigned to proponent elements to which procedures are applicable must be responsible for their enforcement. The general rule is that the individual granted the authority for supervising employees should be the official who enforces the security policy. Hence, in no case should the information security function or audit function be granted enforcement authority in lieu of or in addition to the manager. Although the information security function should not be directly involved in enforcement actions, it is important that it be privy to reports of corrective action so that this information can be integrated into ongoing awareness efforts.
- *Maintenance.* With its overall responsibility for the organization's information security program, the information security function is best positioned to maintain security policies, guidelines, standards, and baselines having organization-wide applicability to ensure they remain current and available to those affected by them. At lower levels of the organization, proponent elements as owners of security procedures should perform the maintenance function for procedures that they develop for their organizations.

- *Retirement.* When a policy, standard, baseline, or guideline is no longer necessary and must be retired, the proponent for it should have the responsibility for retiring it. Normally, the organization's information security function will perform this function for organization-wide security policies, standards, baselines, and guidelines, while the proponent element that serves as the owner of security procedures should have responsibility for retiring the procedure under these circumstances.

Although this methodology is presented as an approach for developing information security policies specifically, its potential utility should be fairly obvious to an organization in the development, implementation, maintenance, and disposal of the full range of its policies — both security related and otherwise.

Conclusion

The life cycle of a security policy is far more complex than simply drafting written requirements to correct a deviation or in response to a newly deployed technology and then posting it on the corporate intranet for employees to read. Employment of a comprehensive policy life cycle as described here will provide a framework to help an organization ensure that these interrelated functions are performed consistently over the life of a policy through the assignment of responsibility for the execution of each policy development function according to policy type. Utilization of the security policy life-cycle model can result in policies that are timely, well written, current, widely supported and endorsed, approved, and enforceable for all levels of the organization to which they apply.

References

Fites, Philip and Martin P. J. Kratz. *Information Systems Security: A Practitioner's Reference,* International Thomson Computer Press, London, 1996.

Hutt, Arthur E., Seymour Bosworth, and Douglas B. Hoyt. *Computer Security Handbook, 3rd ed.,* John Wiley & Sons, New York, 1995.

National Institute of Standards and Technology, *An Introduction to Computer Security: The NIST Handbook*, Special Publication 800-12, October 1995.

Peltier, Thomas R., *Information Security Policies and Procedures: A Practitioner's Reference*, Auerbach Publications, Boca Raton, FL, 1999.

Tudor, Jan Killmeyer, *Information Security Architecture: An Integrated Approach to Security in the Organization,* Auerbach Publications, Boca Raton, FL, 2001.

Security Awareness Program

Tom Peltier

INTRODUCTION

Development of security policies, standards, procedures, and guidelines is only the beginning of an effective information security program. A strong security architecture will be less effective if there is no process in place to make certain that the employees are aware of their rights and responsibilities. All too often, security professionals implement the "perfect" security program, and then forget to factor the customer into the formula. In order for the product to be as successful as possible, the information security professional must find a way to sell this product to the customers. An effective security awareness program could be the most cost-effective action management can take to protect its critical information assets.

Implementing an effective security awareness program will help all employees understand why they need to take information security seriously, what they will gain from its implementation, and how it will assist them in completing their assigned tasks. The process should begin at new-employee orientation and continue annually for all employees at all levels of the organization.

KEY GOALS OF AN INFORMATION SECURITY PROGRAM

For security professionals there are three key elements for any security program: *integrity*, *confidentiality*, and *availability*. Management wants information to reflect the real world and to have confidence in the information available to them so they can make informed business decisions. One of the goals of an effective security program is to ensure that the organization's information and its information processing resources are properly protected.

The goal of confidentiality extends beyond just keeping the bad guys out; it also ensures that those with a business need have access to the resources they need to get their jobs done. Confidentiality ensures that

controls and reporting mechanisms are in place to detect problems or possible intrusions with speed and accuracy.

DELOITTE & TOUCHE	RATE 1–3	ERNST & YOUNG
1	**Availability**	2
3	**Confidentiality**	3
2	**Integrity**	1

1 = Most Important, 2 = next, 3 = least

Exhibit 12.1. Fortune 500 Managers Rate the Importance of Information

In a pair of recent surveys, the Big Four Accounting firms of Ernst & Young and Deloitte & Touche interviewed Fortune 500 managers and asked them to rank (in importance to them) information availability, confidentiality, and integrity. As can be seen from Exhibit 12.1, the managers felt that information needed to be available when they needed to have access to it. Implementing access control packages that rendered access difficult or overly restrictive is a detriment to the business process. Additionally, other managers felt that the information must reflect the real world. That is, controls should be in place to ensure that the information is correct. Preventing or controlling access to information that was incorrect was of little value to the enterprise.

An effective information security program must review the business objectives or the mission of the organization and ensure that these goals are met. Meeting the business objectives of the organization and understanding the customers' needs are what the goal of a security program is all about. An awareness program will reinforce these goals and will make the information security program more acceptable to the employee base.

KEY ELEMENTS OF A SECURITY PROGRAM

The starting point with any security program is the implementation of policies, standards, procedures, and guidelines. As important as the written word is in defining the goals and objectives of the program and the organization, the fact is that most employees will not have the time or the desire to read these important documents. An awareness program will ensure that the messages identified as important will get to all of those who need them.

Having individuals responsible for the implementation of the security program is another key element. To be most effective, the enterprise will

need to have leadership at a minimum of two levels. There is a strong need to identify a senior level manager to assume the role of Corporate Information Officer (CIO). In a supporting capacity, an information security coordinator responsible for the day-to-day implementation of the information security program and reporting to the CIO is the second key player in the overall security program. Because a security program is more than just directions from the IT organization, each business unit should have its own coordinator responsible for the implementation of the program within that business unit.

The ability to classify information assets according to their relative value to the organization is the third key element in an information security program. Knowing what information an organization has that is sensitive will allow the informed implementation of controls and will allow the business units to use their limited resources where they will provide the most value. Understanding classification levels, employee responsibilities (owner, custodian, user), intellectual property requirements (copyright, trade secret, patent), and privacy rights is critical. An effective awareness program will have to take this most confusing message to all employees and provide training material for all nonemployees needing access to such resources.

The fourth key element is the implementation of the basic security concepts of separation of duties and rotation of assignments. *Separation of duties* — No single individual should have complete control of a business process or transaction from inception to completion. This control concept limits the potential error, opportunity, and temptation of personnel, and can best be defined as segregating incompatible functions (e.g., accounts payable activities with disbursement). The activities of a process are split among several people. Mistakes made by one person tend to be caught by the next person in the chain, thereby increasing information integrity. Unauthorized activities will be limited since no one person can complete a process without the knowledge and support of another. *Rotation of assignments* — Individuals should alternate various essential tasks involving business activities or transactions periodically. There are always some assignments that can cause an organization to be at risk unless proper controls are in place. To ensure that desk procedures are being followed and to provide for staff backup on essential functions, individuals should be assigned to different tasks at regular intervals.

One of the often-heard knocks against rotation of assignments is that it reduces job efficiency. However, it has been proven that an employee's interest declines over time when doing the same job for extended periods. Additionally, employees sometimes develop dangerous shortcuts when they have been in a job too long. By rotating assignments, the organization

can compare the different ways of doing the task and determine where changes should be made.

The final element in an overall security program is an employee awareness program. Each of these elements will ensure that an organization meets its goals and objectives. The employee security awareness program will ensure that the program has a chance to succeed.

SECURITY AWARENESS PROGRAM GOALS

In order to be successful, a security awareness program must stress how security will support the enterprise's business objectives. Selling a security program requires the identification of business needs and how the security program supports those objectives. Employees want to know how to get things accomplished and to whom to turn for assistance. A strong awareness program will provide those important elements.

All personnel need to know and understand management's directives relating to the protection of information and information processing resources. One of the key objectives of a security awareness program is to ensure that all personnel get this message. It must be presented to new employees as well as existing employees. The program must also work with the Purchasing people to ensure that the message of security is presented to contract personnel. It is important to understand that contract personnel need to have this information, but it must be handled through their contract house. Work with Purchasing and Legal to establish the proper process.

All too often the security program fails because there is little or no follow-up. There is usually a big splash with all the fanfare that kicks off a new program. Unfortunately this is where many programs end. Employees have learned that if they wait long enough, the new programs will die from lack of interest or follow-up. It is very important to keep the message in front of the user community and to do this on a regular basis. To assist you in this process, there are a number of "Days" that can be used in conjunction with your awareness program.

- May 10 — International Emergency Response Day
- September 8 — Computer Virus Awareness Day
- November 30 — International Computer Security Day

Keeping the message in front of the user community is not enough. The message must make the issues of security alive and important to all employees. It is important to find ways to tie the message in with the goals and objectives of each department. Every department has different objectives and different security needs. The awareness message needs to reflect those concerns. We will discuss this in more detail shortly.

Find ways to make the message important to employees. When discussing controls, identify how they help protect the employee. When requiring employees to wear identification badges, many security programs tell the employees that this has been implemented to meet security objectives. What does this really mean? What the employees should be told is that the badges ensure that only authorized persons have access to the workplace. By doing this, the company is attempting to protect the employees. Finding out how controls support or protect the company's assets (including the employees) will make the security program message more acceptable.

Finally, a security program is meant to reduce losses associated with either intentional or accidental information disclosure, modification, destruction, and or denial of service. This can be accomplished by raising the consciousness of all employees regarding ways to protect information and information processing resources. By ensuring that these goals are met, the enterprise will be able to improve employee efficiency and productivity.

IDENTIFY CURRENT TRAINING NEEDS

To be successful, the awareness program should take into account the needs and current levels of training and understanding of the employees and management. There are five keys to establishing an effective awareness program. These include:

- Assess the current level of computer usage:
- Determine what the managers and employees want to learn.
- Examine the level of receptiveness to the security program.
- Map out how to gain acceptance.
- Identify possible allies.

To assess the current level of computer usage, it will be necessary to ask questions of the audience. While sophisticated work stations may be found in employees' work areas, their understanding of what these devices can do may be very limited. Ask questions as to what the jobs are and how the tools available are used to support these tasks. It may come as a surprise to find that the most sophisticated computer is being used as a glorified 3270 terminal.

Be an effective listener. Listen to what the users are saying and scale the awareness and training sessions to meet their needs. In the awareness field, one size (or plan) does not fit everyone.

Work with the managers and supervisors to understand what their needs are and how the program can help them. It will become necessary for you to understand the language of the business units and to interpret their needs. Once you have an understanding, you will be able to modify

the program to meet these special needs. No single awareness program will work for every business unit. There must be alterations and a willingness to accept suggestions from nonsecurity personnel.

Identify the level of receptiveness to the security program. Find out what is accepted and what is meeting resistance. Examine the areas of noncompliance and try to find ways to alter the program if at all possible. Do not change fundamental information security precepts just to gain unanimous acceptance; this is an unattainable goal. Make the program meet the greater good of the enterprise and then work with pockets of resistance to lessen the impact.

The best way to gain acceptance is to make your employees and managers partners in the security process. Never submit a new control or policy to management without sitting down with them individually and reviewing the objectives. This will require you to do your homework and to understand the business process in each department. It will be important to know the peak periods of activity in the department and what the manager's concerns are. When meeting with the managers, be sure to listen to their concerns and be prepared to ask for their suggestions on how to improve the program. Remember the key here is to partner with your audience.

Finally, look for possible allies. Find out what managers support the objectives of the security program and identify those who have the respect of their peers. This means that it will be necessary to expand the area of support beyond physical security and the audit staff. Seek out business managers who have a vested interest in seeing this program succeed. Use their support to springboard the program to acceptance.

A key point in this entire process is to never refer to the security program or the awareness campaign as "my program." The enterprise has identified the need for security, and you and your group are acting as the catalysts for moving the program forward. When discussing the program with employees and managers, it will be beneficial to refer to it as "their program" or "our program." Make them feel that they are key stakeholders in this process.

In a presentation used to introduce the security concept to the organization, it may be beneficial to say something like:

> Just as steps have been to taken to ensure the safety of the employees in the workplace, the organization is now asking that the employees work to protect the second most important enterprise asset — information. If the organization fails to protect its information from unauthorized access, modification, disclosure, or destruction, the organization faces the prospect of loss of customer confidence, com-

petitive advantage, and possibly jobs. All employees must accept the need and responsibility to protect our property and assets.

Involve the user community and accept their comments whenever possible. Make information security their program. Use what they identify as important in the awareness program. By having them involved, the program truly becomes theirs and they are more willing to accept and internalize the process.

SECURITY AWARENESS PROGRAM DEVELOPMENT

Not everyone needs the same degree or type of information security awareness to do their jobs. An awareness program that distinguishes between groups of people, and presents only information that is relevant to that particular audience will have the best results. Segmenting the audiences by job function, familiarity with systems, or some other category can improve the effectiveness of the security awareness and acceptance program. The purpose of segmenting audiences is to give the message the best possible chance of success. There are many ways in to segment the user community. Some of the more common methods are provided for you here.

- *Level of Awareness* — Employees may be divided up based on their current level of awareness of the information security objectives. One method of determining levels of awareness is to conduct a "walkabout." A walkabout is conducted after normal working hours and looks for certain key indicators. Look for just five key indicators:
  1. Offices locked
  2. Desks and cabinets locked
  3. Work stations secured
  4. Information secured
  5. Recording media (diskettes, tapes, CDs, cassettes, etc.) Secured
- *Job category* — Personnel may be grouped according to their job functions or titles.
  1. Senior managers (including officers and directors)
  2. Middle management
  3. Line supervision
  4. Employees
  5. Others
- *Specific job function* — Employees and personnel may be grouped according to:
  1. Service providers
  2. Information owners
  3. Users
- *Information processing knowledge* — As discussed above, not every employee has the same level of knowledge on how computers work. A security message for technical support personnel may be very differ-

ent from that for data entry clerks. Senior management may have a very different level of computer skills than their office administrator.
- *Technology, system, or application used* — To avoid "religious wars," it may be prudent to segment the audience based on the technology used. Mac users and users of Intel-based systems often have differing views, as do MVS users and UNIX users. The message may reach the audience faster if the technology used is considered.

Once the audience has been segmented, it will be necessary to establish the roles expected of the employees. These roles may include information owners, custodians of the data and systems, and general users. For all messages it will be necessary to employ the KISS process. That is, Keep It Simple, Sweetie. Inform the audience, but try to stay away from commandments or directives. Discuss the goals and objectives using real-world scenarios. Whenever possible, avoid quoting policies, procedures, standards, or guidelines.

Policies and procedures are boring, and if employees want more information, they can access the documents on the organization intranet. If you feel that you must resort to this method, you have missed the most important tenet of awareness: to identify the business reason *why*. Never tell employees that something is being implemented to "be in compliance with audit requirements." This is, at best, a cop out and fails to explain in business terms why something is needed.

METHODS USED TO CONVEY THE AWARENESS MESSAGE

How do people learn and where do people obtain their information? These are two very important questions to understand when developing an information security awareness program. Each one is different. If we were implementing a training program, we would be able to select from three basic methods of training:

- Buy a book and read about the subject
- Watch a video on the subject
- Ask someone to show you how

For most employees, the third method is best for training. They like the hands-on approach and want to have someone there to answer their questions. With security awareness, the process is a little different. According to findings reported in *USA Today*, over 90 percent of Americans obtain their news from television or radio. To make an awareness program work, it will be necessary to tap into that model.

There are a number of different ways to get the message out to the user community. The key is to make the message stimulating to the senses of the audience. This can be accomplished by using posters, pictures, and

videos. Because so many of our employees use television as their primary source of information, it is important to use videos to reinforce the message. The use of videos will serve several purposes.

With the advent of the news-magazine format so popular in television today, our employees are already conditioned to accept the information presented as factual. This allows us to use the media to present the messages we consider important. Because the audience accepts material presented in this format, the use of videos allows us to bring in an informed outsider to present the message. Many times our message fails because the audience knows the messenger. Being a fellow worker, our credibility may be questioned. A video provides an expert on the subject.

There are a number of organizations that offer computer and information security videos (a listing of how to contact them is included at the end of this chapter). You might want to consider having a senior executive videotape a message that can be run at the beginning of the other video. Costs for creating a quality in-house video can be prohibitive. A 20-minute video that is more than just "talking heads" can run $90,000 to $100,000. Check out the quality and messages of the vendors discussed later in this chapter.

An effective program will also take advantage of brochures, newsletters, or booklets. In all cases, the effectiveness of the medium will depend on how well it is created and how succinct the message is. One major problem with newsletters is finding enough material to fill the pages each time you want to go to print. One way to present a quality newsletter is to look for vendors to provide such material. The Computer Security Institute offers a document titled *Frontline*. This newsletter is researched and written every quarter by CSI's own editorial staff. It provides the space for a column written by your organization to provide information pertinent for your organization. Once the materials are ready, CSI sends out either camera-ready or PDF format versions of the newsletter. The customer is then authorized to make unlimited copies.

As we discussed above, many organizations are requiring business units to name information protection coordinators. One of the tasks of these coordinators is to present awareness sessions for their organizations. An effective way to get a consistent message out is to "train the trainers." Create a security awareness presentation and then bring in the coordinators to train them in presenting the corporate message to their user community. This will ensure that the message presented meets the needs of each organization and that they view the program as theirs.

It will be necessary to identify those employees who have not attended awareness training. By having some form of sign-in or other recording mechanism, the program will be assured of reaching most of the employees. By having the coordinator submit annual reports on the number of

employees trained, the enterprise will have a degree of comfort in meeting its goals and objectives.

PRESENTATION KEY ELEMENTS

While every organization has its own style and method for training, it might help to review some important issues when creating an awareness program. One very important item to keep in mind is that the topic of information security is very broad. Do not get overwhelmed with the prospect of providing information on every facet of information security in one meeting. Remember the old adage, "How do you eat an elephant? One bite at a time."

Prioritize your message for the employees. Start small and build on the program. Remember you are going to have many opportunities to present your messages. Identify where to begin, present the message, reinforce the message, and then build to the next objective. Keep the training session as brief as possible. It is normally recommended to limit these sessions to no more than 50 minutes. There are a number of reasons for this: biology (you can only hold coffee for so long), attention spans, and productive work needs. Start with an attention-grabbing piece and then follow up with additional information.

Tailor the presentations to the vocabulary and skill of the audience. Know to whom you are talking and provide them with information they can understand. This will not be a formal doctoral presentation. The awareness session must take into account the audience and the culture of the organization. Understand the needs, knowledge, and jobs of the attendees. Stress the positive and business side of security — protecting the assets of the organization. Provide the audience with a reminder (booklet, brochure, or trinket) of the objectives of the program.

TYPICAL PRESENTATION FORMAT

In a program that hopes to modify behavior, the three keys are: tell them what you are going to say; say it; and then remind them of what you said. A typical agenda appears in Exhibit 12.2.

Start with an introduction of what information security is about and how it will impact their business units and departments. Follow with a video that will reinforce the message and present the audience with an external expert supporting the corporate message. Discuss any methods that will be employed to monitor compliance to the program and provide the audience with the rationale for the compliance checking. Provide them with a time for questions and ensure that every question either gets an answer or is recorded and the answer provided as soon as possible. Finally, give them some item that will reinforce the message.

Information Security Awareness
Date
Time
Place

Agenda:

Introduction	**CIO**
Goals and Objectives	**ISSO**
Video	
Questions/Answer	**All**
Next Steps	**ISSO**

Exhibit 12.2. Typical Security Awareness Meeting Agenda

WHEN TO DO AWARENESS

Any awareness program must be scheduled around the work patterns of the audience. Take into account busy periods for the various departments and make certain that the sessions do not impact their peak periods. The best times for having these sessions is in the morning on Tuesday, Wednesday, and Thursday. A meeting first-thing Monday morning will impact those trying to get the week's work started. Having the session on Friday afternoon will not be as productive as you would like. Scheduling anything right after lunch is always a worry. The human physiological clock is at its lowest productivity level right after lunch. If you turn out the lights to show a movie, the snoring may drown out the audio. Also, schedule sessions during off-shift hours. Second- and third-shift employees should have the opportunity to view the message during their work hours just as those on the day shift do.

SENIOR MANAGEMENT PRESENTATIONS

While most other sessions will last about an hour, senior management has less time, even for issues as important as this. Prepare a special brief, concise presentation plus in-depth supporting documents. Unlike other presentations, senior management often does not want the "dog and pony show." They may not even want presentation foils to be used. They prefer that you sit with them for a few minutes and discuss the program and how it will help them meet their business objectives.

Quickly explain the purpose of the program, identify any problem areas and what solutions you propose. Suggest a plan of action. Do not go to them with problems for which you do not have a solution. Do not give them a number of solutions and ask them to choose. You are their expert and

they are expecting you to come to them with your informed opinion on how the organization should move forward.

GROUP	BEST TECHNIQUES	BEST APPROACH	EXPECTED RESULTS
Senior Management	Cost justification Industry comparison Audit report Risk analysis	Presentation Video Violation reports	Funding Support
Line Supervisors	Demonstrate job performance benefits Perform security reviews	Presentation Circulate news articles Video	Support Resource help Adherence
Users	Sign responsibility statements Policies and procedures	Presentation Newsletters Video	Adherence Support

Exhibit 12.3. Three Groups

Senior management — will be expecting a sound, rational approach to information security. They will be interested in the overall cost of implementing the policies and procedures and how this program stacks up against others in the industry. A key concern will be how their policies and procedures will be viewed by the audit staff and that the security program will give them an acceptable level of risk.

Line supervisors — These individuals are focused on getting their job done. They will not be interested in anything that appears to slow down their already tight schedule. To win them over, it will be necessary to demonstrate how the new controls will improve their job performance process. As we have been stressing since the beginning, the goal of security is to assist management in meeting the business objectives or mission.

It will be self-defeating to tell supervisors that the new policies are being implemented to allow the company to be in compliance with audit requirements. This is not the reason to do anything, and a supervisor will find this reason useless. Stress how the new process will give the employees the tools they need (access to information and systems) in a timely and efficient manner. Show them where the problem-resolution process is and who to call if there are any problems with the new process.

Employees — are going to be skeptical. They have been through so many company initiatives that they have learned to wait. If they wait long enough and do nothing new, the initiative will generally die on its own. It will be necessary to build employees' awareness of the information security policies and procedures. Identify what is expected of them and how it will assist them in gaining access to the information and systems they need to complete their tasks. Point out that by protecting access to information, they can have a reasonable level of assurance (remember, never use absolutes) that their information assets will be protected from unauthorized access, modification, disclosure, or destruction.

The type of approach chosen will be based on whether your organization has an information security program in place and how active it is. For those organizations with no information security program, it will be necessary to convince management and employees of its importance. For organizations with an existing or outdated program, the key will be convincing management and employees that there is a need for change.

THE INFORMATION SECURITY MESSAGE

The employees need to know that information is an important enterprise asset and is the property of the organization. All employees have a responsibility to ensure that this asset, like all others, must be protected and used to support management-approved business activities. To assist them in this process, employees must be made aware of the possible threats and what can be done to combat those threats. The scope of the program must be identified. Is the program dealing only with computer-held data or does it reach to all information wherever it resides? Make sure the employees know the total scope of the program. Enlist their support in protecting this asset. The mission and business of the enterprise may depend on it.

INFORMATION SECURITY SELF-ASSESSMENT

Each organization will have to develop a process by which to measure the compliance level of the information security program. As part of the awareness process, staff should be made aware of the compliance process. Included for you here is an example of how an organization might evaluate the level of information security within a department or throughout the enterprise.

INFORMATION PROTECTION PROGRAM AND ADMINISTRATION ASSESSMENT QUESTIONNAIRE

Rating Scale

1 = Completed
2 = Being implemented
3 = In development
4 = Under discussion
5 = Haven't begun

FACTORS	RATING/VALUE 1 2 3 4 5
A. ADMINISTRATION	
1. A Corporate Information Officer (CIO) or equivalent level of authority has been named and is responsible for implementing and maintaining an effective IP program.	1 2 3 4 5
2. An individual has been designated as the organization information protection coordinator (OIPC) and has been assigned overall responsibility for the IP program.	1 2 3 4 5
3. The OIPC reports directly to the CIO or equivalent.	1 2 3 4 5
4. IP is identified as a separate and distinct budget item (minimally 1 to 3 percent of the overall ISO budget).	1 2 3 4 5
5. Senior management is aware of the business need for an effective program and is committed to its success.	1 2 3 4 5
6. Each business unit, department, agency, etc., has designated an individual responsible for implementing the IP program for the organization.	1 2 3 4 5
B. PROGRAM	
1. The IP program supports the business objectives or mission statement of the enterprise.	1 2 3 4 5
2. An enterprise-wide IP policy has been implemented.	1 2 3 4 5
3. The IP program is an integral element of the enterprise's overall management practices.	1 2 3 4 5
4. A formal risk analysis process has been implemented to assist management in making informed business decisions.	1 2 3 4 5
5. Purchase and implementation of IP countermeasures are based on cost/benefit analysis utilizing risk analysis input.	1 2 3 4 5
6. The IP program is integrated into a variety of areas both inside and outside the "computer security" field.	1 2 3 4 5
7. Comprehensive information-protection policies, procedures, standards, and guidelines have been created and disseminated to all employees and appropriate third parties.	1 2 3 4 5
8. An ongoing IP awareness program has been implemented for all employees.	1 2 3 4 5
9. A positive, proactive relationship between IP and audit has been established and is actively cultivated.	1 2 3 4 5
C. COMPLIANCE	
1. Employees are made aware that their data processing activities may be monitored.	1 2 3 4 5

FACTORS	RATING/VALUE 1 2 3 4 5
2. An effective program to monitor IP program-related activities has been implemented.	1 2 3 4 5
3. Employee compliance with IP-related issues is a performance appraisal element.	1 2 3 4 5
4. The ITD Project Team members have access to individuals who have leading-edge hardware/software expertise to help the Project Team, as needed.	1 2 3 4 5
5. The application development methodology addresses IP requirements during all phases, including the initiation or analysis (first) phase.	1 2 3 4 5
6. The IP program is reviewed annually and modified where necessary.	1 2 3 4 5
OTHER FACTORS	
1.	1 2 3 4 5
2.	1 2 3 4 5
3.	1 2 3 4 5
TOTAL SCORE	

Interpreting the Total Score: Use this table of risk assessment questionnaire score ranges to assess resolution urgency and related actions.

IF THE SCORE IS...	AND...	THE ASSESSMENT RATE IS …	ACTIONS MIGHT INCLUDE...
21 to 32	• Most activities have been implemented • Most employees are aware of the program	Superior	• Annual reviews and reports to management • Annual recognition days (Computer Security Awareness Day) • Team recognition may be appropriate!
32 to 41	• Many activities have been implemented • Many employees are aware of the program and its objectives	Excellent	• Formal action plan must be implemented • Obtain appropriate sponsorship • Obtain senior management commitment
42 to 62	• Some activities are under development • An IP team has been identified	Solid	• Identify IP program goals • Identify management sponsor • Implement IP policy
63 to 83	• There is a plan to begin planning • Some benchmarking has begun	Low	• Identify roles and responsibilities • Conduct formal risk analysis
84 to 105	• Policies, standards, procedures are missing or not implemented • Management and employees are unaware of the need for a program	Poor	• Conduct risk assessment • Prioritize program elements • Obtain budget commitment • Identify OIPC

CONCLUSION

Information security is more than just policies, standards, procedures, and guidelines. It is more than audit comments and requirements. It is a cultural change for most employees. Before any employee can be required to comply with a security program, he first must become aware of the program. Awareness is an ongoing program that employees must have contact with on at least an annual basis.

Information security awareness does not require huge cash outlays. It does require time and proper project management. Keep the message in front of the employees. Use different methods and means. Bring in outside speakers whenever possible, and use videos to your best advantage.

Video Sources

Commonwealth Films, Inc.
223 Commonwealth Ave.
Boston, MA 02116
617.262.5634
www.commonwealthfilms.com

Mediamix Productions
6812(F) Glenridge Dr.
Atlanta, GA 770.512.7007
www.mediamixus.com

82

Maintaining Management's Commitment

William Tompkins, CISSP, CBCP

After many information security and recovery/contingency practitioners have enjoyed the success of getting their programs off the planning board and into reality, they are then faced with another, possibly more difficult challenge … keeping their organization's program "alive and kicking." More accurately, they seem to be struggling to keep either or both of these programs (business continuity and information security) active and effective.

In many instances, it is getting the initial buy-in from management that is difficult. However, if practitioners "pass the course" (i.e., Management Buy-in 101), they could be faced with a more difficult long-term task: maintaining management's commitment. That "course" could be called Management Buy-in 201. This chapter addresses what can be done beyond initial buy-in, but it will also expand on some of those same initial buy-in principles.

This chapter discusses methods to keep management's attention, keep them involved, and keep all staff members aware of management's buy-in and endorsement. One of the primary requirements to continuing the success of these programs is keeping management aware and committed. When management does not visibly support the program or if they think it is not important, then other employees will not participate.

"What Have You Done for Me Lately?!"

Up to this point in time, most practitioners have not had a manager say this to them, although there have been a few practitioners who have actually heard it from their managers. But, in many instances, the truth is that many managers think of these programs only as a project; that is, the manager thinks "… when this is completed, I can move on to other, more important … ." With this in mind, InfoSec and disaster recovery planners always seem to be under this "sword of Damocles." A key item the practitioner must continually stress is that this is a journey, not a destination.

What does this journey include? This chapter concentrates on four categories:

1. *Communication*. What are we trying to communicate? Who are we communicating with? What message do we want them to hear?
2. *Meetings*. The practitioner will always be meeting with management; so, what should be said to the *different* levels of management we meet with?
3. *Education*. Educating anyone, including management, is a continuous process. What information is it that management should learn?
4. *Motivation*. What one can (or should) use to encourage and inspire management and to keep their support.

Communication

Why is it difficult to communicate with management? "Management does not understand what the practitioner does." "Management is only worried about costs." Or, "Management never listens." These are familiar thoughts with which a practitioner struggles.

The message must be kept fresh in management's mind. However, the underlying issues here are that the practitioner (1) must keep up-to-date, (2) must speak in terms managers can associate with the business, and (3) is obligated to come up with cost-saving ideas (this idea itself may need some work). One more consideration: do managers only pay attention to those who make them look good? Well, yes, but it is not always the same people who appear to make them look good. The practitioner must continuously work at being "the one to make them look good."

Assumptions versus Reality

What to communicate or what to avoid communicating? Both are important, but it is critical in both the security and business continuity professions to avoid assumptions. Many examples can probably be imagined of management and security/BCP (business continuity planning) practitioners suffering from the after-effects of incorrect assumptions.

In the area of disaster recovery planning, it is of paramount importance to ensure that upper management is aware of the actual recovery capabilities of the organization. Management can easily assume that the organization could recover quickly from a crisis — possibly in terms of hours rather than the reality, at a minimum, of days to recover. Management may be assuming that all organizational units have coordinated their recovery plans through the Disaster Recovery Coordinator rather than the reality that business units have been purchasing and installing their own little networks and sub-nets with no thought for organization-wide recovery. Management may be assuming that, regardless of the severity of the disaster, all information would be recovered up to the point of failure when the reality is that the organization might be able to recover using last night's backups but more probable is that the recovery may only be to a point several days previous.

Then there is the flip-side of mistaken assumptions. At a security conference in March 2000, Dr. Eugene Schulz, of Global Integrity Corp., related a story about the peers of a well-respected information security practitioner who believed that this person had a very good security program. Unfortunately, the reality was that senior management in the company was very dissatisfied with the program because the security practitioner had developed it without becoming familiar with the organization's real business processes. This type of dissatisfaction will precipitate the loss of management as stakeholders in the program and loss of budgetary support or, at the least, management will no longer view themselves as a partner in the program development process.

Differing Management Levels … Different Approach

Who a practitioner works with in any organization or, more accurately, who is communicated with should dictate what will be discussed and whatever is said must be in terms that is certain to be understood by any manager. Avoid techno-babble; that is, do not try to teach somebody something they probably will not remember and, typically, not even care to know.

The references used by a practitioner to increase understanding in any topic area must be interpreted into management's terms, that is, terms that management will understand. When possible, stick to basic business principles: cost-benefit and cost-avoidance considerations and business enablers that can be part of an organization's project planning and project management. Unless contingency planning services or information security consulting is the organization's business, it is difficult to show how that company can make a revenue profit from BCP or InfoSec. But, always be prepared to discuss the benefits to be gained and what excessive costs could be avoided if BCP and InfoSec are included in any MIS project plan from the beginning of the project.

Exhibit 82.1 provides some simple examples of cost benefits and cost avoidance (versus return on investment) that most companies can recognize.

EXHIBIT 82.1 Cost Benefits and Cost Avoidance

	BCP	InfoSecurity
Benefits		
Protect the organization	X	X
Maintain the company's reputation	X	X
Assurance of availability	X	
Minimize careless breach of security		X
Maximize effort for intentional breaches		X
Avoidance		
Increase cost for unplanned recovery	X	
Possibly up to four times (or more) of an increase in total project costs to add InfoSec (or BCP) to an application or system that has already been completed	X	X
The cost of being out of business is …?	X	X

The Practitioner(s) … A Business Enabler?

Hopefully, the organization is not in what might be the "typical" recovery posture; that is, information technology (IT) recovery is planned, but not business process recovery. Whatever the requirements for an IT project, the practitioner must continually strive to be perceived as a value-added member of the team and to ensure significant factors (that might keep the business process going) are considered early in development stages of a project. Practitioners will be recognized as business enablers when they do not rely on management's assumptions and they clearly communicate (and document) explicit recovery service level agreements, such as time to recovery (maximum acceptable outage duration), system failure monitoring, uptime guarantees (internal and external), performance metrics, and level-of-service price models.

In today's business world, it is generally accepted that almost all businesses will have some dependence on the Internet. It has become a critical requirement to communicate that the success of the business processes will depend significantly on how quickly the company can recover and restore the automated business process in real-time. Successfully communicating this should increase the comfort level the organization's customers and partners have in the company because it demonstrates how effectively the company controls its online business processes.

Get involved early with "new" system development. It is imperative to do whatever is reasonable to get policy-based requirements for info security and contingency planning considered in the earliest phases of developing a business process. Emphasize that these are part of infrastructure costs — not add-on costs.

Avoid the current trend (organization pitfall, really) of trying to drive the development of a new business process from the IT perspective rather than the reverse. That is, automated business processes should be structured from the perspective of the business needs.

Meetings

As stated, where the practitioner is located within the organizational structure of the company will determine whom to start working with, but first, (1) know the business, (2) know what management desires, and (3) know the technical requirements. Practitioners must have some kind of advance understanding of what their administration will "move" on or they will probably do more harm than good if they try to push an idea that is certain to die on the drawing board (see Exhibit 82.2).

Some of the most important things that should be on the practitioner's mind include:

- What are management's concerns?
- What are the organizational accomplishments?

EXHIBIT 82.2 Introductory Meetings

One of the most important tasks I assign myself when starting at a new organization is to schedule a one-on-one "Introductory Meeting" with as many managers as is possible. The stated objective of this meeting is to get to know the business. I tell each manager that I am not there to discuss my role in the organization, typically because my role is still in its formative stages. I tell them up front that I need to know about *this* section's business processes to become better able to perform my role. Sometimes, I have to remind them that I am really interested in learning about the business process and not necessarily about the IT uses in the section. Next, I ask them if they would suggest someone else in the organization that they feel would be helpful for me to meet to get a more complete "picture" of the organization (a meeting is subsequently scheduled based on this recommendation). Finally, if it seems appropriate, I ask them if they have any security concerns. I try to keep this initial meeting around half an hour long and not more than 45 minutes at the outside. You will find that many times higher level managers will only be able to "squeeze" in 15 minutes or so … take what you can get!

EXHIBIT 82.3 Topics for Discussion

Be prepared to discuss:

- Total cost of recovery
- Moving from EDI on VANs to VPNs
- Total cost of operations
- Voice-over-IP
- Voice recognition systems
- Wireless networking
- Self-healing networks
- IT risk insurance
- Data warehousing impacts
- Charge-back accounting
- BCP and InfoSec at conception
- Virtual Router Redundancy Protocol

- How can I help? Go into any meeting prepared to discuss a long-term strategic plan. Be prepared to discuss short-term tactical efforts. Always be ready to discuss probable budget requirements.

Restating one of the "planks" in the practitioner's management commitment platform, practitioners must keep themselves up-to-date regarding changes in technology. Be prepared to discuss information technology impacts on the organization. Exhibit 82.3 lists just a few of the items with which the practitioner should be familiar.

On the administrative side, the practitioner should always be comfortable discussing policy. Creating or modifying policy is probably one of the most sensitive areas in which one is involved. Typically, it is not within the practitioner's appropriate scope of authority to set policy, but one is expected to make recommendations for and draft policies in one's area of expertise. Here again, the practitioner can be viewed as a value-added part of the team in making recommendations for setting policy; specifically, does the company perform a periodic review of policy (making timely changes as appropriate)? Also, to what level does the organization's policy address those pesky details; for example, does the policy say who is responsible/accountable? Does the policy address compliance; that is, is there a "hammer?" How is the policy enforced? The practitioner should be able to distinguish different levels of policy; for example, at a high level (protect information resources) and at a more detailed level (a policy for use of the WWW or a procedure for recovering a Web site).

Meetings with Executive and Senior Management

When (and if) practitioners get onto the executive committee agenda, they must be prepared! Only you can make yourself look good (or bad) when these opportunities arise. Typically, a status update should be simple and to-the-point: what has been accomplished, what is now happening, and what is in the works. Again, it cannot be over-emphasized that it is important to keep the information relevant to the organization's industry

segment and keep the (planned) presentation brief. Remember: do not try to teach management something they probably are not interested in learning and probably will not remember anyway.

Meeting Mid-level Managers

Try to concentrate on how things have changed since the last meeting with them. For management, what has changed in their business area; for the practitioner, what has changed in continuity and security activities. Ensure that any changes in their recovery or security priorities, due to the changes that have been experienced, are discussed.

It will probably be productive to develop a friendly relationship with the folks in the organization's human resources section. One obvious reason is to promote the inclusion of an information security introduction within the company's new employee orientation program. Another benefit is to try to become informed of "new" managers in the organization. It is also significant to try to find out when a current employee is promoted to a management position and, probably more important, to learn when someone from outside the organization fills an open management position.

Education

A continuing education program is another good example that this is a journey and not a destination. Because one is confronted with almost continual changes in business processes and the technology that supports them, one knows how important it is to continually educate everyone within the organization. Although it may seem to be an uphill battle, it must be emphasized, once again, that one must keep one's company and oneself up-to-date on the vulnerabilities and exposures brought about by new technology.

The practitioner must read the current industry magazines, not only business continuity and information security magazines, but also industry magazines that are relevant to the organization's industry. Articles to support the education efforts must always be close at hand, ready to be provided to management. Also, the practitioner is obligated to inform management of changes in technology as it directly relates to recovery or security. But here, it is necessary to urge caution that these articles will be primarily used with mid-level managers. It is most effective to provide supporting documents (articles, etc.) to senior management only after the executive manager has broached a topic and a clear interest on their part for additional information is perceived.

Another form of "education" can be provided through the use of routine e-mails. Simply "cc:" appropriate managers when sending e-mail within the organization relating to InfoSec/BCP planning tasks.

Be prepared for an opportunity to discuss (or review) the risk management cycle (see Exhibit 82.4). That is, there will be a time when the practitioner is confronted with a "this project is complete" attitude. The practitioner should be ready, at any time, to provide a quick summary of the risk management cycle.

Step 1 Define/update the organization's environment/assets.
Step 2 Perform business impact/risk analyses.
Step 3 Develop/update policies, guidelines, standards, and procedures based on the current organization operations and impacts to the assets.
Step 4 Design and implement systems/processes to reinforce policies, etc. that support the company's mission and goals.
Step 5 Administer and maintain the systems.
Step 6 Monitor the systems and business processes by testing and auditing them to ensure they meet the desired objectives … and as time goes on, the cycle must repeat itself when it is determined (through monitoring, testing and auditing) that things have changed and the company needs to reassess the environment and its assets.

Most companies have regularly scheduled/occurring employee meetings, whether at the lowest levels (e.g., a section meeting) or at the annual/semi-annual employee meetings. The practitioner should attempt to get items of importance added to the agenda of these meetings. Preferably, these presentations will be given by the practitioner to increase recognition within organization. Or, at a minimum, ask management to reinforce these items when they get up to the podium to speak to the employees.

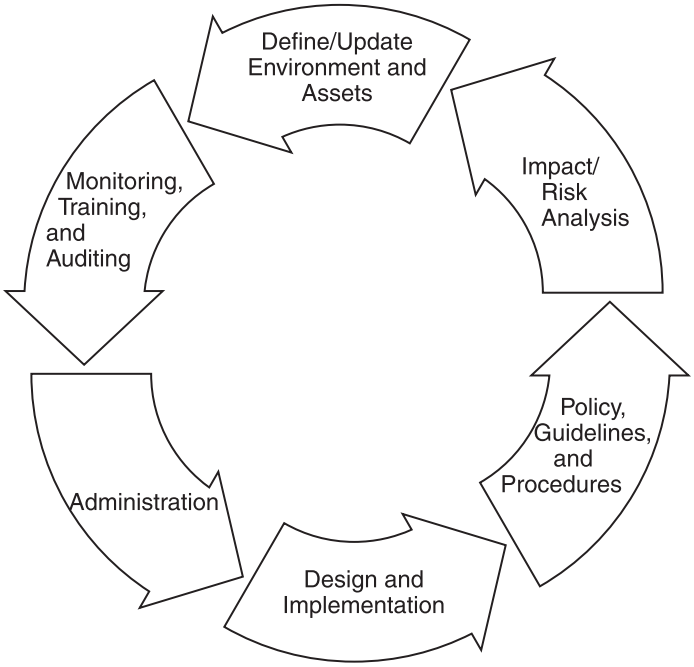


EXHIBIT 82.4 Risk management cycle.

Management Responsibilities

The practitioner must carefully choose the timing for providing some of the following information (education) to managers; but, here again, be ready to emphasize that the success of the continuity/security program is dependent on management's understanding and their support. Management responsibilities include:

- Ensuring that all employees are familiar with IT user responsibilities before accessing any organizational resource
- Leading by example: active, visual support of BCP/InfoSec initiatives
- Praise and reward for those who protect information and improve policies (*Note:* if management is reluctant to do this, then at least try to convince them to allow it to be done, preferably by the practitioner personally.)

Do not overlook the influence that employee involvement can have on management's education. Employee involvement in the program should be encouraged. The employees who recognize that their involvement is a significant factor to the success of an information security or recovery program will enhance a strong self-image. The employee will realize an increased importance to the organization; but most important is that this effort will reinforce the success of the program from the bottom up. When management begins hearing about recovery or security issues from the employees, management will remain (or become more) interested in what is being done for the company.

Motivators

This chapter section reviews the issues that typically stimulate management to action, or at least what will motivate management to support continued recovery and information security planning and the recurring program activities.

There is little argument that the primary management motivator is money. If something increases revenue for the organization, then management is usually happy. Conversely, if doing something costs the organization money and there is no foreseeable return on investment, then management will be much more critical of and less motivated to evaluate and approve the activity. Beyond the issue of finances there are a number of items

EXHIBIT 82.5 Real-World FUD Examples

Tornado	Downtown Ft. Worth, Texas; 6:00 p.m., March 28; downtown area closed until emergency crews investigated buildings and determined structural damage
Hurricane	Gordon; Tampa Bay, Florida; in p.m., September 17, tornadoes and flooding
Fire	Los Alamos, New Mexico; May 12; fires were started by Forest Service officials — intentional brush clearing fires … 11,000 citizens were evacuated (from AP, 5/10/00)
Terrorism	Numerous occurrences: (1) Arab hackers launched numerous attacks in the U.S. and in Israel against Jewish Web sites, (2) Pakistani groups periodically target Web sites in India, etc.
Espionage	QUALCOMM Inc.'s CEO had his laptop stolen from the hotel conference room while at a national meeting; it is suspected the reason for the theft was to obtain the sensitive QUALCOMM info on the laptop (from AP, 9/18/00)
Public image	(embarrassment) In September, during repairs to the Web site, hackers electronically copied over 15,000 credit and debit card numbers belonging to people who used the Western Union Web site (from AP, 9/11/00)

that will motivate management to support the business continuity and information security program(s). Unfortunately, the most used (and abused) method is FUD — Fear, Uncertainty, and Doubt. A subset of FUD could include the aspects of a higher-authority mandate, for example, an edict from the company's Board of Directors or its stockholders. Additionally, the requirements to comply with statutory, regulatory, and contractual obligations are more likely to make an impression on management. A positive motivation factor in management's view is the realization of productivity — if not increased productivity, then at least the assurance that InfoSec and business contingency planning will help ensure that productivity levels remain stable. Fortunately, many practitioners have begun to successfully use due-care motivation. The following chapter subsections review each of these areas of motivation along with some of their details.

FUD = Fear, Uncertainty, and Doubt

One of the fastest things that will get management's attention is an adverse happening; for example, a fire in a nearby office building or an occurrence of a new virus. Exhibit 82.5 identifies only a few of the significant events that occurred in the year 2000.

It Is Easier to Attract Flies with Honey than with Vinegar

Although there are innumerable examples of FUD, the practitioner should be wary of using FUD as a lever to attempt to pry management's support. Maintaining management's commitment is more likely to happen if the practitioner is recognized as an enabler, a person who can be turned to and relied upon as a facilitator, one who provides solutions instead of being the person who makes the proverbial cry, "Wolf!" Granted, there may be an appropriate time to use FUD to advantage, and a case can be made in many organizations that if there was not a real example of FUD to present to management then, subsequently, there would not be any management support for the InfoSec or business contingency program in the first place.

To management, probably the most worrying aspect of FUD is public embarrassment. The specter of bad press or having the company's name appear in newspaper headlines in an unfavorable way is high on management's list of things to avoid. Another example of the practitioner being a facilitator, hopefully to assist in avoiding the possibility of public embarrassment or exposure of a critical portion of the company's vital records, is to be recognized as a mandatory participant in all major information technology projects. Planning must include reliable access management controls and the capability for quick, efficient recovery of the automated business process. During the development of or when making significant changes to an information technology-supported business process within the organization, access controls and recovery planning should be mandatory milestones to be addressed in all projects. Within various organizations, there are differing criteria to determine vital records. A recurring question for management to consider: Does the company want its vital records to become public? In today's rapidly advancing technology environment, the reality is that incomplete planning in a project development life cycle can easily lead to the company's vital records becoming public records.

Due Care

Today's business world is thoroughly (almost totally) dependent on the support information resources provided to its business processes. The practitioner is confronted with the task of protecting and controlling the use of those supporting resources as well as ensuring the organization that these resources will be available when needed. It presents a practitioner with the responsibility to effectively balance protection versus ease of use and the risk of loss versus the cost of security controls. Many practitioners have determined that it is more productive to apply due care analysis in determining the reasonable (and acceptable) balance of these organizational desires, as opposed to trying to convince management of protection and recoverability "minimum" requirements that are based on the inconsistencies that plague a (subjective) risk analysis process.

To summarize due care considerations for any company: Can management demonstrate that (1) security controls and recovery plans have been deployed that are comparable to those found in similar organizations, and (2) they have also made a comparable investment in business continuity/information security? … or else, has the organization documented a good business reason for *not* doing so?

Mandates: Statutory, Regulatory, and Contractual

All organizations are accountable to some type of oversight body, whether it is regulatory (Securities and Exchange Commission, Federal Financial Institutions Examination Council, or Health Care Financial Administration); statutory (Healthcare Insurance Portability and Accountability Act of 1996, IRS Records Retention, and various state and federal computer security and crime acts); an order from the company Board of Directors; or of course, recommendations based on findings in an auditor's report. The practitioner should reasonably expect management to be aware of those rules and regulations that affect their business, but it can only benefit the practitioner to become and remain familiar with these same business influences. Within each company an opportunity will present itself for the practitioner to demonstrate management's understanding of these rules and regulations and to provide management with an interpretation, particularly in relation to how it impacts implementation of information technology-supported business processes.

> *…the hallmark of an effective program to prevent and detect violations of law is that the organization exercised due diligence in seeking to prevent and detect criminal conduct by its employees and other agents…*
>
> — U.S. Sentencing Guidelines, §8A1.2

Every practitioner should also try to be included, or at least provide input, in the contract specifications phase of any large information technology project. Organizations have begun anticipating that E-commerce is a routine part of doing business. In that regard the company is more likely to be confronted with a contractual requirement to allow its external business partners to actually perform a security or disaster recovery assessment of all business partners' security and contingency readiness. Is the practitioner ready to detail the acceptable level of intrusive review into their company's networks? The practitioner can be management's facilitator in this process by expecting the business partners to continue expanding requirements for determining the actual extent of protection in place in the operating environment and then being prepared to provide detailed contractual specifics that are acceptable within their own organization.

Productivity

Automated access management controls … Controlling access is essential if the organization wants to charge for services or provide different levels of service for premier customers and partners. Ensuring a system is properly developed, implemented, and maintained will ensure that only appropriate users access the system and that it is available when the users want to work.

In today's technological work environment, most managers will insist that the information technology section unfailingly install and keep up-to-date, real-time technology solutions. Without automated virus detection and eradication, there is little doubt that the organizational use of information resources might be nonexistent. With virus protection in place and kept up-to-date, employee productivity is, at the least, going to be stable.

There are varying opinions as to whether encryption enhances productivity, but there are few managers who will dispute that it is a business enabler. Encryption enables added confidence in privacy and confidentiality of information transmitted over shared networks, whether these are extranet, intranets, or the Internet. There

is and will continue to be a business need for the confidentiality assurances of encryption. Increasing use of PGP and digital signature advances provides a greater assurance that sensitive or proprietary corporate information can be transmitted over open networks with confidence that the intended recipient will be the only one to view the information.

A basic part of the technology foundation in any organization is being prepared to respond to any computer incident. Having an active and trained response team will minimize downtime and, conversely, lend assurance to increased productivity.

Team-up to Motivate Management

Practitioners typically feel that the auditor is an ally in obtaining management's buy-in, but remember to look at any situation from the auditor's perspective. It is their responsibility to verify that business processes (including continuity and security processes) are performed in a verifiable manner with integrity of the process ensured. This basic premise sets up a conflict of interest when it comes to attempting to involve the auditor in recommendations for developing controls in a business process. But at the same time, it is a very good idea for the practitioner to develop a modified "teaming" relationship with the company's internal audit staff. One of the most likely places to obtain useful organizational information regarding what is successful within the organization and what might stand to be improved is in working in concert with internal audit.

Similarly, the practitioner can be an ally to the legal staff, and vice versa. This "motivator" is not addressed in this chapter as it has been well-documented in earlier editions of this handbook.

Summary

Management says:	You can do this yourself; aren't you the expert?
The practitioners' response:	This will always be a team effort; as much as I know the business, I will never understand the level of detail known by the people who actually do the work.

Practitioners should try to make their own priorities become management's priorities, but more important for the practitioner is to ensure that management's priorities are their own priorities. If the practitioner knows management's concerns and what items management will "move" on, they will be more successful than if they try to make managers accept "requirements" that the managers do not view as important to the success of the business.

The practitioner must strive to be recognized as a facilitator within the organization. The successful practitioner will be the one who can be depended upon to be an effective part of a project team and is relied upon to bring about satisfactory resolution of conflicts, for example, between users' desires (ease of use) and an effective automated business process that contains efficient, programmed controls that ensure appropriate segregation of duties.

It is an old euphemism but with all things considered it should hold a special significance to the practitioner: "The customer is always right." It is a rare situation where the practitioner can force a decision or action that management will not support. If the practitioner makes the effort to know the business and keeps up-to-date with industry changes that impact the organization's business processes, then the practitioner will know what the customer wants. That practitioner will be successful in maintaining management's commitment.

83

Making Security Awareness Happen

Susan D. Hansche, CISSP

Information technology (IT) is apparent in every aspect of our daily life — so much so that in many instances, it seems completely natural. Imagine conducting business without e-mail or voice mail. How about handwriting a report that is later typed using an electric typewriter? Computer technology and open-connected networks are the core components of all organizations, regardless of the industry or the specific business needs.

Information technology has enabled organizations in the government and private sectors to create, process, store, and transmit an unprecedented amount of information. The IT infrastructure created to handle this information flow has become an integral part of how business is conducted. In fact, most organizations consider themselves dependent on their information systems. This dependency on information systems has created the need to ensure that the physical assets, such as the hardware and software, and the information they process are protected from actions that could jeopardize the ability of the organization to effectively perform official duties.

> Several IT security reports estimate that if a business does not have access to its data for more than ten days, it cannot financially recover from the economic loss.

While advances in IT have increased exponentially, very little has been done to inform users of the vulnerabilities and threats of the new technologies. In March 1999, Patrice Rapalus, Director of the Computer Security Institute, noted that "corporations and government agencies that want to survive in the Information Age will have to dedicate more resources to staffing and training of information system security professionals." To take this a step further, not only must information system security professionals receive training, but every employee who has access to the information system must be made aware of the vulnerabilities and threats to the IT system they use and what they can do to help protect their information.

Employees, especially end users of the IT system, are typically not aware of the security consequences caused by certain actions. For most employees, the IT system is a tool to perform their job responsibilities as quickly and efficiently as possible — security is viewed as a hindrance rather than a necessity. Thus, it is imperative for every organization to provide employees with IT-related security information that points out the threats and ramifications of not actively participating in the protection of their information. In fact, federal agencies are required by law (Computer Security Act of 1987) to provide security awareness information to all end users of information systems.

Employees are one of the most important factors in ensuring the security of IT systems and the information they process. In many instances, IT security incidents are the result of employee actions that originate from inattention and not being aware of IT security policies and procedures. Therefore, informed and trained employees can be a crucial factor in the effective functioning and protection of the information system. If employees are aware of IT security issues, they can be the first line of defense in the prevention and early detection of problems. In addition, when everyone is concerned and focused on IT security, the protection of assets and information can be much easier and more efficient.

To protect the confidentiality, integrity, and availability of information, organizations must ensure that all individuals involved understand their responsibilities. To achieve this, employees must be adequately informed

of the policies and procedures necessary to protect the IT system. As such, all end users of the information system must understand the basics of IT security and be able to apply good security habits in their daily work environment. After receiving commitment from senior management, one of the initial steps is to clearly define the objective of the security awareness program. Once the goal has been established, the content must be decided, including the type of implementation (delivery) options available. During this process, key factors to consider are how to overcome obstacles and face resistance. The final step is evaluating success. This chapter focuses on these steps of developing an IT security awareness program.

> The first step in any IT security awareness program is to obtain a commitment from executive management.

Setting the Goal

Before beginning to develop the content of a security awareness program, it is essential to establish the objective or goal. It may be as simple as "all employees must understand their basic security responsibilities" or "develop in each employee an awareness of the IT security threats the organization faces and motivate the employees to develop the necessary habits to counteract the threats and protect the IT system." Some may find it necessary to develop something more detailed, as shown here:

Employees must be aware of:

- Threats to physical assets and stored information
- How to identify and protect sensitive (or classified) information
- Threats to open network environments
- How to store, label, and transport information
- Federal laws they are required to follow, such as copyright violations or privacy act information
- Who they should report security incidents to, regardless of whether it is just a suspected or actual incident
- Specific organization or department policies they are required to follow
- E-mail/Internet policies and procedures

When establishing the goals for the security awareness program, keep in mind that they should reflect and support the overall mission and goals of the organization. At this point in the process, it may be the right (or necessary) time to provide a status report to the Chief Information Officer (CIO) or other executive/senior management members.

Deciding on the Content

An IT security awareness program should create sensitivity to the threats and vulnerabilities of IT systems and also remind employees of the need to protect the information they create, process, transmit, and store. Basically, the focus of an IT security awareness program is to raise the security consciousness of all employees.

The level and type of content are dependent on the needs of an organization. Essentially, one must tell employees what they need to protect, how they should protect it, and how important IT system security is to the organization.

Implementation (Delivery) Options

The methods and options available for delivering security awareness information are very similar to those used for delivering other employee awareness information, such as sexual harassment or business ethics. Although this is true, it may be time to break with tradition and step out of the box — in other words, it may be time to try something new.

Think of positive, fun, exciting, and motivating methods that will give employees the message and encourage them to practice good computer security habits.

Keep in mind that the success of an awareness program is its ability to reach a large audience through several attractive and engaging materials and techniques. Examples of IT security awareness materials and techniques include:

- Posters
- Posting motivational and catchy slogans
- Videotapes
- Classroom instruction
- Computer-based delivery, such as CD-ROM or intranet access
- Brochures/flyers
- Pens/pencils/keychains (any type of trinket) with motivational slogans
- Post-It notes with a message on protecting the IT system
- Stickers for doors and bulletin boards
- Cartoons/articles published monthly or quarterly in in-house newsletter or specific department notices
- Special topical bulletins (security alerts in this instance)
- Monthly e-mail notices related to security issues or e-mail broadcasts of security advisories
- A security banner or pre-logon message that appears on the computer monitor
- Distribution of food items as an incentive. For example, distribute packages of the gummy-bear type candy that is shaped into little snakes. Attach a card to the package, with the heading "Gummy Virus Attack at XYZ." Add a clever message such as: "Destroy all viruses wiggling through the network — make sure your anti-virus software is turned on."

The Web site http://awarenessmaterials.homestead.com/ lists the following options:

- First aid kit with slogan "It's healthy to protect our patient's information; it's healthy to protect our information."
- Mirror with slogan: "Look who is responsible for protecting our information."
- Toothbrush with slogan: "Your password is like this toothbrush; use it regularly, change it often, and do not share it with anyone else."
- Badge holder retractable with slogan: "Think Security"
- Key-shaped magnet with slogan: "You are the key to good security!"
- Flashlight with slogan: "Keep the spotlight on information protection."

Another key success factor in an awareness program is remembering that it never ends — the awareness campaign must repeat its message. If the message is very important, then it should be repeated more often — and in a different manner each time. Because IT security awareness must be an ongoing activity, it requires creativity and enthusiasm to maintain the interest of all audience members. The awareness materials should create an atmosphere that IT security is important not only to the organization, but also to each employee. It should ignite an interest in following the IT security policies and rules of behavior.

An awareness program must remain current. If IT security policies are changing, the employees must be notified. It may be necessary and helpful to set up a technical means to deliver immediate information. For example, if the next "lovebug" virus has been circulating overnight, the system manager could post a pre-logon message to all workstations. In this manner, the first item the users see when turning on the machine is information on how to protect the system, such as what to look for and what not to open.

Finally, the security awareness campaign should be simple. For most organizations, the awareness campaign does not need to be expensive, complicated, or overly technical in its delivery. Make it easy for employees to get the information and make it easy to understand.

Security awareness programs should (be):

- Supported and led by example from management
- Simple and straightforward
- Positive and motivating
- A continuous effort
- Repeat the most important messages
- Entertaining
- Humor, where appropriate; make slogans easy to remember
- Tell employees what the threats are and their responsibilities for protecting the system

In some organizations, it may be a necessary (or viable) option to outsource the design and development of the awareness program to a qualified vendor. To find the best vendor to meet an organization's needs, one can review products and services on the Internet, contact others and discuss their experiences, and seek proposals from vendors that list previous experiences and outline their solutions to the stated goals.

Overcoming Obstacles

As with any employee-wide program, the security awareness campaign must have support from senior management. This includes the financial means to develop the program. For example, each year management must allocate dollars that will support the awareness materials and efforts. Create a project plan that includes the objectives, cost estimates for labor and other materials, time schedules, and outline any specific deliverables (i.e., 15-minute video, pens, pencils, etc.). Have management approve the plan and set aside specific funds to create and develop the security awareness materials.

Keep in mind that some employees will display passive resistance. These are the employees who will not attend briefings and create a negative atmosphere by ignoring procedures and violating security policies. There is also active resistance where an employee may purposefully object to security protections and fights with management over policies. For example, many organizations disable the floppy drive in workstations to reduce the potential of viruses entering the network. If an employee responds very negatively, management may stop disabling the floppy drives. For this reason, management support is important to obtain before beginning any type of security procedures associated with the awareness campaign.

Although one will have resistance, most employees (the author is convinced it is 98 percent) want to perform well in their job, do the right thing, and abide by the rules. Do not let the naysayers affect your efforts — computer security is too important to let a few negative people disrupt achieving good security practices for the organization.

What should one do if frustrated? It is common for companies to agree to an awareness program, but not allocate any human or financial resources. Again, do not be deterred. Plan big, but start small. Something as simple as sending e-mail messages or putting notices in the newsletter can be a cost-effective first step. When management begins to see the effect of the awareness material (of course, they will notice; you will be pointing them out) then the resources needed may be allocated. The important thing is to keep trying and doing all that one can with one's current resources (or lack of them).

Employees are the single most important asset in protecting the IT system. Users who are aware of good security practices can ensure that information remains safe and available.

> Check out the awareness tip from Mike Lambert, CISSP, on his Web page: http://www.frontiernet.net/~mlambert/awareness/. Step-by-step directions and information is provided on how to develop "pop-up announcements." It is a great idea!

Evaluation

All management programs, including the security awareness program, must be periodically reviewed and evaluated. In most organizations, there will be no need to conduct a formal quantitative or qualitative analysis. It should be sufficient to informally review and monitor whether behaviors or attitudes have changed. The following provides a few simple options to consider:

1. Distribute a survey or questionnaire seeking input from employees. If an awareness briefing is conducted during the new-employee orientation, follow up with the employee (after a specified time period of three to six months) and ask how the briefing was perceived (i.e., what do they remember, what would they have liked more information on, etc.).
2. While getting a cup of coffee in the morning, ask others in the room about the awareness campaign. How did they like the new poster? How about the cake and ice cream during the meeting? Remember that the objective is to heighten the employee's awareness and responsibilities of computer security. Thus, even if the response is "that poster is silly," do not fret; it was noticed and that is what is important.
3. Track the number and type of security incidents that occur before and after the awareness campaign. Most likely, it is a positive sign if one has an increase in the number of reported incidents. This is an indication that users know what to do and who to contact if they suspect a computer security breach or incident.

4. Conduct "spot checks" of user behavior. This may include walking through the office checking if workstations are logged in while unattended or if sensitive media are not adequately protected.
5. If delivering awareness material via computer-based delivery, such as loading it on the organization's intranet, record student names and completion status. On a periodic basis, check to see who has reviewed the material. One could also send a targeted questionnaire to those who have completed the online material.
6. Have the system manager run a password-cracking program against the employee's passwords. If this is done, consider running the program on a stand-alone computer and not installing it on the network. Usually, it is not necessary or desirable to install this type of software on one's network server. Beware of some free password-cracking programs available from the Internet because they may contain malicious code that will export one's password list to a waiting hacker.

Keep in mind that the evaluation process should reflect and answer whether or not the original objectives/goals of the security awareness program have been achieved. Sometimes, evaluations focus on the wrong item. For example, when evaluating an awareness program, it would not be appropriate to ask each employee how many incidents have occurred over the last year. However, it would be appropriate to ask each employee if they know who to contact if they suspect a security incident.

Summary

> Employees are the single most important aspect of an information system security program, and management support is the key to ensuring a successful awareness program.

The security awareness program needs to be a line item in the information system security plan of any organization. In addition to the operational and technical countermeasures that are needed to protect the system, awareness (and training) must be an essential item. Various computer crime statistics show that the threat from insiders ranges from 65 to 90 percent. This is not an indication that 65 percent of the employees in an organization are trying to hack into the system; it does mean employees, whether intentionally or accidentally, may allow some form of harm into the system. This includes loading illegal copies of screensaver software, downloading shareware from the Internet, creating weak passwords, or sharing their passwords with others. Thus, employees need to be made aware of the IT system "rules of behavior" and how to practice good computer security skills. Further, in federal organizations, it is a law (Computer Security Act of 1987) that every federal employee must receive security awareness training on an annual basis.

The security awareness program should be structured to meet the organization's specific needs. The first step is deciding on the goals of the program — what it should achieve — and then developing a program plan. This plan should then be professionally presented to management. Hopefully, the program will receive the necessary resources for success, such as personnel, monetary, and moral support. In the beginning, even if there are insufficient resources available, start with the simple and no-cost methods of distributing information. Keep in mind that it is important just to begin, and along the way, seek more resources and ask for assistance from key IT team members.

The benefit of beginning with an awareness campaign is to set the stage for the next level of IT security information distribution, which is IT security training. Following the awareness program, all employees should receive site-specific training on the basics of IT security. Remember that awareness does not end when training begins; it is a continuous and important feature of the information system security awareness and training program.

Training

Training is more formal and interactive than an awareness program. It is directed toward building knowledge, skills, and abilities that facilitate job capabilities and performance. The days of long, and dare one say, boring lectures have been replaced with interactive and meaningful training. The days when instructors were chosen for their specific knowledge, regardless of whether they knew how to communicate that knowledge, have disappeared. Instructional design (i.e., training) is now an industry that requires professionals to know instructional theories, procedures, and techniques. Its focus is on ensuring that students develop skills and practices

that, once they leave the training environment, will be applicable to their job. In addition, training needs to be a motivator; thus, it should spark the student's curiosity to learn more.

During the past decade, the information systems security training field has strived to stay current with the rapid advances of information technologies. One example of this is the U.S. National Institute of Standards and Technology (NIST) document, SP800-16 "IT Security Training Requirements: A Role- and Performance-based Model." This document, developed in 1998, provides a guideline for federal agencies developing IT security training programs. Even if an organization is in the private sector, NIST SP800-16 may be helpful in outlining a baseline of what type and level of information should be offered. For this reason, a brief overview of the NIST document is included in this chapter. Following this overview, the chapter follows the five phases of the traditional instructional systems design (ISD) model for training: needs analysis and goal formation, design, development, implementation, and evaluation. The ISD model provides a systematic approach to instructional design and highlights the important relationship and linkage between each phase. When following the ISD model, a key significant aspect is matching the training objectives with the subsequent design and development of the content material. The ISD model begins by focusing on what the student is to know or be able to do after the training. Without this beginning, the remaining phases can be inefficient and ineffective. Thus, the first step is to establish the training needs and outline the program goals. In the design and development phase, the content, instructional strategies, and training delivery methods are decided. The implementation phase includes the actual delivery of the material. Although the evaluation of the instructional material is usually considered something that occurs after completing the implementation, it should be considered an ongoing element of the entire process. The final section of the article provides a suggested IT security course curriculum. It lists several courses that may be needed to meet the different job duties and roles required to protect the IT system. Keep in mind that course curriculum for an organization should match its identified training needs.

NIST SP800-16 "IT Security Training Requirements: A Role- and Performance-Based Model" (Available from the NIST Web site http://csrc.nist.gov/nistpubs/)

The NIST SP800-16 IT Security Learning Continuum provides a framework for establishing an information systems security training program. It states that after beginning an awareness program, the transitional stage to training is "Security Basics and Literacy." The instructional goal of "Security Basics and Literacy" is to provide a foundation of IT security knowledge by providing key security terms and concepts. This basic information is the basis for all additional training courses.

Although there is a tendency to recognize employees by specific job titles, the goal of the NIST SP800-16 IT Security Learning Continuum is to focus on IT-related job functions and not job titles. The NIST IT Security Learning Continuum is designed for the changing workforce: as an employee's role changes or as the organization changes, the need for IT security training also changes. Think of the responsibilities and daily duties required of a system manager ten years ago versus today. Over the course of time, employees will acquire different roles in relationship to the IT system. Thus, instead of saying the system manager needs a specific course, SP800-16 states that the person responsible for a specific IT system function will need a specific type of training.

Essentially, it is the job function and related responsibilities that will determine what IT system security course is needed. This approach recognizes that an employee may have several job requirements and thus may need several different IT security training classes to meet the variety of duties. It can be a challenge to recognize this new approach and try to fit the standard job categories into this framework. In some organizations, this may not be possible. However, irrespective of the job function or organization, there are several IT security topics that should be part of an IT system security curriculum. Always keep in mind that the training courses that are offered must be selected and prioritized based on the organization's immediate needs.

In an ideal world, each organization would have financial resources to immediately fund all aspects of an IT security training program. However, the reality is that resource constraints will force an evaluation of training needs against what is possible and feasible. In some cases, an immediate training need will dictate the beginning or first set of training courses.

> If one is struggling with how to implement a training program to meet one's needs, training professionals can help to determine immediate needs and provide guidance based on previous experiences and best practices.

Management Buy-In

Before the design and development of course content, one of the first challenges of a training program is receiving support from all levels of the organization, especially senior management. Within any organization are the "training believers" and the "on-the-job-learning believers." In other words, some managers believe that training is very important and will financially support training efforts, while others believe that money should not be spent on training and employees should learn the necessary skills while performing their job duties. Thus, it is an important first step to convince senior managers that company-provided training is valuable and essential.

Senior management needs to understand that training belongs on the top of everyone's list. When employees are expected to perform new skills, the value of training must be carefully considered and evaluated.

To help persuade senior management of the importance of sponsoring training, consider these points:

1. *Training helps provide employee retention.* To those who instantly thought that, "No, that is not right; we spend money to train our employees and then they leave and take those skills to another company," there is another side. Those employees will leave anyway; but, on average, employees who are challenged by their job duties (and … satisfied with their pay) and believe that the company will provide professional growth and opportunities will stay with the company.
2. *Find an ally in senior management who can be an advocate.* When senior managers are discussing business plans, it is important to have someone speak positively about training programs during those meetings.
3. *Make sure the training program reflects the organizational need.* In many instances, one will need to persuade management of the benefits of the training program. This implies that one knows the weaknesses of the current program and that one can express how the training program will overcome the unmet requirements.
4. *Market the training program to all employees.* Some employees believe they can easily learn skills and do not need to take time for training. Thus, it is important to emphasize how the training will meet the employee's business needs.
5. *Start small and create a success.* Management is more likely to dedicate resources to training if an initial program has been successful.
6. *Discover management's objections.* Find out the issues and problems that may be presented. Also, try to find out what they like or do not like in training programs; then make sure the training program used will overcome these challenges. Include management's ideas in the program; although one may not be able to please everyone, it is a worthy goal to meet most everyone's needs.

Be an enthusiastic proponent. If one does not believe in the training program and its benefits, neither will anyone else.

Establishing the Information System Security Training Need

After receiving management approval, the next step in the development of a training program is to establish and define the training need. Basically, a training need exists when an employee lacks the knowledge or skill to perform an assigned task. This implies that a set of performance standards for the task must also exist. The creation of performance standards is accomplished by defining the task and the knowledge, skills, abilities, and experiences (KSA&Es) needed to perform the task. Then compare what KSA&Es the employees currently possess with those that are needed to successfully perform the task. The differences between the two are the training needs.

In the information systems security arena, several U.S. Government agencies have defined a set of standards for job functions or tasks. In addition to the NIST SP800-16, the National Security Telecommunications and Information Systems Security Committee (NSTISSC) has developed a set of INFOSEC training standards. For example, the NSTISSC has developed national training standards for four specific IT security job functions: Information Systems Security Professionals (NSTISSC #4011); the Designated Approving Authority (NSTISSIC #4012); System Administrator in Information System Security (NSTISSC #4013); and Information System

Security Officer (NSTISSC #4014). The NIST and NSTISSC documents can be helpful in determining the standards necessary to accomplish the information system security tasks or responsibilities.

Once the needs analysis has been completed, the next step is to prioritize the training needs. When making this decision, several factors should be considered: legal requirements; cost-effectiveness; management pressure; the organization's vulnerabilities, threats, information sensitivity, and risks; and who is the student population. For some organizations (i.e., federal agencies, banking, health care), the legal requirements will dictate some of the decisions about what training to offer. To determine cost-effectiveness, think about the costs associated with an untrained staff. For example, the costs associated with a network failure are high. If an information system is shut down and the organization's IT operations cease to exist for an extended period of time, the loss of money and wasted time would be enormous. Thus, training system administrators would be a high priority. Executive pressures will come from within, usually the Chief Information Officer (CIO) or IT Security Officer. If an organization has conducted a risk assessment, executive-level management may prioritize training based on what it perceives as the greatest risks. Finally, and what is usually the most typical determining factor, training is prioritized based on the student population that has the most problems or the most immediate need.

Due to the exponential technological advances, information system security is continually evolving. As technology changes, so do the vulnerabilities and threats to the system. Taking it one step further, new threats require new countermeasures. All of these factors necessitate the continual training of IT system professionals. As such, the IT Security Training Program must also evolve and expand with the technological innovations.

In conducting the needs analysis, defining the standards, prioritizing the training needs, and finalizing the goals and objectives, keep in mind that when beginning an information system security training program, it is necessary to convince management and employees of its importance. Also, as with all programs, the training program's success will be its ability to meet the organization's overall IT security goals, and these goals must be clearly defined in the beginning of the program.

Developing the Program Plan

Once the training needs are known, the plan for the training program can be developed. The program plan outlines the specific equipment, material, tasks, schedule, and personnel and financial resources needed to produce the training program. The program plan provides a sequence and definition of the activities to be performed, such as deliverables for specific projects. One of the most common mistakes that training managers make is thinking they do not need a plan.

Remember this common saying: If you do not plan your work, you cannot work your plan.

Another mistake is not seeking approval from senior management for the program plan. An integral part of program planning is ensuring that the plan will work. Thus, before moving to the next step, review the plan with senior managers. In addition, seeking consensus and agreement at this stage allows others to be involved and feel a part of the process — an essential component of success.

Instructional Strategy (Training Design and Development)

The design of the training program is based on the learning objectives. The learning objectives are based on the training needs. Thus, the instructional strategy (training delivery method) is based on the best method of achieving the learning objectives.

In choosing an instructional strategy, the focus should be on selecting the best method for the learning objectives, the number of students, and the organization's ability to efficiently deliver the instructional material. The key is to understand the learning objectives, the students, and the organization.

During the design and development phase, the content material is outlined and developed into instructional units or lessons. Remember that content should be based on what employees need to know and do to perform their job duties. During the needs analysis, one may have established the tasks and duties for specific job functions. If the content is not task-driven, the focus is on what type of behaviors or attitudes are expected. This involves defining what performance employees would exhibit when demonstrating the objective and what is needed to accomplish the goal. The idea is to describe what someone would do or display to be considered competent in the behavior or attitude.

The course topics must be sequenced to build new or complex skills onto existing ones and to encourage and enhance the student's motivation for learning the material.

A well-rounded information system security training program will involve multiple learning methods. When making a decision about the instructional strategy, one of the underlying principles should be to choose a strategy that is as simple as possible while still achieving the objectives. Another factor is the instructional material itself; not all content fits neatly into one type of instructional strategy. That is, for training effectiveness, look at the learning objectives and content to determine what would be the best method for students to learn the material. One of the current philosophies for instructional material is that it should be "edutainment," which is the combination of education and entertainment. Because this is a hotly debated issue, the author's advice is not to get cornered into taking a side. Look at who the audience will be, what the content is, and then make a decision that best fits the learning objective.

When deciding on the method, here are a few tips:

- *Who is the audience?* It is important to consider the audience size and location. If the audience is large and geographically dispersed, a technology-based solution (i.e., computer-based [CD-ROM] or Web-based training [delivery over the Internet]) may be more efficient.
- *What are the business needs?* For example, if a limited amount of travel money is available for students, then a technology-based delivery may be applicable. Technology-based delivery can reduce travel costs. However, technology-based training usually incurs more initial costs to design and develop; thus, some of the travel costs will be spent in developing the technology-based solution.
- *What is the course content?* Some topics are better suited for instructor-led, video, Web, or CD-ROM delivery. Although there are many debates as to the best delivery method (and everyone will have an opinion), seek out the advice of training professionals who can assess the material and make recommendations.
- *What type of learner interaction is necessary?* Is the course content best presented as self-paced individual instruction or as group instruction? Some instructional materials are better suited for face-to-face and group interaction, while other content is best suited for creative, interactive, individualized instruction. For example, if students are simply receiving information, a technology-based solution may be more appropriate. If students are required to perform problem-solving activities in a group, then a classroom setting would be better.
- *What types of presentations or classroom activities need to be used?* If the course content requires students to install or configure an operating system, a classroom lab might be best.
- *How stable is the instructional material?* The stability of content can be a cost issue. If content will change frequently, the expense of changing the material must be estimated in difficulty, time, and money. Some instructional strategies can be revised more easily and cost-efficiently than others.
- *What type of technology is available for training delivery?* This is a critical factor in deciding the instructional strategy. The latest trend is to deliver training via the Internet or an intranet. For this to be successful, students must have the technological capability to access the information. For example, in instances where bandwidth could limit the amount of multimedia (e.g., audio, video, and graphic animations) that can be delivered, a CD-ROM solution may be more effective.

Regardless of the instructional strategy, there are several consistent elements that will be used to present information. This includes voice, text, still or animated pictures/graphics, video, demonstrations, simulations, case studies, and some form of interactive exercises. In most courses, several presentation methods are combined. This allows for greater flexibility in reaching all students and also for choosing the best method to deliver the instructional content. If unfamiliar with the instructional strategies available, refer to the appendices in Chapter 85 for a detailed definition of instructor-led and technology-based training delivery methods.

While deciding on what type of instructional strategy is best suited for the training needs, it is necessary to explore multiple avenues of information. Individuals should ask business colleagues and training professionals about previous training experiences and then evaluate the responses. Keep in mind that the instructional strategy decision must be based on the instructional objectives, course content, delivery options, implementation options, technological capabilities, and available resources, such as time and money.

Possible Course Curriculum

Appendix B in Chapter 84 contains a general list of IT security topics that can be offered as IT system security training courses. The list is intended to be flexible; remember that as technologies change, so will the types of courses. It merely represents the type of training courses that an organization might consider. Additionally, the course content should be combined and relabeled based on the organization's particular training needs.

The appendices in Chapter 84 contain more detailed information for each course, including the title, brief description, intended audience, high-level list of topics, and other information as appropriate. The courses listed in Appendix B are based on some of the skills necessary to meet the requirements of an information system security plan. It is expected that each organization will prioritize its training needs and then define what type of courses to offer. Because several of these topics (and many more) are available from third-party training companies, it is not necessary to develop custom courses for an organization. However, the content within these outside courses is general in nature. Thus, for an organization to receive the most effective results, the instructional material should be customized by adding one's own policies and procedures. The use of outside sources in this customization can be both beneficial and cost-effective for the organization.

Evaluating the Information System Security Training Plan

Evaluating training effectiveness is an important element of an information system security training plan. It is an ongoing process that starts at the beginning of the training program. During all remaining phases of the training program, whether it is during the analysis, design, development, or implementation stage, evaluation must be built into the plan.

Referring back to NIST SP800-16, the document states that evaluating training effectiveness has four distinct but interrelated purposes to measure:

1. The extent that conditions were right for learning and the learner's subjective satisfaction
2. What a given student has learned from a specific course
3. A pattern of student outcomes following a specified course
4. The value of the class compared to other options in the context of an organization's overall IT security training program

Further, the evaluation process should produce four types of measurement, each related to one of the evaluation's four purposes. Evaluation should:

1. Yield information to assist the employees themselves in assessing their subsequent on-the-job performance
2. Yield information to assist the employee's supervisors in assessing individual students' subsequent on-the-job performance
3. Produce trend data to assist trainers in improving both learning and teaching
4. Produce return-on-investment statistics to enable responsible officials to allocate limited resources in a thoughtful, strategic manner among the spectrum of IT security awareness, security literacy, training, and education options for optimal results among the workforce as a whole

To obtain optimal results, it is necessary to plan for the collection and organization of data, and then plan for the time an analyst will need to evaluate the information (data) and extrapolate its meaning to the organization's goals.

One of the most important elements of effective measurement and evaluation is selecting the proper item to measure. Thus, regardless of the type of evaluation or where it occurs, the organization must agree on what it should be evaluating, such as perceptions, knowledge, or a specific set of skills.

> Because resources, such as labor hours and monies, are at a premium for demand, the evaluation of the training program must become an integral part of the training plan.

Keep in mind that evaluation has costs. The costs involve thought, time, energy, and money. Therefore, evaluation must be thought of as an ongoing, integral aspect of the training program and both time and money must be budgeted appropriately.

Summary

IT system security is a rapidly evolving, high-risk area that touches every aspect of an organization's operations. Both companies and federal agencies face the challenge of providing employees with the appropriate awareness, training, and education that will enable employees to fulfill their responsibilities effectively and to protect the IT system assets and information.

> Employees are an organization's greatest asset, and trained employees are crucial to the effective functioning and protection of the information system.

This chapter has outlined the various facets of developing an information system (IS) security training program. The first step is to create an awareness program. The awareness program helps to set the stage by alerting employees to the issues of IT security. It also prepares users of the IT system for the next step of the security training program — providing the basic concepts of IT security to all employees. From this initial training effort, various specialized and detailed training courses should be offered to employees. These specific training courses must be related to the various job functions that occur within an organization's IT system security arena.

Critical to the success of a training program is having senior management's support and approval. During each step of the program's life cycle, it is important to distribute status reports to keep all team members and executive-level managers apprised of progress. In some instances, it may be important (or necessary) to receive direct approval from senior management before proceeding to the next phase.

The five steps of the instructional process are relevant to all IS security training programs. The first step is to analyze the training needs and define the goals and objectives for the training program. Once the needs have been outlined, the next step is to start designing the course. It is important to document this process into some type of design document or blueprint for the program. Because the design document provides the direction for the course development, all parties involved should review and approve the design document before proceeding.

The development phase involves putting all the course elements together, such as the instructor material, student material, classroom activities, or if technology-based, storyboarding and programming of media elements. Once course development has been completed, the first goal of the implementation phase is to begin with a pilot or testing of the materials. This allows the instructional design team to evaluate the material for learner effectiveness and rework any issues prior to full-scale implementation. Throughout the IS security training program, the inclusion of an evaluation program is critical to the program's success. Resources, such as time and money, must be dedicated to evaluate the instructional material in terms of effectiveness and meeting the learning and company's needs. Keep in mind that the key factor in an evaluation program is its inclusion throughout the design, development, and implementation of the IT security training program.

Several examples of training courses have been suggested for an IS security training program. Remember that as technology changes, the course offerings required to meet the evolving IT security challenges must also change. These changes will necessitate modifications and enhancements to current courses. In addition, new courses will be needed to meet the ever-changing IT system advances and enhancements. Thus, the IS security training program and course offerings must be flexible to meet the new demands.

Each organization must also plan for the growth of the IT professional. IT security functions have become technologically and managerially complex. Companies are seeking educated IT security professionals who can solve IT security challenges and keep up with the changing technology issues. Currently, there is a lack of IT security professionals in the U.S. workforce; thus, organizations will need to identify and designate appropriate individuals as IT security specialists and train them to become IT security professionals capable of problem-solving and creating vision.

As one faces the challenges of developing an information system security training program, it is important to remember that the process cannot be accomplished by one person working alone. It requires a broad, cross-organizational effort that includes the executive level bringing together various divisions to work on projects. By involving everyone in the process, the additional benefit of creating ownership and accountability is established. Also, the expertise of both training personnel (i.e., training managers, instructional designers, and trainers) and IT security specialists are needed to achieve the training goals.

> Always remember the end result: "A successful IT security training program can help ensure the integrity, availability, and confidentiality of the IT system assets and its information — the first and foremost goal of IT security."

84

Making Security Awareness Happen: Appendices

Susan D. Hansche, CISSP

Appendix A: Instructional Strategies (Training Delivery Methods)

Instructor-Led

The traditional instructional strategy is instructor-led and considered a group instruction strategy. This involves bringing students together into a common place, usually a classroom environment, with an instructor or facilitator. It can provide considerable interaction between the instructor and the students. It is usually the least expensive as far as designing and development of instructional material. However, it can be the most expensive during implementation, especially if it requires students to travel to a central location.

Text-Based

Text-based training is an individual, self-paced form of training. The student reads a standard textbook (or any book) on the training content. Text-based training does not allow for interaction with an instructor. However, the book's information is usually written by an individual with expertise in the subject matter. In addition, students can access the material when it is needed and can review (or re-read) sections as needed.

Paper-Based or Workbook

Paper-based or workbook training is a type of individual, self-paced instruction. It is the oldest form of distance learning (i.e., correspondence courses). Workbooks include instructional text, graphical illustrations, and practice exercises. The workbooks are written specifically to help student's learn particular subjects or techniques. The practice exercises help students remember what is covered in the books by giving them an opportunity to work with the content. In some cases, students may be required to complete a test or exam to show competency in the subject.

Video-Based

Video-based training is usually an individual, self-paced form of instruction. The information is provided on a standard VHS video cassette tape that can be played using a standard VHS video cassette recorder (VCR). If used as a self-paced form of instruction, it does not allow for interaction with the instructor. However, if used

in the classroom, a video can be discussed and analyzed as an interactive exercise. Video does allow for animated graphics that can show processes or a demonstration of step-items. It is flexible as far as delivery time and location, and if necessary, can be repeated.

Technology-Based, Including CBT and WBT

Technology-based training is also an individual, self-paced instructional strategy. It is any training that uses a computer as the focal point for instructional delivery. With technology-based training, instructional content is provided through the use of a computer and software that guides a student through an instructional program.

This can be either computer-based training delivered via a floppy disk, CD-ROM, or loaded on a server; or Web-based training delivered via the Internet or an intranet.

Computer-based training (CBT) involves several presentation methods, including tutorials, practice exercises, simulations or emulations, demonstrations, problem-solving exercises, and games. CBT has many positive features that can be of importance to agencies that need to deliver a standard set of instructional material to a large group of students who are in geographically separate areas. The benefits of CBT include immediate feedback, student control of instructional material, and the integration of multimedia elements such as video, audio, sounds, and graphical animations.

After the initial CBT development costs, CBT can be used to teach any number of students at any time. Customized CBT programs can focus only on what students need to learn, thus training time and costs can be significantly reduced. In addition, CBT can enable one to reduce or eliminate travel for students; thus, total training costs can also be reduced. As a self-paced, individualized form of instruction, CBT provides flexibility for the student. For example, the student can control the training environment by selecting specific lessons or topics. In addition, for some students, the anonymous nature of CBT can be nonthreatening.

Although CBT has many benefits, it is important to remember that CBT is not the answer to all training needs. It some situations, it can be more appropriate, effective, and cost-efficient. However, in other situations, it may produce a negative student attitude and destroy the goodwill and goals of the training program. For example, students who are offered CBT courses and instructed to fit it in to their schedule may believe they are expected to complete the training outside of the workday. These same students know that taking an instructor-led course allows them to complete the training during a workday. Therefore, they may view CBT as an unfair time requirement.

CBT includes computer-assisted learning (CAL), which uses a computer as a tool to aid in a traditional learning situation, such as classroom training. The computer is a device to assist the instructor during the training process, similar to an overhead projector or handouts. It also includes computer-assisted testing (CAT), which assesses an individual through the medium of a computer. Students take the test at the computer, and the computer records and scores the test. CAT is embedded in most computer-based training products.

Web-based training (WBT) is a new, creative method for delivering computer-based training to widespread, limitless audiences. WBT represents a shift from the current delivery of CBT. In the CBT format, the information is usually stored on the local machine, server, or a CD-ROM. In WBT, the information is distributed via the World Wide Web (WWW) and most likely is stored at a distant location or an agency's central server. The information is displayed to the user using a software application called a browser, such as Internet Explorer. The content is presented by text, graphics, audio, video, and graphical animations. WBT has many of the same benefits as CBT, including saving time and easy access. However, one of the key advantages of WBT over CBT is the ease of updating information. If changes need to be made to instructional material, the changes are made once to the server, and then everyone can access the new information. The challenges of WBT are providing the technical capability for the student's computer, the agency's server, and the available bandwidth.

Appendix B: Suggested IT System Security Training Courses

What follows is a description of suggested IT system security training courses; these are summarized in Exhibit 84.1

INFOSEC 101: IT Security Basics

Brief Description

This course should describe the core terms and concepts that every user of the IT system must know, the fundamentals of IT security and how to apply them, plus the IT system security rules of behavior. This will allow all individuals to understand what their role is in protecting the IT systems assets and information.

Intended Audience

This course is intended for all employees who use the IT system, regardless of their specific job responsibilities. Essentially, all employees should receive this training.

List of Topics

What Is IT Security and Why Is It Important; Federal Laws and Regulations; Vulnerabilities, Threats, and Sensitivity of the IT System; Protecting the Information, Including Sensitive but Unclassified and Classified Information; Protecting the Hardware; Password Protections; Media Handling (i.e., how to process, store, and dispose of information on floppy disks); Copyright Issues; Laptop Security; User Accountability; Who to Contact with Problems; and other specific agency policies related to all users of the IT system. Note that if the agency processes classified information, a separate briefing should be given.

Note: Because most agencies will require this course for all employees, it is a good example of content that should be delivered via a technology-based delivery. This includes either video, computer-based training via CD-ROM, or Web-based training via the agency's intranet.

INFOSEC 102: IT Security Basics for a Network Processing Classified Information

Brief Description

This course describes the core terms and concepts that every user of the IT system must know, the fundamentals of IT security and how to apply them, and the rules of behavior. It is similar to INFOSEC 101 except that it also provides information pertinent to employees who have access to a network processing classified information.

Intended Audience

This course is intended for all employees with access to a network processing classified information.

List of Topics

What Is IT Security and Why Is It Important; Federal Laws and Regulations; Vulnerabilities, Threats, and Sensitivity of the IT System; Protecting Classified Information; Protecting the Hardware, Including TEMPEST Equipment; Password Protections; Media Handling (i.e., how to process, store, and dispose of classified information); Copyright Issues; Laptop Security; User Accountability; Who to Contact with Problems; and other specific agency policies related to users of a classified IT system.

INFOSEC 103: IT Security Basics — Annual Refresher

Brief Description

This is a follow-on course to the IT Security Basics (INFOSEC 101). As technology changes, the demands and challenges for IT security also change. In this course, the agency will look at the most critical challenges for the end user. The focus of the refresher course will be on how to meet those needs.

Intended Audience

This course is for all employees who use the IT system.

List of Topics

The topics would be specific to the agency and the pertinent IT security challenges it faces.

EXHIBIT 84.1 Suggested Information Technology System Security Training Courses

Course Number and Content Level	Course Title	Intended Audience	Possible Prerequisite
INFOSEC 101 Basic	IT Security Basics	All employees	None
INFOSEC 102 Basic	IT Security Basics for Networks Processing Classified Information	All employees with access to a network processing classified information	None
INFOSEC 103 Basic	IT Security Basics — Annual Refresher	All employees	INFOSEC 101
INFOSEC 104 Basic	Fundamentals of IT Security	Individuals directly responsible for IT security	None
INFOSEC 201 Intermediate	Developing the IT System Security Plan	Individuals responsible for developing the IT system security plan	INFOSEC 101 or 103
INFOSEC 202 Intermediate	How to Develop an IT System Contingency Plan	Individuals responsible for developing the IT system contingency plan	INFOSEC 101 or 103
INFOSEC 203 Intermediate	System/Technical Responsibilities for Protecting the IT System	Individuals responsible for the planning and daily operations of the IT system	INFOSEC 101 or 103
INFOSEC 204 Intermediate	Life Cycle Planning for IT System Security	Managers responsible for the acquisition and design of the IT system	INFOSEC 101 or 103
INFOSEC 205 Intermediate	Basic Information System Security Officer (ISSO) Training	Individuals assigned as the ISSO or alternate ISSO	INFOSEC 101 or 103
INFOSEC 206 Intermediate	Certifying the IT System	Individuals responsible for the Designated Approving Authority (DAA) role	INFOSEC 101 or 103 INFOSEC 203
INFOSEC 207 Intermediate	Information System Security for Executive Managers	Executive-level managers	None
INFOSEC 208 Intermediate	An Introduction to Network and Internet Security	Individuals responsible for network connections	INFOSEC 101 or 103 INFOSEC 203
INFOSEC 209	An Introduction to Cryptography	Individuals responsible for network connections information and security	INFOSEC 101 or 103 INFOSEC 203 or 205
INFOSEC 301 Advanced	Understanding Audit Logs	Individuals responsible for reviewing audit logs	INFOSEC 101 or 103 INFOSEC 203 or 205
INFOSEC 302 Advanced	Windows NT 4.0 Security	Individuals responsible for networks using Windows NT 4.0	INFOSEC 101 or 103 INFOSEC 203

INFOSEC 303 Advanced	Windows 2000 Security	Individuals responsible for networks using Windows 2000	INFOSEC 101 or 103 INFOSEC 203
INFOSEC 304 Advanced	UNIX Security	Individuals responsible for networks using UNIX	INFOSEC 101 or 103 INFOSEC 203
INFOSEC 305 Advanced	Advanced ISSO Training	Individuals assigned as the ISSO or alternate ISSO	INFOSEC 205
INFOSEC 306 Advanced	Incident Handling	Individuals responsible for handling IT security incidents	INFOSEC 101 or 103 INFOSEC 205
INFOSEC 307 Advanced	How to Conduct a Risk Analysis/ Assessment	Individuals responsible for conducting risk analyses	INFOSEC 101 or 103 INFOSEC 205

INFOSEC 104: Fundamentals of IT Security

Brief Description

This course is designed for employees directly involved with protecting the IT system. It provides a basic understanding of the federal laws and agency-specific policies and procedures, the vulnerabilities and threats to IT systems, the countermeasures that can help to mitigate the threats, and an introduction to the physical, personnel, administrative, and system/technical controls.

Intended Audience

The course is for employees who need more than just the basics of IT security. It is an introductory course that can be used as a prerequisite for higher-level material. This could include System Administrators, System Staff, Information Officers, Information System Security Officers, Security Officers, and Program Managers.

Note: This course can be taken in place of the INFOSEC 101 course. It is designed as an introductory course for those employees who have job responsibilities directly related to securing the IT system.

INFOSEC 201: Developing the IT System Security Plan

Brief Description

By law, every IT federal system must have an IT system security plan for its general support systems and major applications. This course explains how to develop an IT System Security Plan following the guidelines set forth in NIST SP 800-18 "Guide for Developing Security Plans for Information Technology Systems."

Intended Audience

The system owner (or team) responsible for ensuring that the IT system security plan is prepared and implemented. In many agencies, the IT system security plan will be developed by a team, such as the System Administrator, Information Officer, Security Officer, and the Information System Security Officer.

List of Topics

System Identification; Assignment of Security Responsibilities; System Description/Purpose; System Interconnection; Sensitivity and Sharing of Information; Risk Assessment and Management; Administrative, Physical, Personnel, and System/Technical Controls; Life Cycle Planning; and Security Awareness and Training.

Note: The design of this course should be customized with an agency-approved methodology and a predefined set of templates on how to develop an IT system security plan. The students should leave the class with the agency-approved tools necessary to develop the plan.

INFOSEC 202: How to Develop an IT System Contingency Plan

Brief Description

The hazards facing IT systems demand that effective business continuity plans and disaster-recovery plans be in place. Business continuity plans define how to recover from disruptions and continue support for critical functions. Disaster recovery plans define how to recover from a disaster and restore critical functions to normal operations. The first step is to define one's agency's critical functions and processes, and determine the recovery timeframes and trade-offs. This course discusses how to conduct an in-depth Business Impact Analysis (BIA) (identifying the critical business functions within an agency and determining the impact of not performing the functions beyond the maximum acceptable outage) that defines recovery priorities, processing interdependencies, and the basic technology infrastructure required for recovery.

Intended Audience

This course is for those employees responsible for the planning and management of the IT system. This may include the System Administrator, Information Officer, Security Officer, and Information System Security Officer.

List of Topics

What Is an IT System Contingency Plan; Conducting a Business Impact Analysis (BIA); Setting Your Site (hot site, cold site, warm site); Recovery Objectives; Recovery Requirements; Recovery Implementation; Backup Options and Plans; Testing the Plan; and Evaluating the Results of Recovery Tests.

Note: The content of this course should be customized with an agency-approved methodology for creating an IT system contingency plan. If possible, preapproved templates or tools should be included.

INFOSEC 203: System/Technical Responsibilities for Protecting the IT System

Brief Description

This course begins by explaining the vulnerabilities of and threats to the IT system and what is necessary to protect the physical assets and information. It focuses on specific requirements such as protecting the physical environment, installing software, access controls, configuring operating systems and applications to meet security requirements, and understanding audit logs.

Intended Audience

This course is intended for those employees who are involved in and responsible for the planning and day-to-day operations of the IT system. This would include System Administrators, System Staff, Information Officers, and Information System Security Officers.

List of Topics

Overview of IT System Security; Identifying Vulnerabilities, Threats, and Sensitivity of the IT System; Identifying Effective Countermeasures; Administrative Responsibilities (e.g., management of logs and records); Physical Responsibilities (e.g., server room security); Interconnection Security; Access Controls (identification and authentication); Group and File Management (setting up working groups and shared files); Group and File Permissions (configuring the system for access permissions); Audit Events and Logs; and IT Security Maintenance.

INFOSEC 204: Life Cycle Planning for IT System Security

Brief Description

The system life cycle is a model for building and operating an IT system from its inception to its termination. This course covers the fundamentals of how to identify the vulnerabilities of and threats to IT systems before they are implemented and how to plan for IT security during the acquisition and design of an IT system. This includes identifying the risks that may occur during implementation of the IT system and how to minimize those risks, describing the standard operating procedures with a focus on security, how to test that an IT system is secure, and how to dispose of terminated assets.

Intended Audience

This course is designed for managers tasked with the acquisition and design of IT systems. This could include Contracting Officers, Information Officers, System Administrators, Program Managers, and Information System Security Officers.

List of Topics

Identify IT Security Needs during the Design Process; Develop IT Security in the Acquisition Process; Federal Laws and Regulations; Agency Policies and Procedures; Acquisition, Development, Installation, and Implementation Controls; Risk Management; Establishing Standard Operating Procedures; and Destruction and Disposal of Equipment and Media.

Note: The course focus should be on the implementation and use of organizational structures and processes for IT security and related decision-making activities. Agency-specific policies, guidelines, requirements, roles, responsibilities, and resource allocations should be previously established.

INFOSEC 205: Basic Information System Security Officer (ISSO) Training

Brief Description

This course provides an introduction to the ISSO role and responsibilities. The ISSO implements the IT system security plan and provides security oversight on the IT system. The focus of the course is on understanding the importance of IT security and how to provide a security management role in the daily operations.

Intended Audience

This course is for employees assigned as the ISSO or equivalent. This could be System Administrators, Information Officers, Program Managers, or Security Officers.

List of Topics

Overview of IT Security; Vulnerabilities, Threats, and Sensitivity; Effective Countermeasures; Administrative Controls; Physical Controls; Personnel Controls; System/Technical Controls; Incident Handling; and Security Awareness Training.

Note: Each agency should have someone designated as the Information System Security Officer (ISSO) who is responsible for providing security oversight on the IT system.

INFOSEC 206: Certifying and Accrediting the IT System

Brief Description

This course provides information on how to verify that an IT system complies with information security requirements. This includes granting final approval to operate an IT system in a specified security mode and ensure that classified or sensitive but unclassified (SBU) information is protected according to federal and agency requirements.

Intended Audience

This course is for individuals assigned the Designated Approving Authority (DAA) role and responsibilities. This includes Program Managers, Security Officers, Information Officers, or Information System Security Officers.

List of Topics

Federal Laws and Regulations; Agency Policies and Procedures; Understanding Vulnerabilities, Threats, and Sensitivities; Effective Countermeasures; Access Controls; Groups and File Permissions; Protection of Classified and SBU Information; Protection of TEMPEST and Other Equipment; The Accreditation Process; Incident Handling; Life Cycle Management; Standard Operating Procedures; and Risk Management.

INFOSEC 207: Information System Security for Executive Managers

Brief Description

This course provides an overview of the information system security concerns for executive-level managers. It emphasizes the need for both planning and managing security on the IT system, how to allocate employee and financial resources, and how to lead the IT security team by example.

Intended Audience

This course is for executive-level managers.

List of Topics

Overview of IT System Security; Federal Laws and Regulations; Vulnerabilities and Threats to the IT System; Effective Countermeasures; Need for IT Security Management and Oversight; and Budgeting for IT Security.

Note: This course content should be customized for each agency to make sure it meets the specific needs of the executive-level management team. It is anticipated that this would be several short, interactive sessions based on specific topics. Some sessions could be delivered via a technology-based application to effectively plan for time limitations.

INFOSEC 208: An Introduction to Network and Internet Security

Brief Description

In this course, the focus is on how develop a network and Internet/intranet security policy to protect the agency's IT system assets and information. The focus is on how to analyze the vulnerabilities of the IT system and review the various external threats, how to manage the risks and protect the IT system from unauthorized access, and how to reduce one's risks by deploying technical countermeasures such as firewalls and data encryption devices.

Intended Audience

This course is for employees involved with the implementation, day-to-day management, and oversight responsibilities of the network connections, including internal intranet and external Internet connections. This could include System Administrators, System Staff, Information Officers, Information System Security Officers, Security Officers, and Program Managers.

List of Topics

Overview of IT Network Security and the Internet; Introduction to TCP/IP and Packets; Understanding Vulnerabilities and Threats to Network Connections (hackers, malicious codes, spoofing, sniffing, denial-of-service attacks, etc.); Effective Countermeasures for Network Connections (policies, access controls, physical protections, anti-virus software, firewalls, data encryption, etc.); Developing a Network and Internet/intranet Security Policy; and How to Recognize an Internet Attack.

INFOSEC 209 An Introduction to Cryptography

Brief Description

The focus of this course is to provide an overview of cryptography. This includes the basic concepts of cryptography, public and private key algorithms in terms of their applications and uses, key distribution and management, the use of digital signatures to provide authenticity of electronic transactions, and non-repudiation.

Intended Audience

This course is for employees involved with the management and security responsibilities of the network connections. This could include System Administrators, System Staff, Information Officers, Information System Security Officers, Security Officers, and Program Managers.

List of Topics

Cryptography Concepts; Authentication Methods Using Cryptographic Modules; Encryption; Overview of Certification Authority; Digital Signatures; Non-repudiation; Hash Functions and Message Digests; Private Key and Public Key Cryptography; and Key Management.

INFOSEC 301: Understanding Audit Logs

Brief Description

This is an interactive class focusing on how to understand and review audit logs. It explains what types of events are captured in an audit log, how to search for unusual events, how to use audit log tools, how to record and store audit logs, and how to handle an unusual audit event.

Intended Audience

This course is for employees assigned to manage and provide oversight of the daily IT system operations. This includes System Administrators, Information Officers, and Information System Security Officers.

List of Topics

Understanding an IT System Event, Planning for Audit Log Reviews; How to Review Audit Logs; How to Find and Search Through Audit Logs; Using Third-Party Tools for Audit Log Reviewing; How to Handle an Unusual System Event in the Audit Log.

Note: As a prerequisite, students should have completed either INFOSEC 203 or INFOSEC 205 so that they have a basic understanding of IT security concepts.

INFOSEC 302: Windows NT 4.0 Server and Workstation Security

Brief Description

This course focuses on how to properly configure the Windows NT 4.0 security features for both the server and workstation operating systems. Students learn the security features of Windows NT and participate in installing and configuring the operating systems in a hands-on computer lab.

Intended Audience

This course is designed for employees who are responsible for installing, configuring, and managing networks using the Windows NT 4.0 server and workstation operating system. This may include Information Officers, System Administrators, and System Staff.

List of Topics

Overview of the Windows NT 4.0 Server and Workstation Operating Systems; Identification and Authentication Controls; Discretionary Access Controls; Group Organization and Permissions; Directory and File Organization and Permissions; Protecting System Files; Auditing Events; Using the Windows NT Tools to Configure and Maintain the System.

Note: As a prerequisite, students should complete INFOSEC 203 so they have a basic understanding of IT security concepts.

INFOSEC 303: Windows 2000 Security

Brief Description

This course is similar to INFOSEC 302 except that it focuses on how to properly configure the security features of the Windows 2000 operating system. Students learn the security features of Windows 2000 by installing and configuring the operating system in a hands-on computer lab.

Intended Audience

This course is designed for employees who are responsible for installing, configuring, and managing networks using the Windows 2000 operating system. This may include Information Officers, System Administrators, and System Staff.

List of Topics

Overview of the Windows 2000 Operating System; The Domain Name System (DNS); Migrating Windows NT 4.0 Domains; Identification and Authentication Controls; Discretionary Access Controls; File System Resources (NTFS); Group Organization and Permissions; Directory and File Organization and Permissions; Protecting System Files; Auditing Events; Using the Windows 2000 Tools to Configure and Maintain the System.

Note: As a prerequisite, students should complete INFOSEC 203 so they have a basic understanding of IT security concepts.

INFOSEC 304: UNIX Security

Brief Description

In this hands-on course, students will gain the knowledge and skills needed to implement security on the UNIX operating system. This includes securing the system from internal and external threats, protecting the UNIX file system, controlling superuser access, and configuring tools and utilities to minimize vulnerabilities and detect intruders.

Intended Audience

This course is designed for employees who are responsible for installing, configuring, and managing networks using the UNIX operating system. This may include Information Officers, System Administrators, and System Staff.

List of Topics

Introduction to UNIX Security; Establishing Secure Accounts; Storing Account Information; Controlling Root Access; Directory and File Permissions; Minimize Risks from Unauthorized Programs; and Understanding TCP/IP and Security.

Note: As a prerequisite, students should complete INFOSEC 203 so that they have a basic understanding of IT security concepts.

INFOSEC 305: Advanced ISSO Training

Brief Description

This course provides an in-depth look at ISSO responsibilities. The focus is on how to review security plans, contingency plans/disaster recover plans, and IT system accreditation; how to handle IT system incidents; and how specific IT security case studies are examined and evaluated.

Intended Audience

This cource is intended for ISSOs who have completed INFOSEC 205 and have at least one year of experience as the ISSO.

List of Topics

Oversight Responsibilities for Reviewing IT System Security Plans and Contingency Plans; How to Handle IT System Incidents; and Case Studies.

INFOSEC 306: Incident Handling

Brief Description

This course explains the procedures for handling an IT system security incident. It begins by defining how to categorize incidents according to risk, followed by how to initiate and conduct an investigation and who to contact for support. Key to handling incidents is ensuring that equipment and information is not compromised during an investigation. Thus, students learn the proper procedures for safekeeping assets and information.

Intended Audience

This course is designed for employees who are responsible for handling IT security incidents. This could include Information Officers, Information System Security Officers, Security Officers, and individuals representing a computer incident response team.

List of Topics

Understanding an IT System Security Incident; Federal Laws and Civil/Criminal Penalties; Agency Policies and Penalties; The Agency-Specific Security Incident Reporting Process; Security Investigation Procedures; Identify Investigative Authorities; Interfacing with Law Enforcement Agencies; Witness Interviewing; Protecting the Evidence; and How to Write an IT System Security Incident Report.

Note: As a prerequisite, students should complete INFOSEC 205 so that they have a basic understanding of IT security concepts.

INFOSEC 307: How to Conduct a Risk Analysis/Assessment

Brief Description

This course explains the process of conducting a risk analysis/assessment. It reviews why a risk analysis is important, the objectives of a risk analysis, when the best time is to conduct a risk analysis, the different

methodologies to conduct a risk assessment (including a review of electronic tools), and provides plenty of hands-on opportunities to complete a sample risk analysis. A critical element of a risk analysis/assessment is considering the target analysis and target assessment. The unauthorized intruder may also be conducting an analysis of the information system risks and will know the vulnerabilities to attack.

Intended Audience

This course is for individuals tasked with completing a risk analysis. This could include the Information Officer, System Administrator, Program Manager, Information System Security Officer, and Security Officer.

List of Topics

Overview of a Risk Analysis; Understanding Vulnerabilities, Threats, and Sensitivity and Effective Countermeasures of IT Systems; Objectives of a Risk Analysis; Risk Analysis Methodologies; Federal Guidance on Conducting a Risk Analysis; Process of Conducting a Risk Analysis; Electronic Risk Analysis Tools; Completing Sample Risk Analysis Worksheets (asset valuations, threat, and vulnerability evaluation; level of risk; and countermeasures); and Reviewing Target Analysis/Assessments.

Note: This course may be offered in conjunction with INFOSEC 201 and INFOSEC 206.

85

Maintaining Information Security during Downsizing

Thomas J. Bray, CISSP

Today, companies of every size are relying on Internet and other network connections to support their business. For each of those businesses, information and network security have become increasingly important. Yet, achieving a security level that will adequately protect a business is a difficult task because information security is a multifaceted undertaking. A successful information security program is a continuous improvement project involving people, processes, and technology, all working in unison.

Companies are especially vulnerable to security breaches when significant changes occur, such as a reduction in workforce. Mischievous individuals and thieves thrive on chaos. Companies need even more diligence in their security effort when executing a reduction in workforce initiative. Security is an essential element of the downsizing effort.

Even in Good Times

In good times, organizations quickly and easily supply new employees with access to the computer and network systems they need to perform their jobs. A new employee is a valuable asset that must be made productive as soon as possible. Computer and network administrators are under pressure to create accounts quickly for the new hires. In many instances, employees may have more access than they truly need. The justification for this, however misguided, is that "it speeds up the process."

When an employee leaves the company, especially when the departure occurs on good terms, server and network administrators tend to proceed more slowly. Unfortunately, the same lack of urgency exists when an employee departure is not on good terms or a reduction in the workforce occurs.

Disgruntled Employees

Preparing for the backlash of a disgruntled employee is vital during an employee layoff. Horror stories already exist, including one about an ex-employee who triggered computer viruses that resulted in the deletion of sales commission records. In another company, an ex-employee used his dial-up access to the company network to copy a propriety software program worth millions of dollars. An article in *Business Week* sounded an alarm of concern.[1]

The biggest threat to a company's information assets can be the trusted insiders. This is one of the first concepts learned by information security professionals, a concept substantiated on several occasions by surveys conducted by the Computer Security Institute (CSI) and the Federal Bureau of Investigation (FBI).

The market research firm Digital Research conducted a survey for security software developer Camelot and *eWeek* magazine. They found that, "Insiders pose the greatest computer security threat. Disgruntled insiders

and accounts held by former employees are a greater computer security threat to U.S. companies than outside hackers." Out of 548 survey respondents, 43 percent indicated that security breaches were caused by user accounts being left open after employees had left the company.[2]

Yeah, Right. What Are the Cases?

In many cases of ex-employees doing harm to their former employers, the extent of the problem is difficult to quantify. Some companies do not initially detect many of the incidents, and others prefer to handle the incidents outside the legal system. A small percentage of incidents have gone through the legal system and, in some cases, the laws were upheld. Each time this occurs, it strengthens support for the implementation of information security best practices. Although many states have computer crime laws, there is still only a small percentage of case law.

Example Incident: *The Boston Globe,* by Stephanie Stoughton, Globe Staff, 6/19/2001[3]

> Ex-tech worker gets jail term in hacking. A New Hampshire man who broke into his former employer's computer network, deleted hundreds of files, and shipped fake e-mails to clients was sentenced yesterday to six months in federal prison. U.S. District Judge Joseph DiClerico also ordered Patrick McKenna, 28, to pay $13,614.11 in restitution to Bricsnet's offices in Portsmouth, N.H. Following McKenna's release from prison, he will be under supervision for two years.

High-Tech Measures

E-Mail

E-mail is one of the most powerful business tools in use today. It can also be a source of communications abuse and information leakage during a downsizing effort. The retention or destruction of stored e-mail messages of ex-employees must also be considered.

Abuse

Do not allow former employees to keep e-mail or remote access privileges in an attempt to ease the pain of losing their jobs or help in their job searches. The exposure here is the possibility of misrepresentation and inappropriate or damaging messages being received by employees, clients, or business partners. If the company wants to provide e-mail as a courtesy service to exiting employees, the company should use a third party to provide these services. Using a third party will prevent employees from using existing group lists and addresses from their address books, thus limiting the number of recipients of their messages.

Employees who know they are to be terminated typically use e-mail to move documents outside the organization. The company's termination strategy should include a method for minimizing the impact of confidential information escaping via the e-mail system. E-mail content filters and file-size limitations can help mitigate the volume of knowledge and intellectual capital that leaves the organization via e-mail.

Leakage

E-mail groups are very effective when periodic communication to a specific team is needed. The management of the e-mail group lists is a job that requires diligence. If ex-employees remain on e-mail group lists, they will continue to receive company insider information. This is another reason the company should not let former employees keep company e-mail accounts active as a courtesy service.

Storage

E-mail messages of ex-employees are stored on the desktop system and the backup disk or tapes of the e-mail server. The disposal of these documents should follow the company's procedure for e-mail document retention.

In the absence of an e-mail document retention policy, the downsizing team should develop a process for determining which e-mail messages and attachments will be retained and which will be destroyed.

Low-Tech Measures

The fact that information security is largely a people issue is demonstrated during a reduction in force initiative. It is the business people working hand in hand with the people staffing the technical and physical security controls who will ensure that the company is less vulnerable to security breaches during this very disruptive time in the company.

Document Destruction

As people exit the company during a downsizing effort, mounds of paper will be thrown in the trash or placed in the recycling bin. Ensuring that confidential paper documents are properly disposed of is important in reducing information leaks to unwanted sources.

After one company's downsizing effort, I combed through their trash and recycling bins. During this exercise, I found in the trash several copies of the internal company memo from the CEO that explained the downsizing plan. The document was labeled *"Company Confidential — Not for Distribution Outside of the Company."* This document would have been valuable to the news media or a competitor.

All companies have documents that are confidential to the business; however, most companies do not have a document classification policy. Such a policy would define the classification designations, such as:

- Internal Use Only
- Confidential
- Customer Confidential
- Highly Restricted

Each of these classifications has corresponding handling instructions defining the care to be taken when storing or routing the documents. Such handling instructions would include destroying documents by shredding them when they are no longer needed.

Many organizations have also been entrusted with confidential documents of business partners and suppliers. The company has a custodial responsibility for these third-party documents. Sorting through paper documents that are confidential to the company or business partners and seeing that they are properly destroyed is essential to the information protection objective.

Security Awareness

Security awareness is a training effort designed to raise the security consciousness of employees (see Exhibit 85.1). The employees who remain with the organization after the downsizing effort must be persuaded to rally around the company's security goals and heightened security posture. Providing the remaining team of employees with the knowledge required to protect the company's vital information assets is paramount. Employees should leave the security training with a mission to be security-aware as they perform their daily work. Some of the topics to be covered in the security awareness sessions include:

- Recognizing social engineering scenarios
- Speaking with the press
- Keeping computer and network access credentials, such as passwords, confidential
- Changing keys and combinations
- Encouraging system administrators and security administrators to be vigilant when reviewing system and security logs for suspicious activity
- Combining heightened computer and network security alertness with heightened physical security alertness

EXHIBIT 85.1 Checklist of Security Actions during Reduction in Workforce Effort

General

- Assemble a team to define the process for eliminating all computer and network access of downsized employees. The team should include representation from Human Resources, Legal, Audit, and Information Security.
- Ensure that the process requires managers to notify the employees responsible for Information Security and the Human Resources department at the same time.
- Educate remaining employees about Information Security company policy or best practices.
- Change passwords of all employees, especially employees with security administrative privileges.
- Check the computer and laptop inventory list and ensure that downsized employees return all computer equipment that was issued to them as employees.
- Be current with your software licenses — ex-employees have been known to report companies to the Software Piracy Association.

Senior Managers

- Explain the need for the downsizing.
- Persuade key personnel that they are vital to the business.
- Resist the temptation to allow downsized officers, senior managers, or any employees to keep e-mail and remote access privileges to ease the pain or help in their job search. If the company wants to provide courtesy services to exiting employees, the company should use a third party to provide these services, not the company's resources.

Server Administrators, Network Administrators, and Security Administrators

- Identify all instances of employee access:
  - — Scan access control systems for IDs or accounts of downsized employees.
  - — Scan remote access systems for IDs or accounts of downsized employees.
  - — Call business partners and vendors for employee authorizations.
- Consult with departing employee management:
  - — Determine who will take on the exiting employee's access.
  - — Determine who will take control of exiting employee's files.

E-mail System Administrators

- Identify all instances of employee access:
  - — Scan the e-mail systems for IDs or accounts of downsized employees.
- Forward inbound e-mail messages sent to an ex-employees' e-mail account to their manager.
- Create a professional process for responding to individuals who have sent e-mails to ex-employees, with special emphasis on the mail messages from customers requiring special care.
- Remove ex-employees from e-mail group lists.

Managers of Exiting Employees

- Determine who will take on the access for the exiting employees.
- Determine who will take control of exiting employee computer files.
- Sort through exiting employee paper files for documents that are confidential or sensitive to the business.

Prepare for the Worst

- Develop a list of likely worst-case scenarios.
- Develop actions that will be taken when worst-case scenarios occur.

Conclusion

Information security involves people, processes, and technical controls. Information security requires attention to detail and vigilance because it is a continuous improvement project. This becomes especially important when companies embark on a downsizing project.

Companies should always be mindful that achieving 100 percent security is impossible. Mitigating risk to levels that are acceptable to the business is the most effective methodology for protecting the company's information assets and the network systems.

Businesses need to involve all employees in the security effort to have an effective security program. Security is most effective when it is integrated into the company culture. This is why security awareness training is so important.

Technology plays a crucial role in security once the policies and processes have been defined to ensure that people properly manage the technological controls being deployed. A poorly configured firewall provides a false sense of security. This is why proper management of security technologies provides for a better information protection program.

Notes

1. http://www.businessweek.com/bwdaily/dnflash/jun2001/nf20010626_024.htm.
2. http://www.usatoday.com/life/cyber/tech/2001-06-20-insider-hacker-threat.htm
   http://www.zdnet.com/zdnn/stories/news/0,4586,2777325,00.html.
   http://www.cnn.com/2001/TECH/Internet/06/20/ security.reut/index.html.
3. http://www.boston.com/dailyglobe2/170/business/
   Ex_tech_worker_gets_jail_term _in_ hacking+.shtml.

86

The Business Case for Information Security: Selling Management on the Protection of Vital Secrets and Products

Sanford Sherizen, Ph.D., CISSP

If the world was rational and individuals as well as organizations always operated on that basis, this chapter would not have to be written**.** After all, who can argue with the need for protecting vital secrets and products? Why would senior managers not understand the need for spending adequate funds and other resources to protect their own bottom line? Why not secure information as it flows throughout the corporation and sometimes around the world?

Unfortunately, rationality is not something that one can safely assume when it comes to the field of information security. Therefore, this chapter is not only required, but it needs to be presented as a bilingual document, that is, written in a way that reveals strategies by which senior managers as well as information security professionals can maximize their specific interests.

This chapter is based on over 20 years of experience in the field of information security, with a special concentration on consulting with senior- and middle-level managers. The suggestions are based on successful projects and, if followed, can help other information security professionals achieve successful results with their management.

The State of Information Security

Improving information security for an organization is a bit like an individual deciding to lose weight, to exercise, or to stopping smoking. Great expectations. Public declarations of good intentions. A projected starting date in the near future. And then the realization that this is a constant activity, never to end and never to be resolved without effort.

Why is it that there are so many computer crime and abuse problems at the same time that an increasing number of senior executives are declaring that information security is an absolute requirement in their organizations? This question is especially perplexing when one considers the great strides that have been made in the field of information security in allowing greater protection of assets. While the skill levels of the perpetrators have increased and the complexity of technology today leaves many exposures, one of the central issues for today's information security professional is nontechnical in nature. More and more, a challenge that

many in the field face is how to inform, convince, influence, or in some other way "sell" their senior management on the need for improving information security practices.

This chapter looks at the information security–senior executive dialogue, offering the reasons why such exchanges often do not work well and suggesting ways to make this a successful discussion.

Senior Management Views of Information Security

Information security practitioners need to understand two basic issues regarding their senior management. The first is that computer crime is only one of the many more immediate risks that executives face today. The second is that thinking and speaking in managerial terms is a key to even gaining their attention in order to present a business case for improvements.

To the average senior executive, information security may seem relatively easy — simply do not allow anyone who should not see certain information to see that information. Use the computer as a lock against those who would misuse their computer use. Use all of that money that has been given for information technology to come up with the entirely safe computer. Stop talking about risks and vulnerabilities and solve the problem. In other words, information security may be so complex that only simple answers can be applied from the non-practitioner's level.

Among all the risks that a manager must respond to, computer crime seems to fall into the sky-is-falling category. The lack of major problems with the Y2K issue has raised questions in some managerial and other circles as to whether the entire crisis was manufactured by the media and technical companies. Even given the extensive media coverage of major incidents, such as the Yahoo, etc. distributed denial-of-service attack, the attention of managers is quickly diverted as they move on to other, "more important issues." To managers, who are faced with making the expected profits for each quarter, information security is a maybe type of event. Even when computer crime happens in a particular organization, managers are given few risk figures that can indicate how much improvement in information security (X) will lead to how much prevention of crime (Y).

With certain notable exceptions, there are fundamental differences and perceptions between information security practitioners and senior executives. For example, how can information security professionals provide the type of cost-justification or return-on-investment (ROI) figures given the current limited types of tools? A risk analysis or similar approach to estimating risks, vulnerabilities, exposures, countermeasures, etc. is just not sufficient to convince a senior manager to accept large allocations of resources.

The most fundamental difference, however, is that senior executives now are the Chief Information Security Manager (or Chief Corporate Cop) of their organizations. What that quite literally means is that the executives — rather than the information security manager or the IS manager — now have legal and fiduciary responsibilities to provide adequate resources and support for information protection.

Liabilities are now a given fact of life for senior executives. Of particular importance, among the extensive variety of liability situations found in an advanced economy, is the adequacy of information protection. The adequacy of managerial response to information security challenges can be legally measured in terms of due care, due diligence, and similar measures that indicate what would be considered as a sufficient effort to protect their organization's informational assets. Unfortunately, as discussed, senior executives often do not know that they have this responsibility, or are unwilling to take the necessary steps to meet this responsibility. The responsibility for information security is owned by senior management, whether they want it or not and whether they understand its importance or not.

Information Security Views of Senior Management

Just as there are misperceptions of information security, so information security practitioners often suffer from their misperceptions of management. At times, it is as if there are two quite different and quite unconnected views of the world.

In a study done several years ago, CEOs were asked how important information security was to their organization and whether they provided what they felt was adequate assistance to that activity. The results showed an overwhelming vote for the importance of information security as well as the majority of these executives providing sufficient resources. However, when the IS, audit, and information security managers were asked about their executives' views of security, they indicated that there was a large gap between rhetoric

and reality. Information security was often mentioned, but the resources provided and the support given to information security programs often fell below necessary levels.

One of the often-stated laments of information security practitioners is how difficult it is to be truly heard by their executives. Information security can only work when senior management supports it, and that support can only occur when they can be convinced of the importance of information protection. Such support is required because, by the nature of its work, information security is a political activity that crosses departmental lines, chains of command, and even national boundaries.

Information security professionals must become more managerial in outlook, speech, and perspectives. What that means is that it is no longer sufficient to stress the technical aspects of information protection. Rather, the stress needs to be placed on how the information security function protects senior executives from major legal and public relations liabilities. Further, information security is an essential aspect of managing organizations today. Just as information is a strategic asset, so information protection is a strategic requirement. In essence, information security provides many contributions to an organization. The case to be made to management is the business case for information security.

The Many Positive Roles of Information Security

While people may realize that they play many roles in their work, it is worthwhile listing which of those roles apply to "selling information security." This discussion allows the information security practitioner to determine which of the work-related activities that he or she is involved in has implications for convincing senior management of the importance of that work and the need for senior management to provide sufficient resources in order to maximize the protection span of control.

One of the most important roles to learn is how to become an information security "marketeer." Marketing, selling, and translating technical, business, and legal concepts into "managerialese" is a necessary skill for the field of information security today. What are you marketing or selling? You are clarifying for management that not only do you provide information protection but, at the same time, also provide such other valuable services as:

1. *Compliance enforcer and advisor.* As IT has grown in importance, so have the legalities that have to be met in order to be in compliance with laws and regulations. Legal considerations are ever-present today. This could include the discovery of a department using unauthorized copies of programs; internal employee theft that becomes public knowledge and creates opportunity for shareholder suits; a penetration from the outside that is used as a launching pad to attack other organizations, thus creating the possibility of a downstream liability issue; or any of the myriad ways that organizations get into legal problems.
   - — **Benefit to management.** A major role of the information security professional is to assist management in making sure that the organization is in compliance with the law.
2. *Business enabler and company differentiator.* E-commerce has changed the entire nature of how organizations offer goods and services. The business enabler role of information security is to provide an organization with information security as a value-added way of providing ease of purchase as well as security and privacy of customer activities. Security has rapidly become the way by which organizations can provide customers with safe purchasing while offering the many advantages of E-commerce.
   - — **Benefit to management.** Security becomes a way of differentiating organizations in a commercial setting by providing "free safety" in addition to the particular goods and services offered by other corporations. "Free safety" offers additional means of customer satisfaction, encouraging the perception of secure Web-based activities.
3. *Total quality management contributor.* Quality of products and services is related to information security in a quite direct fashion. The confidentiality, integrity, and availability of information that one seeks to provide allow an organization to provide customer service that is protected, personal, and convenient.
   - — **Benefit to management.** By combining proper controls over processes, machines, and personnel, an organization is able to meet the often contradictory requirements of production as well as protection. Information security makes E-commerce possible, particularly in terms of the perceptions of customers that such purchasing is safe and reliable.
4. *"Peopleware" controller.* Peopleware is not the hardware or software of IT. It involves the human elements of the human-machine interface. Information security as well as the audit function serve as

key functions in controlling the unauthorized behavior of people. Employees, customers, and clients need to be controlled in their use of technology and information. The need-to-know and separation-of-duties concepts become of particular importance in the complex world of E-commerce. Peopleware are the elements of the control structure that allow certain access and usage as well as disallow what have been defined as unauthorized activities.

— **Benefit to management.** Managerial policies are translated into information security policies, programs, and practices. Authorized usage is structured, unauthorized usage is detected, and a variety of access control and similar measures offer protections over sensitive informational assets.

The many roles of information security are of clear benefit to commercial and governmental institutions. Yet, these critical contributions to managing complex technical environments tend not to be considered when managers view the need for information security. As a result, one of the most important roles of information security practitioners is to translate these contributions into a business case for the protection of vital information.

Making the Business Case for Information Security

While there are many different ways to make the business case and many ways to "sell" information security, the emphasis of this section is on the Common Body of Knowledge (CBK) and similar sources of explication or desired results. These are a highly important source of professional knowledge that can assist in informing senior executives regarding the importance of information security.

CBK, as well as other standards and requirements (such as the Common Criteria and the British Standards 7799), are milestones in the growth of the professional field of information security. These compendia of the best ways to evaluate security professionals as well as the adequacy of their organizations serve many purposes in working with senior management.

They offer information security professionals the ability to objectively recommend recognized outside templates for security improvements to their own organizations. These external bodies contain expert opinion and user feedback regarding information protection. Because they are international in scope, they offer a multinational company the ability to provide a multinational overview of security.

Further, these enunciations of information security serve as a means of measuring the adequacy of an organization's information security program and efforts. In reality, they serve as an indication of "good practices" and "state of knowledge" needed in today's IT environments. They also provide legal authorities with ways to measure or evaluate what are considered as appropriate, necessary, or useful for organizations in protecting information. A "good-faith effort" to secure information, a term used in the U.S. Federal Sentencing Guidelines, becomes an essential legal indicator of an organization's level of effort, concern, and adequacy of security programs. Being measured against these standards and being found lax may cost an organization millions of dollars in penalties as well as other serious personal and organizational punishments. (For further information on the U.S. Sentencing Guidelines as they relate to information security, see the author's publication on the topic at http://www.computercrimestop.com/.)

Meeting the Information Security Challenge

The many challenges of information security are technical, organizational, political, legal, and physical. For the information security professional, these challenges require new skills and new orientations. To be successful in "selling" information security to senior executives, information security practitioners should consider testing themselves on how well they are approaching these decision makers.

One way to do such a self-evaluation is based on a set of questions used in forensic reviews of computer and other crimes. Investigators are interested in determining whether a particular person has motive, opportunity, and means (MOM). In an interesting twist, this same list of factors can be helpful in determining whether information security practitioners are seeking out the many ways to get the attention of their senior executives.

1. *Motivation.* Determine what motivates executives in their decisions. Understand the key concepts and terms they use. Establish a benefits approach to information security, stressing the advantages of securing

information rather than emphasizing the risks and vulnerabilities. Find out what "marketeering" means in your organization, including what are the best messages, best media, and best communicators needed for this effort.

2. *Opportunity.* Ask what opportunities are available, or can be made, to meet with, be heard by, or gain access to senior executives. Create openings as a means to stress the safe computing message. Opportunities may mean presenting summaries of the current computer crime incidents in memos to management. An opportunity can be created when managers are asked for a statement to be used in user awareness training. Establish an Information Security Task Force, composed of representatives from many units, including management. This could be a useful vehicle for sending information security messages upward. Find out the auditor's perspectives on controls to see how these may reinforce the messages.
3. *Means.* The last factor is means. Create ways to get the message heard by management. Meeting may be direct or indirect. Gather clippings of current computer crime cases, particularly those found in organizations or industries similar to one's own. Do a literature review of leading business, administrative, and industry publications, pulling out articles on computer crime problems and solutions. Work with an organization's attorneys in gathering information on the changing legal requirements around IT and security.

Conclusion

In the "good old days" of information security, security was relatively easy. Only skilled data processing people had the capability to operate in their environment. That, plus physical barriers, limited the type and number of people who could commit computer crimes.

Today's information security picture is far more complicated. The environment requires information security professionals to supplement their technical skills with a variety of "soft skills" such as managing, communicating, and stressing the business reasons for security objectives. The successful information security practitioner will learn these additional skills in order to be heard in the on-rush of challenges facing senior executives.

The technical challenges will certainly not go away. However, it is clear that the roles of information security will increase and the requirements to gain the acceptance of senior management will become more important.

Information Security Management in the Healthcare Industry

Micki Krause

INTRODUCTION

Proper management of the information security program addresses two very important areas: technological, because many of the controls we implement are technical security mechanisms, and people, because security is first and foremost a people issue. However, the information security manager in the healthcare industry is forced to heed another very important area: federal and state regulations.

Recently enacted government legislation, such as the Balanced Budget Act and the Health Insurance Portability and Accountability Act (HIPAA), are adding immense pressure to healthcare organizations, the majority of which have not yet adopted the generally accepted system-security principles common to other regulated industries.

This chapter will address the following issues:

- History of healthcare information systems and the inherent lack of controls
- The challenges the healthcare organization faces, vis à vis its information systems
- The obstacles healthcare companies must overcome in order to implement consumer-centric systems in an environment of consumer distrust of both the healthcare industry and the technology
- The multitude of privacy laws proposed in the last 12 months
- E-commerce and the Internet
- An analysis of the HIPAA security standards

HISTORY OF HEALTHCARE INFORMATION SYSTEMS AND THE INHERENT LACK OF CONTROLS

The goal of today's healthcare organizations' information systems is open, interoperable, standards-compliant, and secure information systems. Unfortunately, this goal does not accurately reflect the state of healthcare's information systems today. We have some very real challenges to understand and overcome.

To begin, the healthcare industry has built information systems without the sufficient granularity required to adequately protect the information for which we are custodians. Many of the existing systems require no more than a three-character log-on ID; some have passwords that are shared by all users; and most have not implemented the appropriate classification of access controls for the jobs that users perform. One healthcare organization realized that their junior claims examiners were authorizing liposuction procedures, which ordinarily are not reimbursed. However, due to a lack of granularity, the junior examiners had the same privileges as the more senior personnel, and thus, the ability to perform inappropriate actions.

Because of this lack of appropriate controls, healthcare companies have recently come to the realization that they will have to invest in retrofitting security in order to be compliant with federal regulations. Not only will they be forced to expend incremental resources in this effort, but they lose the opportunity to utilize those resources for new application development.

Unfortunately, we don't see much of an improvement in many of the commercial product offerings on the market today. Consistently, from operating systems to off-the-shelf applications, too many new products lack sufficient controls. Products from large companies, with wide deployment, such as the Windows NT operating system or the Peoplesoft application, are not built to be compliant with best practices or generally accepted system-security principles. This is poor testimony to the quality of software today. In fact, many security practitioners find it unsettling to get blank stares from their vendor representatives when they ask whether the product has the most basic of controls. Worse yet is the null response security managers receive when they ask the vendor whether or not the manufacturers have a strategy for compliance with federal regulations.

There is no doubt that along with other industries, the healthcare industry must begin to collaborate with product vendors, to ensure that new products are built and implemented by default in a secure manner.

THE CHALLENGES THE HEALTHCARE ORGANIZATION FACES, VIS À VIS ITS INFORMATION SYSTEMS

Another challenge facing organizations today is the pressure of keeping their networked resources open and closed at the same time, a security paradox of doing electronic commerce. Healthcare companies are forced to allow their insecure systems to be accessible to outside constituencies, trading partners, vendors, and members. In these situations, more robust authentication and access controls are mandatory, especially for those users who are not employees of the company. To exacerbate the challenge, the security manager has to reconcile decisions vis à vis the correct balance between access and security, especially with regard to requests for access to internal resources by external trading partners. Questions plaguing the healthcare organization include: "Should an employer have a right to see the patient-identifiable data on their employees?" For example, if a healthcare company is custodian of John Smith's medical records, and John drives a dynamite truck, should the health plan acquiesce to the employer if John's medical records indicate he has epileptic seizures? Should the employer only have this right if the safety of the public is at risk? Should the employer have access only with John's permission? The answers to these dilemmas are not clear today. Thus, health plans struggle with the overriding challenge of maintaining confidentiality of patient information, while providing reasonable access to it. Further, this balance of access and security has to be maintained across a broadly diverse infrastructure of disparate platforms and applications.

Also, there are other business partners that consistently request access to internal resources, e.g., fulfillment houses, marketing organizations, pharmacy companies. Where does it stop? How can it stop — when the competitive imperative for healthcare companies today is providing the ability to connect quickly and meaningfully with business partners and customers to improve the movement and quality of information and services.

Then, of course, there is the new frontier, the Internet, and the challenges that new technologies present. Organizations tread lightly at first, opening up their networks to the Internet by providing the ability for their employees to surf the Web. It wasn't long before they discovered that if an employee using a company computer on company premises downloads pornographic materials, another of their employees could sue the company for sexual harassment. Once the barn door is open, however, it's hard to get the horses back in. Health plans faced increasing demand to accommodate electronic commerce. Surprisingly, the industry that, until very recently, considered sending files on a diskette the definition for electronic data interchange, rapidly found that they were losing membership because employers' benefits administrators were refusing to do business with plans that could not support file transfers over the Internet.

Of course, when the healthcare organization opens its kimono to the Internet, it introduces a multitude of threats to its internal network. Although most organizations implemented perimeter security with the installation of firewalls, business demands forced them to open holes in the defensive device, to allow certain types of inbound and outbound traffic. For example, one health plan encouraged its employees to enroll in courses offered on the Internet which required opening a specific port on the firewall and allowing traffic to and from the university's Internet address. In another instance, a health plan employee needed access to a nonprofit entity's Web site in order to perform Webmaster activities. In order to accomplish this, the employee utilized a service through the Internet, requiring access through the firewall. Thus, the firewall slowly becomes like Swiss cheese, full of holes. Ergo, health plans have the challenge of engaging in business with external partners while *effectively* managing the firewall.

More challenging than managing external connectivity is the security manager's task of hiring security practitioners with the necessary skills and knowledge to effectively manage the firewall. These individuals must have experience managing UNIX systems, since most firewalls are built on a UNIX operating system; must know how the Internet protocols such as file transfer protocol (FTP) work through the firewall; and must have the expertise to monitor network router devices and know how to write rules for those devices, in order to accommodate business requirements while protecting the enterprise. On the other hand, as healthcare organizations seek to outsource networked resources, for example, Web sites and firewalls, the security manager must be able to provide sufficient monitoring and security oversight, to ensure that the outsourcer is meeting its contractual obligations.

It's no wonder that insurance companies are offering a myriad of secure-systems insurance programs. Cigna Insurance, for example, recently developed a program to offer insurance policies of up to $25 million in liability per loss, reflecting the realization that companies are not only more reliant on information systems, but with the introduction of the Internet, the risk is that much greater.

THE OBSTACLES THAT HEALTHCARE COMPANIES MUST OVERCOME IN ORDER TO IMPLEMENT CONSUMER-CENTRIC SYSTEMS IN AN ENVIRONMENT OF CONSUMER DISTRUST OF BOTH THE HEALTHCARE INDUSTRY AND THE TECHNOLOGY

In this competitive industry, the healthcare organization's mandate is to increase customer intimacy while decreasing operational costs; grant external access to internal data and applications, while most existing applications don't have the appropriate controls in place; and secure the new

technologies, especially for third-party access. With all of these issues to resolve, health plans are turning toward Web-based solutions, utilizing public key encryption and digital certificate technologies. But even though health plans have the motivation to move into the Internet mainstream, there are obstacles to overcome that have, for now, slowed the adoption of Web technologies.

First, there are technological weaknesses in the Internet infrastructure. Most organizations have service-level agreements for their internal resources, which guarantee to their employees and customers a certain level of availability and response time. In the Internet space, no one entity is accountable for availability. Also, there are five major electronic junctions where the Internet is extremely vulnerable. When one junction is down, many customers feel the pain of not having reliable service. Since the Internet is not owned or operated by any one person or organization, by its very nature, it cannot be expected to provide the same reliability, availability, and security as a commercial network service provider can. For example, commercial telecommunications companies provide outsourced wide area networks and deploy state of the art communications and security technologies with multiple levels of redundancy and circuitry. The Internet is like a Thomas' English muffin — a maze of nooks and crannies that no one entity controls.

Next, all of the studies show that a large majority of physicians are not comfortable with computers, let alone the Internet. The doctors are ambivalent about adopting information technology, and since there is no control over the content of the information on the net, physicians have been slow to adopt electronic mail communications with their patients on the Internet. They have legitimate concern since there is no positive assurance that we can know exactly who we are communicating with on the Internet. Thus, the healthcare providers distrust the Internet.

They are not the only persons with doubts and concerns. The perception of a lack of security and privacy by consumers is a tremendous challenge for healthcare organizations. Moreover, the media promulgates the paranoia. It's no wonder that consumers are fearful of losing their privacy when publications offer headlines such as "Naked Before the World: Will your Medical Records be safe in a new National Databank?" (*Newsweek* magazine) or "The Death of Privacy: You Have No Secrets." (*Time* magazine).

Therefore, if healthcare organizations are to successfully deploy consumer-intimate Web-based applications, the biggest hurdle they have to overcome is consumer fear, as depicted in the cartoon in Exhibit 17.1.

This consumer fear is not a new phenomenon. For many years, public polls have shown that consumers are increasingly distrustful of organizations that collect their private information. More disconcerting than this,

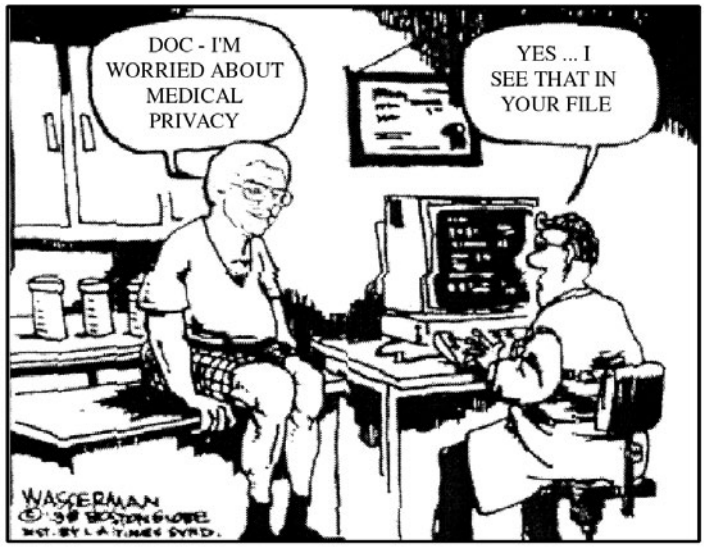


Exhibit 17.1.

from a healthcare perspective, is that this fear is manifesting itself in negative impacts to the quality of their personal health. More and more, consumers are going out of their local areas to obtain healthcare and lying or holding back information from their healthcare providers, primarily to maintain their sense of privacy and maintain some semblance of confidentiality. This reflects a real disconnect between the consumer and the custodians of the consumer data, the health plan and the doctor.

In early 1999, the Consumers Union, the largest consumer advocacy organization in the United States, sponsored a nationwide survey. They sampled 1000 adults in the U.S. and a separate 1000 adults in California. The survey asked people how willing they were to disclose their personal medical information.

In Exhibit 17.2, we can see that the survey found that although people do concede that persons other than their immediate provider require access to their personal medical records, they display a very strong preference for restricting access. Only four of every ten asked were willing to disclose their medical information to health plans. Roughly six in ten would explicitly refuse to grant access to their information to a hospital, even if the hospital were to offer preventive care programs. Also, consumers are not happy having their employers or potential employers view their personal healthcare information. Most are not willing to offer their information to a potential employer who may be considering them for a job. Further, the

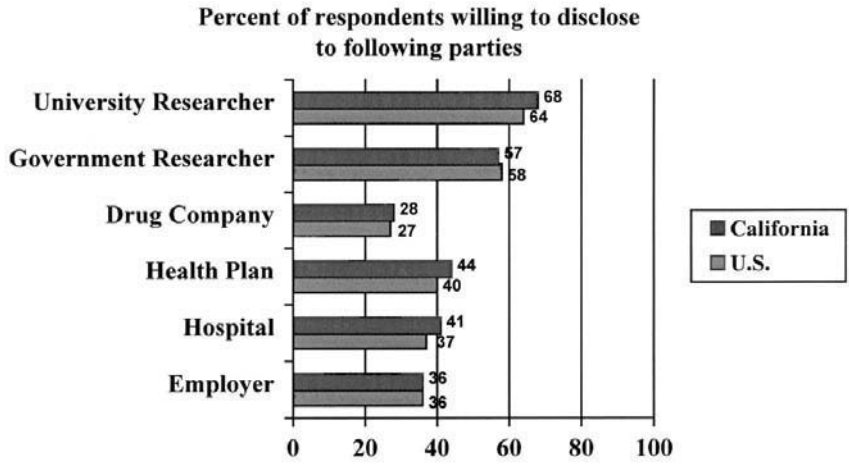


Exhibit 17.2.

drug companies are lowest on the totem pole because Americans do not want their medical data collected for the purposes of marketing new drugs.

In Exhibit 17.3, we see another interesting finding from the survey: most people consider electronic piracy, that is hackers, the biggest threat to their privacy. This is counter to the real threat, which is the disclosure of information by medical personnel, health plans, or other authorized users, but it's not surprising that the average consumer would be very worried about hackers, when we consider how the media exploits attempts by teenagers to hack in to the Pentagon's computers. Moreover, the vendors exacerbate these fears by playing up the evil hacker as they attempt to sell products by instilling fear, uncertainty, and doubt in our hearts and minds.

Exhibit 17.4 shows that most of the survey respondents perceive that if health plans and providers implement security provisions and information security management policies in order to protect medical information, it would make them more inclined to offer their personal information when it was requested. Americans believe that three specific policies should be adopted to safeguard their medical privacy:

1. Impose fines and punishments for violations
2. Require an individual's specific permission to release personal information
3. Establish security systems with security technologies, such as passwords and encryption

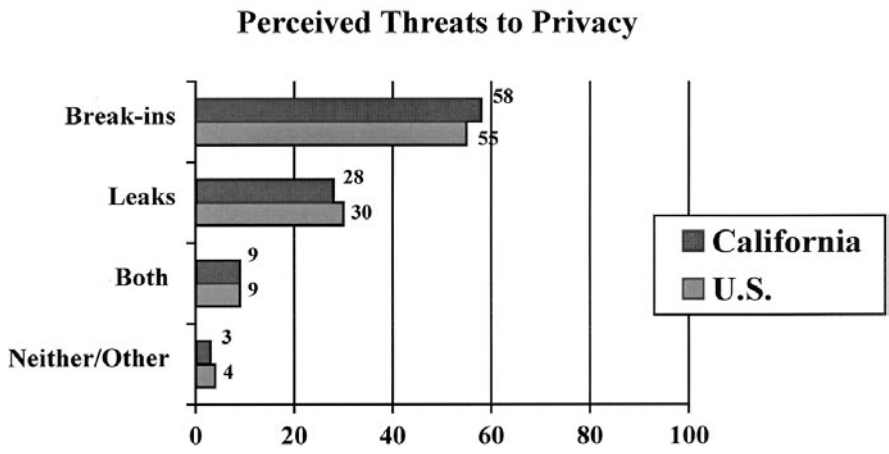


Exhibit 17.3.

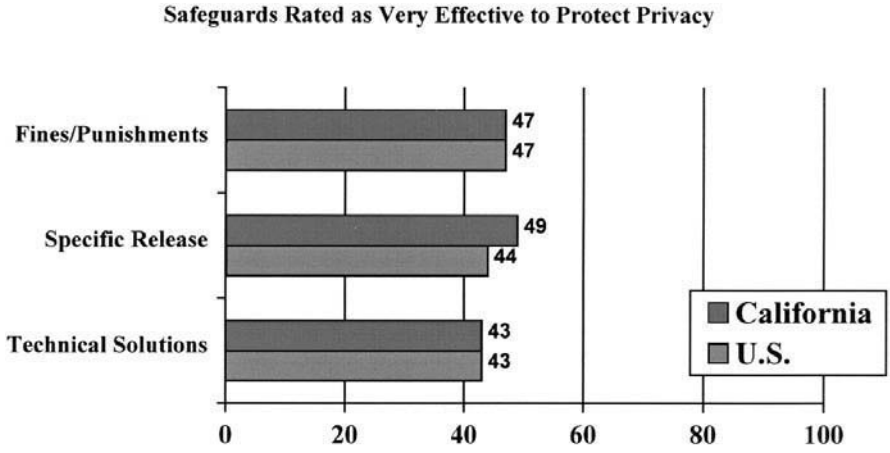


Exhibit 17.4.

Further, the survey respondents were very favorable about sending a health plan's Chief Executive Officer to prison in the event of willful or intentional disclosure of medical information.

The Consumers' Union survey also revealed that consumers are aware — they know that their information is stored in computer databases, and they perceive computerization as the greatest threat to their privacy. In fact, more than one-half of the respondents think that the shift from paper

records to electronic systems makes it *more* difficult to keep personal medical information private and confidential. This should be of interest to any information systems manager, since computerization really provides more of an opportunity to secure data. However, perception *is* reality. Therefore, the lesson from this survey is threefold:

- Consumers do not trust health plans or providers
- Consumers do not trust computers
- Consumers will compromise the quality of their healthcare

all in the name of privacy.

This lesson can be an opportunistic one for the health plan security manager. Healthcare can turn those consumer fears around, and win over the public by showing them that health plans take their obligation for due diligence very seriously, and protecting consumer privacy is in perfect alignment with healthcare organizations' internal values.

Case in point: In December 1998, more people purchased goods on the Internet than ever before. The question is why? Price Coopers, the accounting firm, completed a survey early in 1999 which found that the leading factor that would persuade fearful consumers to log on to the Internet was *an assurance of improved privacy protection.* Healthcare can leverage the capabilities of security to garner that public trust. Privacy is not an arcane or a technical issue. It is, however, a major issue with consumers, and there is heightened urgency around healthcare privacy and security today, more so than ever before.

HISTORY REPEATS ITSELF

In 1972, in a similar environment of public distrust, then Department of Health and Human Services Secretary Elliot Richardson appointed an advisory board to assist the federal government in identifying approaches to protect the privacy of information in an ever-evolving computer age. The board issued a report detailing a code of fair information principles, which became the National Privacy Act of 1974.

The act outlines five separate and distinct practices:

Fair Information Privacy Principles

- "There must be a way … to prevent information about a person that was obtained for one purpose from being used or made available for other purposes without that person's consent.
- There must be no personal data record-keeping systems whose very existence is secret.
- There must be a way for a person to correct or amend a record of identifiable information about that person.

- There must be a way for a person to find out what information about that person is in a record and how it is used.
- Any organization creating, maintaining, using, or disseminating records of identifiable personal data must ensure the reliability of the data for their intended use and must take steps to prevent misuse of the data."

Many bills and proposals concerning privacy of medical information have preceded the most prominent law, the Health Insurance Portability and Accountability Act (HIPAA), enacted in 1996. In 1995, Senator Robert Bennett (R-Utah) sponsored the Medical Records Confidentiality Act, designed to protect the privacy of medical records. Items addressed in the proposed legislation were:

1. Procedures for individuals to examine their medical records and the ability to correct any errors.
2. Identifies persons and entities with access to individually identifiable information as "health information trustees" and defines circumstances under which that information can be released, with or without patient authorization.
3. Establishes federal certification of health information services, which must meet certain requirements to protect identifiable information.
4. Provides both civil and criminal penalties, up to $500,000 and 10 years' imprisonment, for wrongful disclosure of protected information.

It is important to note that Bennett's bill would apply to medical information in any form, as compared to HIPAA legislation, which calls for the protection of *electronic* medical information. Bennett has indicated his resolve and declared his intention to reintroduce his bill, S.2609 in the 106th Congress in 1999.

Heightened interest in patient rights, sparked partially by tragic stories of individuals who died due to delays in medical treatment, led Senate Democratic Leader Tom Daschle to introduce the Patients' Bill of Rights in March of 1998. This law would guarantee patients greater access to information and necessary care, including access to needed specialists and emergency rooms, guarantee a fair appeals process when health plans deny care, expand choice, protect the doctor–patient relationship, and hold HMOs accountable for decisions that end up harming patients. Daschle's bill also:

- Requires plans and issuers to establish procedures to safeguard the privacy of any individually identifiable enrollee information.
- Maintains records and information in an accurate and timely manner.
- Assures the individual's timely access to such records and information.

Additionally, other organizations committed to strong privacy legislation, such as the Electronic Privacy Information Center (EPIC), have proposed multiple versions of similar bills. Most call for stringent controls over medical records. Many go beyond and call for advanced technical controls, including encryption and audit trails which record every access to every individual.

THE MULTITUDE OF PRIVACY LAWS PROPOSED IN RECENT MONTHS

The federal government, very aware of its citizens' concerns, is answering their outcry with no less than a dozen healthcare privacy laws, proposed in recent congressional sessions. Some of the most publicized are:

- McDermott Bill, a.k.a. "Medical Privacy in the Age of New Technologies Act" — 1997
- Jeffords–Dodd Bill, a.k.a. "Health Care Personal Information Non-Disclosure Act" — 1998
- Senate Bill S.2609, a.k.a. the Bennett Bill. This proposed legislation is important to note because it addresses information in all media, whereas the other bills address the protection of information in electronic format only.
- Kennedy–Kassebaum Bill, a.k.a. the Health Insurance Portability and Accountability Act (HIPAA) — 1996

> "Electronic medical records can give us greater efficiency and lower cost. But those benefits must not come at the cost of loss of privacy. The proposals we are making today will help protect against one kind of threat — the vulnerability of information in electronic formats. Now we need to finish the bigger job and create broader legal protections for the privacy of those records."
>
> *— The Honorable Donna E. Shalala, 1997*

Kennedy–Kassebaum Bill: Background

Several iterations of congressional hearings occurred where stories were told of citizens suddenly found to be uninsurable because they had changed jobs. These instances of insurance loss led to a plethora of tragic incidents, motivating Senators Edward M. Kennedy (D-Massachusetts) and Nancy Kassebaum (R-Kansas) to propose the legislation known as the Kennedy–Kassebaum Bill, also known as HIPAA. Because approximately two thirds of Americans are insured through their employers, the loss of a job often means the loss of health insurance — thus the justification for the term "portability," enabling individuals to port their health plan coverage to a new job. Legislators took this opportunity to incorporate privacy provisions into the bill, and thus, under HIPAA, the Health Care Financing

Administration (HCFA) has issued a series of proposed rules that are designed to make healthcare plans operate securely and efficiently.

"For the Record": The Report

In 1997, the government-sponsored National Research Council report, "For the Record: Protecting Electronic Health Information," captured the essence of the status of security in the healthcare industry. The report came to several conclusions, which laid the foundation for the call from Congress and the Department of Health and Human Service, to define security standards for the healthcare industry. The report concluded:

1. Improving the quality of healthcare and lowering its cost will rely heavily on the effective and efficient use of information technology; therefore, it is incumbent on the industry to maintain the security, privacy, and confidentiality of medical information while making it available to those entities with a need.
2. Healthcare organizations, including health maintenance organizations (HMOs), insurance companies, and provider groups, must take immediate steps to establish safeguards for the protection of medical information.
3. Vendors have not offered products with inherent protective mechanisms because customers are not demanding them.
4. Individuals must take a more proactive role in demanding that their personally identifiable medical information is protected adequately.
5. Self-regulation has not proven successful; therefore, the state and federal governments must intercede and mandate legislation.
6. Medical information is subject to inadvertent or malicious abuse and disclosure, although the greatest threat to the security of patient healthcare data is the authorized insider.
7. Appropriate protection of sensitive healthcare data relies on both organizational policies and procedures as well as technological countermeasures.

Satisfying these important security and privacy considerations is the basis for the administrative simplification provisions of HIPAA. At last, the healthcare industry is being tasked to heed the cry that the citizenry has voiced for years, "Maintain my privacy and keep my personal, sensitive information private."

HIPAA ADMINISTRATIVE SIMPLIFICATION: SECURITY STANDARDS

The specific rules that apply to security standards that protect healthcare-related information (code set 6 HCPR 1317) were issued August 17, 1998, for public comment. The deadline for comment was October 13, 1998. According to HCFA, the volume of comments received was extraordinary.

Plans and providers cried that implementation of the standards would be onerous and cost-prohibitive. HCFA essentially replied that "security is a cost of doing business" and the deadlines will stand. Those deadlines include adoption of security standards by 2002. Moreover, HIPAA requires Congress to pass comprehensive privacy legislation to protect individual health information by August 1999. If lawmakers fail to meet that deadline, then the responsibility falls to the Secretary of DHHS to promulgate protections by February 2000.

Throwing her full support behind HIPAA security standards, Shalala stated, "When Americans give out their personal health information, they should feel like they're leaving it in good, safe hands. ….Congress must pass a law that requires those who legally receive health information to take real steps to safeguard it."

President Bill Clinton has publicly supported privacy legislation for the healthcare industry since 1993. In a May 1997 speech at Morgan State University, the President reiterated that "technology should not be used to break down the wall of privacy and autonomy that [sic] free citizens are guaranteed in a free society."

Horror stories of inadvertent or malicious use or disclosure of medical information are held closely by healthcare organizations. No corporate officer wants to admit that information has "leaked" from his company. However, there are several publicized war stories in which sensitive patient healthcare information has been disclosed without proper authorization, resulting in misfortune and tragedy. For example, when former tennis star Arthur Ashe was admitted to a hospital due to chest pains, his HIV-positive status was discovered and leaked to the press, causing great embarrassment and strife not only to Ashe and his family, but to the medical institution as well.

In another instance, a claims processor brought her young teenager to work and sat her in front of a terminal to keep her occupied. The daughter accessed a database of patients who had been diagnosed with any number of maladies. The teenager concocted a game whereby she called several of the patients, pretended to be the provider, and misreported the diagnoses. One patient was told he had contracted AIDS. The man committed suicide before he could be told the report was the prank of a mischievous child.

In another instance, a healthcare maintenance employee, undergoing a nasty child custody battle with his wife's sister, gained access to his company's system, where he discovered some sensitive information about his sister-in-law, also covered by the health plan. He revealed this information in court in an attempt to discredit her. She sued the health plan for negligence and won the case.

These scenarios are not as rare as we would like to believe. The existing legal structure in healthcare does not provide for effective control of patient medical information. The federal government recognizes this and has attempted to forcefully impose stringent regulation over the protection of health information.

Under HIPAA, healthcare organizations must develop comprehensive security programs to protect patient-identifiable information or face severe penalties for noncompliance. Industry experts estimate that HIPAA will be the "next Y2K" in terms of resources and level of effort, and that annual healthcare expenditures for information security will increase from $2.2 million to $125 million over the next 3 years.

The HIPAA standards, designed to protect all electronic medical information from inadvertent or intentional improper use or disclosure, include provisions for the adoption of:

1. Organizational and administrative procedures
2. Physical security safeguards
3. Technological security measures

Health plans have until early 2002 to adopt these requirements. Although the intent of the standards should be uniform and consistent across the healthcare industry, considerable interpretation might alter the implementation of the controls from one organization to another. The HIPAA security requirements are outlined below.

1. Organizational and Administrative Procedures

1. Ensure that organizational structures exist to develop and implement an information security program. This formal, senior management-sponsored and supported organizational structure is required so that the mechanisms needed to protect information and computing resources are not overridden by a senior manager from another function, for example, Operations or Development, with their own "agendas" in mind. This requirement also includes the assignment of a Chief Security Officer responsible for establishing and maintaining the information security program. This program's charter should ensure that a standard of due care and due diligence is applied throughout the enterprise to provide an adequate level of assurance for data security (integrity/reliability, privacy/confidentiality, and availability).
2. The Chief Security Officer is responsible for the development of policies to control access to and for the release of, individually identifiable patient healthcare information. The over-arching information security policy should declare the organization's intent to comply with regulations and protect and control the security of its

information assets. Additional policies, standards, and procedures should define varying levels of granularity for the control of the sensitive information. For example, some of the policies may relate to data classification, data destruction, disaster recovery, and business continuity planning.

One of the most important organizational moves that a healthcare organization must make for HIPAA compliance is in appointing a Chief Security Officer (CSO). This person should report at a sufficiently high level in the organization so as to be able to ensure compliance with regulations. Typically, the CSO reports to the Chief Information Officer (CIO) or higher. This function is tasked with establishing the information security program, implementing best practices management techniques, and satisfying legal and regulatory requirements. Healthcare organizations seeking qualified, experienced security officers prefer or require candidates to be certified information system security professionals (CISSPs). This certification is offered solely by the non-profit International Information Systems Security Certification Consortium (ISC^2) in Massachusetts. More information about professional certification can be obtained from the organization's Web site at www.isc2.org.

3. The organization is required to establish a security certification review. This is an auditable, technical evaluation establishing the extent to which the system, application, or network meets specified security requirements. The certification should also include testing to ensure that the controls actually work as advertised. It is wise for the organization to define control requirements up front and ensure that they are integrated with the business requirements of a system, application, or network. The certification documentation should include details of those control requirements, as well as how the controls are implemented. HIPAA allows for the certification to be done internally, but, it can also be done by an external agency.
4. Establish policies and procedures for the receipt, storage, processing, and distribution of information. Realizing that information is not maintained solely within the walls of an individual organization, HIPAA calls for an assurance that the information is protected as it traverses outside. For example, an organization should develop a policy that mandates authorization by the business owner prior to sending specific data to a third-party business partner.
5. Develop a contractual agreement with all business partners, ensuring confidentiality and data integrity of exchanged information. This standard may manifest itself in the form of a confidentiality clause for all contractors and consultants, which will bind them to maintain the confidentiality of all information they encounter in the performance of their employment.

6. Ensure access controls that provide for an assurance that only those persons with a need can access specific information. A basic tenet of information security is the "need to know." This standard requires that appropriate access is given only to that information an individual requires in order to perform his job. Organizations should establish procedures so that a business manager "owns" the responsibility for the integrity and confidentiality of the functional information, e.g., Claims, and that this manager authorizes approval for each employee to access said information.
7. Implement personnel security, including clearance policies and procedures. Several organizations have adopted human resources procedures that call for a background check of their employment candidates. This is a good practice and one that is recognized as an HIPAA standard. Employees, consultants, and contractors, who have authorized access to an organization's information assets, have an obligation to treat that information responsibly. A clearance of the employee can guarantee a higher degree of assurance that the organization can entrust that individual with sensitive information.
8. Perform security training for all personnel. Security education and awareness training is probably the most cost-effective security standard an organization can adopt. Information security analyses continually reflect that the greatest risk to the security of information is from the "insider threat."
9. Provide for disaster recovery and business resumption planning for critical systems, applications, and networks.
10. Document policies and procedures for the installation, networking, maintenance, and security testing of all hardware and software.
11. Establish system auditing policies and procedures.
12. Develop termination procedures which ensure that involuntarily terminated personnel are immediately removed from accessing systems and networks and voluntarily terminated personnel are removed from systems and networks in an expedient manner.
13. Document security violation reporting policies and procedures and sanctions for violations.

2. Physical Security Safeguards

1. Establish policies and procedures for the control of media (e.g., disks, tapes), including activity tracking and data backup, storage, and disposal.
2. Secure work stations and implement automatic logout after a specified period of nonuse.

3. Technological Security Measures

1. Assure that sensitive information is altered or destroyed only by authorized personnel.
2. Provide the ability to properly identify and authenticate users.
3. Create audit records whenever users inquire or update records.
4. Provide for access controls that are either transaction-based, role-based, or user-based.
5. Implement controls to ensure that transmitted information has not been corrupted.
6. Implement message authentication to validate that a message is received unchanged.
7. Implement encryption or access controls, including audit trails, entity authentication, and mechanisms for detecting and reporting unauthorized activity in the network.

One of the biggest challenges facing the organizations that must comply with HIPAA security standards is the proper interpretation of the regulation. Some of the standards are hazy at this time, but the fines for noncompliance are well-defined. HIPAA enforcement provisions specify financial and criminal penalties for wrongful disclosure or willful misuse of individually identifiable information at $250,000 and 10 years of imprisonment per incident.

SUMMARY

The reader can see that the security manager in the healthcare industry has an ominous task, and federal regulations make that task an urgent one. However, with the adoption of generally accepted system-security principles and the implementation of best-security practices, it is possible to develop a security program that provides for a reasonable standard of due care, and one that is compliant with regulations.

Protecting High-Tech Trade Secrets

William C. Boni

As business organizations enter the 21st century, it is vital that the managers and executives who lead them understand that there is a wide array of dark new threats. These threats strike at the core of what is increasingly the organization's most critical assets — the information, intellectual property and unique "knowledge value" which has been acquired in designing, producing, and delivering products and services. Many of these threats arise from the digital properties now associated with forms of critical information. The methods and techniques of acquiring sensitive information, which were previously available only to the world's leading intelligence services, are now widely available to anyone willing to engage "retired" professionals or acquire sophisticated electronic equipment. These capabilities create a host of new vulnerabilities that extend far beyond the narrow focus on computers and networks. The risk to company information increases as both people and technology, honed in the Cold War, now move into collecting business and technology secrets. Information protection programs for leading organizations must move beyond the narrow focus of physical security and legal agreements, to a program that safeguards their proprietary rights. A new awareness derived from assessing security implications of operational practices and applying a counter-intelligence mindset are essential to protect the enterprises' critical information assets against sophisticated and determined adversaries.

The new opponents of an organization may range from disgruntled insiders seeking revenge, to unethical domestic competitors, to a foreign nation's intelligence services operating on behalf of their indigenous "national flag" industry participant. Such opponents will not be deterred or defeated by boilerplate legal documents nor minimum-wage security guards. Defeating these opponents requires a well-designed and carefully implemented program to deter, detect, and if necessary, actively neutralize efforts to obtain information about the organization's plans, products, processes, people, and facilities capabilities, intentions, or activities.

The fact is that few in business truly appreciate the arsenal now available to "The Dark Side," which is how many protection professionals refer to those who steal the fruits of other's hard work. Understanding how "technology bandits" operate, their methods, targets, capabilities, and limitations, is essential to allow the organization to design safeguards to protect its own critical information against the new dangers. It is also important that managers understand they have a responsibility to help level the global playing field by encouraging foreign and domestic competitors to conform to a common ethical standard. The common theme must be fair treatment of the intellectual property of others. When an organization detects an effort to improperly obtain its intellectual property and trade secrets, it must use the full sanctions of relevant laws. In the U.S., companies now may benefit by seeking federal felony prosecutions under the Economic Espionage Act of 1996!

TRADE SECRET OVERVIEW AND IMPORTANCE

In any discussion of intellectual property and organizational information, it is first important to understand the distinction between trade secrets and patents. The U.S. (or any other national government) grants a patent to the inventor of a novel and useful product or process. In exchange for public disclosure of required information, the government grants the inventor exclusive benefits of ownership and profits derived from ownership for a period of time, commonly 17 years from date of issue or 20 years from date of application for a patent.

However, a business may decide that as a practical matter, it may ultimately derive more commercial advantages by maintaining as a "trade secret" the information, product, or process. The term "trade secret," for those from military or governmental backgrounds, is not the same as national security or "official" secrets. In identifying something as a trade secret, it qualifies as a special form of organizational property, which may be protected against theft or misappropriation. Essentially it means information, generally but not exclusively of a scientific or technical nature, which is held in confidence by the organization and which provides some sort of competitive advantage. The major advantage of protecting something as a trade secret rather than as a patent is that the company may, if it exercises appropriate oversight, continue to enjoy the profits of the "secret" indefinitely.

A practical example of a trade secret's potential for "unlimited" life is the closely guarded formula for Coca-Cola, which has been a carefully protected trade secret for over 80 years. However, there is a downside of protecting valuable discoveries as trade secrets. If the organization fails to take reasonable and prudent steps to protect the secret, they may lose

some or all of the benefits of trade secret status for its information. This may allow another organization to profit from the originator's hard work!

Proprietary Information and Trade Secrets

As a practical matter, all of the information which a company generates or creates in the course of business operations and practices can be considered "proprietary." The dictionary defines proprietary as "used, made, or marketed by one having the exclusive legal rights" (*Webster's Collegiate*), which essentially means the company has an ownership right to its exclusive use. Although ALL trade secrets ARE proprietary information, not *all* proprietary information will meet the specific legal tests which are necessary to qualify them as trade secrets. Therefore, trade secrets are a specialized subset of proprietary information, which meet specific tests established in the law. Trade secrets statutes under U.S. laws provide the following three elements that must *all* be present for a specific piece or category of information to qualify for trade secret status:

- *The information MUST be a genuine, but not absolute or exclusive, "SECRET."* This means that an organization need not employ draconian protection measures and also that even though elements of the secret, indeed the secret itself, may be discoverable, through extraordinary (even legal means), it nonetheless is not generally apparent, and may thus qualify for trade secret status. The owner may even license the secret to others, and as long as appropriate legal and operational protections are applied, it remains a protected asset. It is also possible that a trade secret may be independently discoverable and usable by a competitor, and it can simultaneously be a trade secret for both developers!
- *It must provide the owner competitive or economic advantages.* This means the secret must have real (potential) business value to the holder/owner. A business secret that merely conceals inconsequential information from the general public cannot be protected as a trade secret.
- *The owner must take "reasonable" steps to protect the secret.* For those involved in both protection of an organization's trade secrets as well as those whose responsibility includes ferreting out the business strategies of competitors, *this* is the most crucial element in qualifying for trade secret status and attendant rights. Regrettably, neither courts nor legislatures have provided a convenient checklist of the minimum measures to qualify for the "reasonable" steps. Over the years, courts have applied the "reasonable" test and in a series of cases, defined commonly accepted minimum measures. In many cases the courts have ruled that a plaintiff's lack of a specific safeguard defeated their claim of trade secrets status for the information at

issue. It is critical to understand that a court's decision as to what is necessary to protect an organization's trade secrets will depend on what is "reasonable" under the specific circumstances of a given situation, and therefore is extremely difficult to predict in advance of a trial. As a general standard, the protections that are "reasonable" will also reflect the common business practices of a particular industry.

Economic Espionage Act (EEA) of 1996

The single most significant development in trade secret protection in the U.S. was passage of the EEA in 1996. Title 18 USC sections 1831 and 1832 were added to the federal statutes after a series of disappointing cases became public which proved the need for new laws to deal with theft of technology and trade secrets. When President Clinton signed this act into law on October 11, 1996, American industry was given a strong weapon designed to combat the theft of trade secrets. The act created for the first time a *federal* law that criminalized the theft or misappropriation of organizational trade secrets, whether done by domestic or foreign competitors or by a foreign governmental entity. A key clause in the act defines trade secrets:

EEA Definition of Trade Secrets. The term "trade secret" means all forms and types of financial, business, scientific, technical, economic, or engineering information, including patterns, plans, compilations, program devices, formulas, designs, prototypes, methods, techniques, processes, procedures, programs, or codes, whether tangible or intangible, and whether or how stored, compiled, or memorialized physically, electronically, graphically, photographically, or in writing if:

1. the owner thereof has taken reasonable measures to keep such information secret; and
2. the information derives independent economic value, actual or potential, from not being generally known to, and not being readily ascertainable through proper means by, the public.

Value of Intellectual Property

In reviewing the definition as to what may qualify as a trade secret under the EEA, it seems that almost anything could be declared a trade secret. This seems to be a prudent approach because advanced business organizations in the developed world are largely based on the knowledge that such organizations have captured, for example, in their design, production, and operational systems. New and more advanced products and services derive from the aggregation of the learning organization knowledge, which is translated into "intellectual property" (abbreviated IP) to distinguish it from the tangible property of the organization. IP is generally con-

sidered to consist of the patents, copyrights, trademarks, and trade secrets of the organization, which are normally lumped into the overall category of "intangible assets" on the balance sheet. Although not reflected in traditional accounting practices, the IP of companies has increasingly become the source of competitive advantage. The significance of these assets is demonstrated by the fact that by some estimates over 50 percent or more of the market capitalization of a typical U.S. company is now subsumed under intangible assets, i.e., primarily intellectual property. Several industry segments are especially dependent on aggregating "knowledge" into their products in order to create valuable intellectual property.

Semiconductors. The most significant IP is not merely the designs (the specific masks or etchings) which are the road map of the chips, but also the exact assembly instructions. Although product lifecycles can be measured in months, the effort of thousands of highly educated engineers working in collaborative teams to design, debug, and manufacture leading-edge chips, should be measured in years. If a competitor has both the masks and the assembly instructions, they may anticipate the originator's target and "leap frog" over a current-generation product in price and performance. Alternatively they may merely join the originator in the market with a "me too" product. Such a strategy may be very attractive to an unethical competitor as it could allow them to remain competitive without investing as much time and resources in primary design as the originator.

Biotechnology and Pharmaceutical Products. Often developed over five to seven years and costing hundreds of millions of dollars each, a successful product will represent the work of hundreds of highly trained scientists, engineers, medical experts, physicians, nurses, and others. This highly educated workforce generates a product, which in the end may only be protected by a "production process" patent. The pure science which provides the foundation for such drugs is often public, so the organization's return on investment may well ride on safeguarding the various unique processes associated with development, production, or delivery of a therapeutic drug. Once again a competitor, especially one from a country where intellectual property rights are not well established or respected, may derive significant advantages by misappropriating or stealing product information early in a product's lifecycle. With luck or planning, such thefts may allow development of a competitive alternative that could be produced at minimum cost to the competitor and marketed locally with the encouragement or support of the national government.

Software Products. Without question, the rapid pace of information technology would not be as fast in the absence of sophisticated software products. Applications harness the raw horsepower of the silicon chip and deliver control to a user's business needs. Such tools benefit from highly

skilled programmers working collaboratively to fashion new features and functionality. Their knowledge is captured in the product and becomes the source of an organization's ability to deliver new products.

Source code for new or unreleased software may be targeted by unscrupulous competitors or spirited away by employees lured away by better pay or working conditions. Too often, applications development staff will take with them copies of any new software they helped develop or to which they had access during their term of employment. This is an especially serious problem when contract programmers are employed, because by the nature of their assignments, they know their term is limited (e.g., Year 2000). Thus, they may be tempted to market a product developed for one client to another.

Sensitive Information Is Often Portable and Digital

Sensitive proprietary information and other valuable intellectual property including an organization's trade secrets are now often captured in some digital form. Critical trade secrets worth billions of dollars may be contained in CAD/CAM drawing files, a genetics database, or compiled source code for a breakthrough software application. This digital form creates a whole new class of problems that must be considered by protection professionals. Most new products owe their existence to the computers, networks, and users of those systems. However, in a digital state, and in a typical client-server-based systems environment, the "crown jewels" of organizational sensitive proprietary information are often poorly protected against unauthorized access. Such access may allow the hostile intruder or the malicious insider to purloin a duplicate of the original data, and perhaps corrupt or destroy the original. In a matter of seconds, a misappropriated copy of the corporate "crown jewels" can be sent to an exotic location on the other side of the planet. From there the thief may auction it off to the highest bidder or sell it to a competitor. This frightening possibility should, in and of itself, inspire the senior managers of leading companies to give increased priority to computer and network security. As we shall discuss a little later, it seems many organizations have not yet fully recognized the many risks to their intellectual property and trade secrets that poorly controlled systems and networks create.

Increased Potential for "Loss of Control"

As more organizations deploy network technology and as the IP crown jewels become more digital and portable, it's possible, perhaps even likely, that management will lose control of these key assets. Without constant attention, testing, and monitoring, the risk of a catastrophic loss of control and of the IP assets themselves is high.

Typical Confidential Information

Managers who apply themselves can quickly identify a list of the information about their organization that they consider confidential and which may be considered as sensitive and proprietary information that may also qualify for "trade secret" status. The difference between "confidential" and merely "proprietary" is often based on management's assessment of the competitive advantage that accrues to the organization by managing dissemination of the information. However, given the vast quantity of proprietary information created and stored by contemporary organizations, it is essential to stratify information. This essential step allows organization management to identify the truly critical proprietary information from items that are merely sensitive. Napoleon's maxim of war is appropriate to consider, "He who defends everything, defends nothing!" If an organization does not stratify or prioritize its information assets it is likely to spend too much time and money protecting the "crown jewels" (which typically also qualify as trade secrets), and mundane, low-value information equally. Alternatively, they may not invest sufficiently in protecting their core assets and lose considerable advantage when trade secrets and other critical information are compromised.

In a systematic and well-planned project, managers and corporate attorneys should consider what information, both by type and content, are of value and importance to the organization's business operations, capabilities, and intentions. From this list of valuable information the company should then identify those items or elements of information which are real sources of competitive advantage. Of this last group, the organization should determine which, if any, may qualify for trade secret status. Note that in this process it is likely that some very valuable and useful information will provide competitive advantage, but may not be protectable as a trade secret.

Unquestionably there will be trade secrets that have previously not been considered as such. The following list, while not all-inclusive, at least provides a point of departure for creating an organizational inventory which may be supplemented with industry and organization specific categories.

- Business plans and strategies
- Financial information
- Cost of research, development, and production
- New products: pricing, marketing plans, timing
- Customer lists, terms, pricing
- Research and development priorities, plans, activities
- Inventions and technology information
- Unique or exceptional manufacturing processes

- Facility blueprints, floor plans, layouts
- Employee records and human resources information

While any or all the above categories of information are likely to be considered "confidential," what does that really mean? Essentially "confidential" information if disclosed, modified, or destroyed, without appropriate controls or authorization, would likely have adverse consequences on the organization's business operations. However, any or all of the above information, plus any that is unique to your business could potentially be identified as a "trade secret" and benefit from additional legal protection providing it meets the previously discussed tests.

This "audit" or inventory procedure should then be taken to at least one more level of detail. In cooperation with the organization's information technology (IT) management and line managers, the specific documents, systems (servers, databases, work stations, document imaging/production, networks, etc.), file cabinets, and work areas (buildings) that contain the identified "trade secrets" and sensitive proprietary information should be identified. These environments should then be reviewed/inspected and the degree of compliance with trade secret protection requirements should be the standard for the inspection. At a minimum, all IT systems which contain trade secret and sensitive proprietary information must provide individual accountability for access to their contents and a secure audit trail of the access activity of specific users. Any systems, which do not provide at least these functions, should be upgraded to such functionality on a priority basis.

NEW THREATS TO SENSITIVE, PROPRIETARY INFORMATION

Threats to an organization's sensitive proprietary information have never been more formidable. Each of the following issues is significant and requires that any existing programs to safeguard the "crown jewels" be reassessed to ensure the risks have been appropriately managed.

Decline in Business Ethics and Loyalty

A recent newspaper headline declared "48% of Employees Lie, Cheat, Steal." However surprising such a statement may seem, the conclusions implied by the title were not fully justified in the supporting article, e.g., many employees engage in relatively innocuous acts of petty theft, such as office supplies. However, within the context of other studies, the conclusion is inescapable, there has been a substantial decline in employee loyalty and an increase in the range of actions that are considered acceptable business practices. As further proof of the overall change in business ethics, consider the story related by Staples' Chairman Thomas Stemberg in his book *Staples for Success*. In the book, the author describes how he

asked his wife to apply for a job with arch rival Office Depot's Atlanta delivery-order center, apparently to gain insights concerning their training methods.

It's also important to appreciate the many changes in work force psychology, which grew out of the downsizing and outsourcing efforts of organizations in the late 1980s and early 1990s. Many workers and mid-level managers learned a harsh lesson: the organization will do without them, regardless of the consequences to the individuals. While such actions may have been necessary to survive in a global economy, many people drew the conclusion that the bond of loyalty between employer and employee had become a one-way street. As a consequence, some decided to do whatever they needed to survive. Once an individual reaches this point, it is easy to rationalize serious criminal behavior on the grounds that "everyone is doing it" or they are only getting their "fair share" before the organization eliminates their job. Although the U.S. economy now seems to have weathered the worst of this period, managers and executives must understand that the base of employee loyalty is often very shallow. Executives should consider the degree of employee loyalty as they design their protection measures, especially for the corporate crown jewels.

The Internet: Hacker Playground

One of the most remarkable changes in the late 20th century has been the explosive growth in the use of the Internet. Until the late 1980s it was the playground for hackers and computer nerds. Since that time, tens of millions of individuals have obtained personal accounts and hundreds of thousands of organizations have established Internet connections. As the number of businesses using "the net" has exploded, so too has the reported rate of computer and network intrusions.

Without question many network based "attacks" are not serious. However, the number and consequence of malicious activity are increasing. The 1997 Computer Security Institute/FBI Survey showed an increase of 36 percent in known instances of computer crime from the 1996 survey. The simple equation is increased network connectivity results in more computer crimes. Organizations that blindly hook up to the net without a well-thought-out protection plan place their sensitive intellectual property and trade secrets at serious risk.

The adverse impact on information protection of the global Internet and the rapid increases in Internet users should not be underestimated. Since the "net" now encompasses all continents and more than 100 countries, it is possible to reach anywhere from anywhere. The plans to circle the globe with low-orbiting satellites will increase both access and mobility. It is important to recognize that the Internet is essentially unregulated, and

that there is NO central management or policing. When something happens, whether an attempted intrusion via the net or an unsolicited Spam storm, organizations often have few alternatives but to help themselves.

Growing Threat of Espionage

Perhaps the least appreciated new threat to organization information is the efforts by some companies and many countries to steal critical business information and trade secrets. Is this a real problem? According to the American Society of Industrial Security (ASIS), U.S. companies may have lost as much as $300 billion in trade secrets and other intellectual property in 1997.

A review of recent high-profile cases in the public domain shows that many well-known companies have been targets of industrial espionage and theft of technology and trade secrets. For example, a very short list would include:

- Intel, whose Pentium chip designs were stolen by an employee and offered to AMD.
- Representatives of a Taiwanese company who were willing to bribe a corrupt scientist to steal the secrets of Bristol Myers Taxol® production process information.
- In the another recent case, Avery-Denison learned that one of their research scientists was selling company information to a foreign competitor.
- In the most famous case in recent times, a former high-ranking executive of General Motors was accused of stealing literally box loads of highly confidential documents and offering them to his new employer, Volkswagen.
- Other cases include a retired engineer who sold Kodak trade secrets and a contract programmer who offered to sell key information concerning Gillette's new shaving system.

These scenarios indicate that the theft of trade secrets is a thriving business. According to the FBI, they have literally hundreds of investigations under way. It's important to note that these represent only some of the cases which are publicly known, and do not include cases which are quietly investigated and resolved by organizations fearful of the adverse publicity attendant to a litigation or prosecution. There are likely an even larger number of cases which go completely undetected and which may contribute to the potential failure of large and successful organizations.

Impact of Global Business Operations

Globalization of business operations is a major trend of the late 20th century. It is now a fact that most business organizations operate and compete

throughout the world. An important factor to consider in global operations is that the standards of business and ethics, which prevail in the heartland of the Midwest, are not necessarily those which exist in remote areas of the world. Nations such as China and various Southeast Asian nations are real challenges, as they do not, at present, honor intellectual property rights to the extent common in much of Europe and North America. Unrelenting competition for survival and success may create situations where theft of trade secrets seems to promise the beleaguered executive an easy way to remain in business without the need to invest as much in developing new products or improving his operations.

Threats from Networks, Computers and Phones

Generally it has been argued that advanced nations have reaped increased productivity through many benefits of sophisticated communication. With regard to protecting trade secrets, such technologies raise a host of questions. First, as they proliferate throughout the organization, WHERE are the organization's secrets? This is more than just a question of primary physical storage. To properly answer the question, the organization must consider both hard copy documents, individual desktop microcomputers, file servers, databases, backup files/media, as well as imaging/document management and other computer and networking systems.

The myriad of locations and variety of forms and formats which may contain sensitive proprietary information makes it very difficult, sometimes impossible, to know with certainty WHO has access to company secrets! And in cases where management believes they have adequate control over access to sensitive proprietary information, HOW do they really know? Too often managers rely on simple assertions from the Management Information Systems (MIS) and Information Technology (IT) staff that the system and network controls are adequate to protect the organizational crown jewels. Given the importance of the topic and complexity of the environments, senior management is well advised to verify actual conditions of the security and control measures on a periodic basis.

The advent of inter-organizational networks, typically dubbed "extranets," should cause managers concerned with safeguarding their crown jewels to take a hard look at the function and features of the environment. Without careful attention to the configuration and management, it is possible that outsiders will be able to gain access to organization information that extends well beyond the legitimate scope of the relationship.

WHAT MUST BE DONE?

Managers who appreciate the full nature and scope of the threat to sensitive proprietary information and trade secrets must implement

protective measures to mitigate the most likely vulnerabilities of their organizations. With regard to protecting trade secrets, there are some measures which have been found to be essential. There are now many additional security measures, which are highly recommended, even though they have not yet been held to be essential.

Required Protection Measures for Protecting Trade Secret Information

Although the courts in the U.S. have not published any sort of handbook which describes required protective measures to safeguard intellectual property, review of various case decisions provides various examples where judges have ruled in such a way that clearly indicated the desirability of the security measure.

Visitor Sign In and Escort. Common sense indicates all non-employees entering the company facility should be escorted by host employees, sign in at reception, and be retained until the host escort arrives. Too often, once inside the facility, host employees' excessive hospitality gives the visitor free reign of the site. In the absence of well-maintained internal perimeters, visitors may obtain accidental or deliberate access to sensitive areas, files, documents, and materials. Also, the unguarded conversations of co-workers unaware of the status of the listener may result in disclosure of sensitive information.

Identification Badges. Distinctive badges with photo provide good control over egress and exit. These are also so inexpensive that organization management would appear foolish if they failed to implement some sort of badging system.

Facility Access Control System. Often tied into the photo-ID badge system used by the organization, facility access control systems provide convenient and automated authentication technology. In the past, card readers alone were sufficient. However, many sophisticated organizations with significant assets are implementing biometric (voice, hand geometry, or retina) systems. Such systems dramatically curtail the potential for abuse.

Confidentiality/Nondisclosure Documents. These confidentiality and nondisclosure statements should specify invention assignments as well as an agreement to protect proprietary information.

Exit Interviews with Terminating Employees. Remind employees that are leaving the company of their continuing obligation to protect any trade secrets to which they had access during the time of their employment.

Other "Reasonable" Measures! The courts have a remaining variable, which can be very important. They may decide, entirely after the fact, that

a given organization did or did not act "reasonably" by implementing or failing to implement a specific protective measure. The important fact for protection professionals to consider is that the outcome of a particular ruling is not possible to predict in advance of a trial and a specific set of circumstances.

Recommended Protection Measures

Develop and Disseminate Policies and Procedures. Although not strictly required, a policy that spells out the need for information protection and a procedural framework that addresses issues in both electronic and physical media is a useful tool.

Publication Approval Procedures. Disclosure of the trade secret information in publications will eliminate their trade secret status. Even if the proprietary information disclosed in an article, interview, or press release is not a trade secret, it may damage the company's competitive position. A publication screening procedure involving the company's patent staff or other knowledgeable attorneys, as well as other knowledgeable management, should consider not merely whether the content discloses trade secrets, but also whether it reveals competition-sensitive details. If available, the competitive intelligence group can render valuable service in advising on sensitivity. One must assume that the competitive intelligence analysts working for the most competent opponent will see the release /article and place it in appropriate context.

Contract Language for Vendors, Suppliers, etc. All vendors who provide products, services, even parts and supplies should be required to adhere to a basic confidentiality agreement concerning the nature and extent of the relationship with the company. Appropriate language should be inserted in the contract terms and conditions, specifying exactly how the vendor will act with regard to sensitive proprietary information to which they are granted access in the course of business. In the case of critical suppliers who provide unique or highly specialized elements which are essential to the company's success, it is appropriate to include a supplemental "security guidelines" document. This document should provide additional guidance and direction to the vendor describing (see example table of contents for a typical security guideline for a reprographic service provider).

1. Receipt
2. Storage
3. Handling
4. Work in process
5. Release of finished product
6. Destruction of overruns, QC failed copies, etc.
7. Reportable Incidents

Train Employees. Everyone who creates, processes, and handles company trade secrets and other sensitive proprietary information should be trained. This includes both regular (full-time) as well as contingent employees (temporaries, contractors, consultants, as well as part-time employees). They all need to know what is specifically considered trade secrets of the company, as well as what elements of information may not be trade secrets but are nonetheless considered critical and must not be disclosed outside the company without authorization from appropriate management. Training topics typically include the following:

- Identification of company trade secrets and sensitive proprietary information
- Marking
- Storage
- Physical transportation of hard copy documents and media
- Electronic transmission and storage of documents, materials
- Destruction of physical and electronic copies
- Reportable incidents

In addition a version of training should be tailored to the needs of the contingent employees, which commonly include temporary (clerical) staff as well as any on-site contractors, consultants, or vendor employees.

New-Hire Training Classes. One of the best ways to help people in an organization to change is to indoctrinate the newly hired staff. This way you get your message to the new people before they develop bad habits. This will gradually create a critical mass of supporters for the organization's program to protect information, trade secrets, and other valuable intellectual property. This class and supporting documentation should instruct all employees in the value of trade secrets and company IP, as well as correct procedures for safeguarding these assets.

Develop Incident Response Capability. Assume the worst and you will not be disappointed! There will come a time when the company knows or suspects trade secrets or other valuable intellectual property has been stolen or misappropriated. The statistics are very compelling: nearly 50% of high-technology companies experienced theft or misappropriation of trade secrets in a 1988 Institute of Justice study. Planning for that day is essential. Knowing who to call and what to do will maximize the company's chances for a successful prosecution or litigation.

Conduct Audits, Inspections, and Tests. One of the best ways to know the risks is to conduct a formal trade secret audit or inspection. The process, which must always be conducted under attorney-client privilege, should be a comprehensive review of the company's current inventory of trade secrets, including how well they are managed and protected. A useful

extension to the basic review is to conduct a "valuation estimate" for trade secrets and other critical intellectual property. Such estimates, conducted prior to any possible losses, are a useful guide to management. When estimated values of IP are presented in dollars and cents, it will allow a more rational allocation of investment in protecting what may have seemed previously unsubstantial assets.

CONCLUSION: DON'T RELY EXCLUSIVELY ON THE COURTS TO PROTECT YOUR SECRETS!

If the reader takes only one lesson from this chapter it should be this: Although the legal system exists to provide redress for crimes and grievances through criminal prosecution and civil litigation, the process is laden with uncertainty and burdened with very high costs. It is estimated that General Motors spent millions of dollars pursuing Volkswagen and former executives for alleged theft of trade secrets. Even though in the end they prevailed, it was uncertain whether the German courts would find in favor of GM when the action was initiated. When the vagaries of international relations and politics are overlaid on top of the legal variables, it becomes obvious that prevention is a vastly preferable strategy.

Too often it seems that the organizations value more highly their capability to litigate and prosecute for theft or misappropriation of trade secrets. In the long run it is likely to be effective and more efficient to take reasonable steps to prevent incidents. It is important that management understand that a well-designed information protection program and aggressive, early intervention will often eliminate costly and uncertain legal conflicts. Of course, one could be cynical and assume that some attorneys relish the opportunity to showcase their awesome legal expertise on behalf of clients. There is the potential that such displays of capability will occur less frequently if organizations invest more in procedures and technologies designed to prevent and detect the attempts to steal sensitive proprietary information and trade secrets. However, it's more likely that many lawyers, the same as many executives, do not yet appreciate the vast scope of the problem and are merely applying their past experience.

In summary then, executive management should understand that:

1. Many thefts of sensitive proprietary information are preventable
2. Those that are not prevented can be detected earlier, thus minimizing potential losses
3. A well-designed protection program will enhance the organization's probability for successful prosecution and litigation.

87

How to Work with a Managed Security Service Provider

Laurie Hill McQuillan, CISSP

Throughout history, the best way to keep information secure has been to hide it from those without a need to know. Before there was written language, the practice of information security arose when humans used euphemisms or code words to refer to communications they wanted to protect. With the advent of the computer in modern times, information was often protected by its placement on mainframes locked in fortified rooms, accessible only to those who were trusted employees and capable of communicating in esoteric programming languages.

The growth of networks and the Internet have made hiding sensitive information much more difficult. Where it was once sufficient to provide a key to those with a need to know, now any user with access to the Internet potentially has access to every node on the network and every piece of data sent through it. So while technology has enabled huge gains in connectivity and communication, it has also complicated the ability of networked organizations to protect their sensitive information from hackers, disgruntled employees, and other threats. Faced with a lack of resources, a need to recover from an attack, or little understanding of secure technology, organizations are looking for creative and effective ways to protect the information and networks on which their success depends.

Outsourcing Defined

One way of protecting networks and information is to hire someone with security expertise that is not available in-house. Outsourcing is an arrangement whereby one business hires another to perform tasks it cannot (or does not want to) perform for itself. In the context of information security, outsourcing means that the organization turns over responsibility for its information or assets security to professional security managers. In the words of one IT manager, outsourcing "represents the possibility of recovering from the awkward position of trying to accomplish an impossible task with limited resources."[1] This promising possibility is embodied in a new segment of the information security market called managed system security providers (MSSPs), which has arisen to provide organizations with an alternative to investing in their own systems security.

Industry Perspective

With the exception of a few large companies that have offered security services for many years, providing outsourced security is a relatively new phenomenon. Until the late 1990s, no company described itself exclusively as a provider of security services; while in 2001, several hundred service and product providers are listed in MSSP directories. One company has estimated that companies spent $140 million on security services in

1999; and by 2001, managed security firms had secured almost $1 billion in venture capital.[2] Another has predicted that the demand for third-party security services will exceed $17.2 billion by the end of 2004.[3]

The security products and services industry can be segmented in a number of different ways. One view is to look at the way in which the outsourced service relates to the security program supported. These services include performance of short-term or one-time tasks (such as risk assessments, policy development, and architecture planning); mid-term (including integration of functions into an existing security program); and long-range (such as ongoing management and monitoring of security devices or incidents). By far, the majority of MSSPs fall into the third category and seek to establish ongoing and long-term relationships with their customers.

A second type of market segmentation is based on the type of information protected or on the target customer base. Some security services focus on particular vertical markets such as the financial industry, the government, or the defense industry. Others focus on particular devices and technologies, such as virtual private networks or firewalls, and provide implementation and ongoing support services. Still others offer combinations of services or partnerships with vendors and other providers outside their immediate expertise.

The outsourcing of security services is not only growing in the United States or the English-speaking world, either in terms of organizations that choose to outsource their security or those that provide the outsourced services. Although many U.S. MSSP companies have international branches, MSSP directories turn up as many Far Eastern and European companies as American or British. In fact, these global companies grow because they understand the local requirements of their customer base. This is particularly evident in Europe, where International Security Standard (ISO) 17799 has gained acceptance much more rapidly than in the United States, providing guidance for good security practices to both client and vendor organizations. This, in turn, has contributed to a reduction in the risk of experiencing some of the outsourcing performance issues described below.

Future Prospective

Many MSSPs were formed during the dot.com boom of the mid-1990s in conjunction with the rapid growth of E-commerce and the Internet. Initially, dot.com companies preferred to focus on their core businesses but neglected to secure that business, providing quick opportunity for those who understood newly evolving security requirements. Later, as the boom turned to bust, dot.coms took their expertise in security and new technology and evolved themselves into MSSPs.

However, as this chapter is being written in early 2002, while the number of MSSPs is growing, a rapid consolidation and fallout among MSSPs is taking place — particularly among those that never achieved financial stability or a strong market niche. Some analysts "expect this proliferation to continue, but vendors over the next year will be sharply culled by funding limits, acquisition, and channel limits. Over the next three years, we expect consolidation in this space, first by vendors attempting multifunction aggregation, then by resellers through channel aggregation."[4]

Outsourcing from the Corporate Perspective

On the surface, the practice of outsourcing appears to run contrary to the ancient tenet of hiding information from those without a need to know. If the use of networks and the Internet has become central to the corporate business model, then exposing that model to an outside entity would seem inimical to good security practice. So why, then, would any organization want to undertake an outsourcing arrangement?

Relationship to the Life Cycle

The answer to this question lies in the pace at which the networked world has evolved. It is rare to read a discussion of the growth of the Internet without seeing the word *exponential* used to describe the rate of expansion. But while this exponential growth has led to rapid integration of the Internet with corporate business models, businesses have moved more slowly to protect the information — due to lack of knowledge, to immature security technology, or to a misplaced confidence in a vendor's ability to provide secure IT products. Most automated organizations have 20 or more years of experience with IT management and operations, and their IT departments know how to build systems and integrate them. What they have not known and have

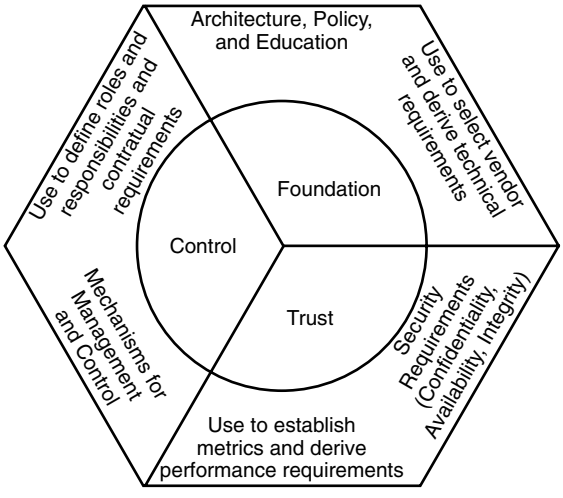


EXHIBIT 87.1 Using a security model to derive requirements.

been slow to learn is how to secure them, because the traditional IT security model has been to hide secret information; and in a networked world, it is no longer possible to do that easily.

One of the most commonly cited security models is that documented by Glen Bruce and Rob Dempsey.[5] This model defines three components: foundation, control, and trust. The foundation layer includes security policy and principles, criteria and standards, and the education and training systems. The trust layer includes the environment's security, availability, and performance characteristics. The control layer includes the mechanisms used to manage and control each of the required components.

In deciding whether to outsource its security and in planning for a successful outsourcing arrangement, this model can serve as a useful reference for ensuring that all aspects of security are considered in the requirements. As shown in Exhibit 87.1, each of the model's components can drive aspects of the arrangement.

The Four Phases of an Outsourcing Arrangement

Phase 1 of an outsourcing arrangement begins when an organization perceives a business problem — in the case of IT, this is often a vulnerability or threat that the organization cannot address. The organization then decides that an outside entity may be better equipped to solve the problem than the organization's own staff. The reasons why this decision is made will be discussed below; but once the decision is made, the organization must put an infrastructure in place to manage the arrangement. In Phase 2, a provider of services is selected and hired. In Phase 3, the arrangement must be monitored and managed to ensure that the desired benefits are being realized. And finally, in Phase 4, the arrangement comes to an end, and the organization must ensure a smooth and nondisruptive transition out.

Phase 1: Identify the Need and Prepare to Outsource

It is axiomatic that no project can be successful unless the requirements are well defined and the expectations of all participants are clearly articulated. In the case of a security outsourcing project, if the decision to bring in an outside concern is made under pressure during a security breach, this is especially true. In fact, one of the biggest reasons many outsourcing projects fail is that the business does not understand what lies behind the decision to outsource or why it is believed that the work cannot (or should not) be done in-house. Those organizations that make the decision to outsource after careful consideration, and who plan carefully to avoid its potential pitfalls, will benefit most from the decision to outsource.

The goal of Phase 1 is to articulate (in writing if possible) the reasons for the decision to outsource. As will be discussed below, this means spelling out the products or services to be acquired, the advantages expected, the legal and business risks inherent in the decision, and the steps to be taken to minimize those risks.

Consider Strategic Reasons to Outsource

Many of the reasons to outsource can be considered strategic in nature. These promise advantages beyond a solution to the immediate need and allow the organization to seek long-term or strategic advantages to the business as a whole:

- Free up resources to be used for other mission-critical purposes.
- Maintain flexibility of operations by allowing peak requirements to be met while avoiding the cost of hiring new staff.
- Accelerate process improvement by bringing in subject matter expertise to train corporate staff or to teach by example.
- Obtain current technology or capability that would otherwise have to be hired or acquired by retraining, both at a potentially high cost.
- Avoid infrastructure obsolescence by giving the responsibility for technical currency to someone else.
- Overcome strategic stumbling blocks by bringing in third-party objectivity.
- Control operating costs or turn fixed costs into variable ones through the use of predictable fees, because presumably an MSSP has superior performance and lower cost structure.
- Enhance organizational effectiveness by focusing on what is known best, leaving more difficult security tasks to someone else.
- Acquire innovative ideas from experts in the field.

Organizations that outsource for strategic reasons should be cautious. The decision to refocus on strategic objectives is a good one, but turning to an outside organization for assistance with key strategic security functions is not. If security is an inherent part of the company's corporate mission, and strategic management of this function is not working, the company might consider whether outsourcing is going to correct those issues. The problems may be deeper than a vendor can fix.

Consider Tactical Reasons

The tactical reasons for outsourcing security functions are those that deal with day-to-day functions and issues. When the organization is looking for a short-term benefit, an immediate response to a specific issue, or improvement in a specific aspect of its operations, these tactical advantages of outsourcing are attractive:

- Reduce response times when dealing with security incidents.
- Improve customer service to those being supported.
- Allow IT staff to focus on day-to-day or routine support work.
- Avoid an extensive capital outlay by obviating the need to invest in new equipment such as firewalls, servers, or intrusion detection devices.
- Meet short-term staffing needs by bringing in staff that is not needed on a full-time basis.
- Solve a specific problem for which existing staff does not have the expertise to address.

While the tactical decision to outsource might promise quick or more focused results, this does not necessarily mean that the outsourcing arrangement must be short-term. Many successful long-term outsourcing arrangements are viewed as just one part of a successful information security program, or are selected for a combination of strategic and technical reasons.

Anticipate Potential Problems

The prospect of seeing these advantages in place can be seductive to an organization that is troubled by a business problem. But for every potential benefit, there is a potential pitfall as well. During Phase 1, after the decision to outsource is made, the organization must put in place an infrastructure to manage that arrangement. This requires fully understanding (and taking steps to avoid) the many problems that can arise with outsourcing contracts:

- Exceeding expected costs, either because the vendor failed to disclose them in advance or because the organization did not anticipate them

- Experiencing contract issues that lead to difficulties in managing the arrangement or to legal disputes
- Losing control of basic business resources and processes that now belong to someone else
- Failing to maintain mechanisms for effective provider management
- Losing in-house expertise to the provider
- Suffering degradation of service if the provider cannot perform adequately
- Discovering conflicts of interest between the organization and the outsourcer
- Disclosing confidential data to an outside entity that may not have a strong incentive to protect it
- Experiencing declines in productivity and morale from staff who believe they are no longer important to the business or that they do not have control of resources
- Becoming dependent on inadequate technology if the vendor does not maintain technical currency
- Becoming a "hostage" to the provider who now controls key resources

Document Requirements and Expectations

As discussed above, the goal of Phase 1 is to fully understand why the decision to outsource is made, to justify the rationale for the decision, and to ensure that the arrangement's risks are minimized. Minimizing this risk is best accomplished through careful preparation for the outsourced arrangement.

Thus, the organization's security requirements must be clearly defined and documented. In the best situation, this will include a comprehensive security policy that has been communicated and agreed to throughout the organization. However, companies that are beginning to implement a security program may be hiring expertise to help with first steps and consequently do not have such a policy. In these cases, the security requirements should be defined in business terms. This includes a description of the information or assets to be protected, their level of sensitivity, their relationship to the core business, and the requirement for maintaining the confidentiality, availability, and integrity of each.

One of the most common issues that surfaces from outsourcing arrangements is financial, wherein costs may not be fully understood or unanticipated costs arise after the fact. It is important that the organization understand the potential costs of the arrangement, which include a complete understanding of the internal costs before the outsourcing contract is established. A cost/benefit analysis should be performed and should include a calculation of return on investment. As with any cost/benefit analysis, there may be costs and benefits that are not quantifiable in financial terms, and these should be considered and included as well. These may include additional overhead in terms of staffing, financial obligations, and management requirements.

Outsourcing will add new risks to the corporate environment and may exacerbate existing risks. Many organizations that outsource perform a complete risk analysis before undertaking the arrangement, including a description of residual risk expected after the outsourcing project begins. Such an analysis can be invaluable during the process of preparing the formal specification, because it will point to the inclusion of requirements for ameliorating these risks. Because risk can be avoided or reduced by the implementation of risk management strategies, a full understanding of residual risk will also aid in managing the vendor's performance once the work begins; and it will suggest areas where management must pay stronger attention in assessing the project's success.

Prepare the Organization

To ensure the success of the outsourcing arrangement, the organization should be sure that it can manage the provider's work effectively. This requires internal corporate knowledge of the work or service outsourced. Even if this knowledge is not deeply technical — if, for example, the business is networking its services for the first time — the outsourcing organization must understand the business value of the work or service and how it supports the corporate mission. This includes an understanding of the internal cost structure because without this understanding, the financial value of the outsourcing arrangement cannot be assessed.

Assign Organizational Roles

As with any corporate venture, management and staff acceptance are important in ensuring the success of the outsourcing project. This can best be accomplished by involving all affected corporate staff in the decision-making process from the outset, and by ensuring that everyone is in agreement with, or is willing to support, the decision to go ahead.

With general support for the arrangement, the organization should articulate clearly each affected party's role in working with the vendor. Executives and management-level staff who are ultimately responsible for the

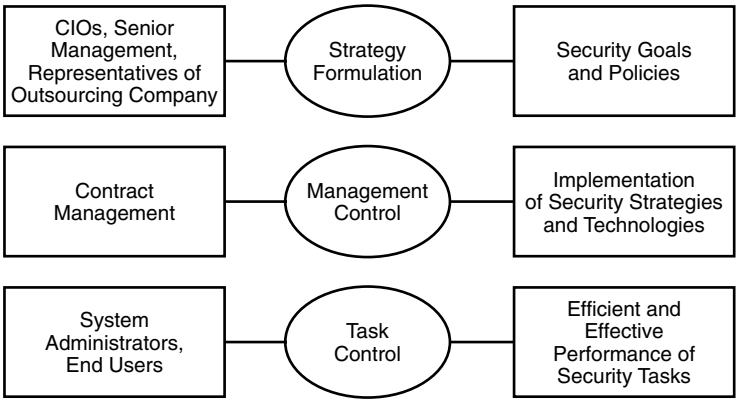


EXHIBIT 87.2 Management control for outsourcing contracts.

success of the arrangement must be supportive and must communicate the importance of the project's success throughout the organization. System owners and content providers must be helped to view the vendor as an IT partner and must not feel their ownership threatened by the assistance of an outside entity. These individuals should be given the responsibility for establishing the project's metrics and desired outcome because they are in the best position to understand what the organization's information requirements are.

The organization's IT staff is in the best position to gauge the vendor's technical ability and should be given a role in bringing the vendor up to speed on the technical requirements that must be met. The IT staff also should be encouraged to view the vendor as a partner in providing IT services to the organization's customers. And finally, if there are internal security employees, they should be responsible for establishing security policies and procedures to be followed by the vendor throughout the term of the contract.

The most important part of establishing organizational parameters is to assign accountability for the project's success. Although the vendor will be held accountable for the effectiveness of its work, the outsourcing organization should not give away accountability for management success. Where to lodge this accountability in the corporate structure is a decision that will vary based on the organization and its requirements, but the chances for success will be greatly enhanced by ensuring that those responsible for managing the effort are also directly accountable for its results.

A useful summary of organizational responsibilities for the outsourcing arrangement is shown in Exhibit 87.2, which illustrates the level of management control for various activities.[6]

Prepare a Specification and RFP

If the foregoing steps have been completed correctly, the process of documenting requirements and preparing a specification should be a simple formality. A well-written request for proposals (RFP) will include a complete and thorough description of the organizational, technical, management, and performance requirements and of the products and services to be provided by the vendor. Every corporate expectation that was articulated during the exploration stage should be covered by a performance requirement in the RFP. And the expected metrics that will be used to assess the vendor's performance should be included in a service level agreement (SLA). The SLA can be a separate document, but it should be legally incorporated into the resulting contract.

The RFP and resulting contract should specify the provisions for the use of hardware and software that are part of the outsourcing arrangements. This might include, for example, the type of software that is acceptable or its placement, so that the provider does not modify the client's technical infrastructure or remove assets from the customer premises without advance approval. Some MSSPs want to install their own hardware or software at the customer site; others prefer to use customer-owned technical resources; and still others perform on their own premises using their own resources. Regardless, the contract should spell out the provisions for ownership of all resources that support the arrangement and for the eventual return of any assets whose control or possession are outsourced. If there is intellectual property involved, as might be the case in a custom-developed security solution, the contract should also specify how the licensing of the property works and who will retain ownership of it at the end of the arrangement.

During the specification process, the organization should have determined what contractual provisions it will apply for nonperformance or substandard performance. The SLA contract should clearly define items considered to be performance infractions or errors, including requirements for correction of errors. This includes any financial or nonfinancial penalties for noncompliance or failure to perform.

The contract may not be restricted to technical requirements and contractual terms but may also consider human resources and business management issues. Some of the requirements that might be included govern access to vendor staff by the customer, and vice versa, and provisions for day-to-day management of the staff performing the work. In addition, requirements for written deliverables, regular reports, etc. should be specified in advance.

The final section of the RFP and contract should govern the end of the outsourcing arrangement and provisions for terminating the relationship with the vendor. The terms that govern the transition out should be designed to reduce exit barriers for both the vendor and the client, particularly because these terms may need to be invoked during a dispute or otherwise in less-than-optimal circumstances. One key provision will be to require that the vendor cooperates fully with any vendor that succeeds it in performance of the work.

Specify Financial Terms and Pricing

Some of the basic financial considerations for the RFP are to request that the vendor provide evidence that its pricing and terms are competitive and provide an acceptable cost/benefit business case. The RFP should request that the vendor propose incentives and penalties based on performance and warrant the work it performs.

The specific cost and pricing sections of the specification depend on the nature of the work outsourced. Historically, many outsourcing contracts were priced in terms of unit prices for units provided, and may have been measured by staff (such as hourly rates for various skill levels), resources (such as workstations supported), or events (such as calls answered). The unit prices may have been fixed or varied based on rates of consumption, may have included guaranteed levels of consumption, and may have been calculated based on cost or on target profits.

However, these types of arrangements have become less common over the past few years. The cost-per-unit model tends to cause the selling organization to try to increase the units sold, driving up the quantity consumed by the customer regardless of the benefit to the customer. By the same token, this causes the customer to seek alternative arrangements with lower unit costs; and at some point the two competing requirements diverge enough that the arrangement must end.

So it has become more popular to craft contracts that tie costs to expected results and provide incentives for both vendor and customer to perform according to expectations. Some arrangements provide increased revenue to the vendor each time a threshold of performance is met; others are tied to customer satisfaction measures; and still others provide for gain-sharing wherein the customer and vendor share in any savings from reduction in customer costs. Whichever model is used, both vendor and customer are given incentives to perform according to the requirements to be met by each.

Anticipate Legal Issues

The RFP and resulting contract should spell out clear requirements for liability and culpability. For example, if the MSSP is providing security alert and intrusion detection services, who is responsible in the event of a security breach? No vendor can provide a 100 percent guarantee that such breaches will not occur, and organizations should be wary of anyone who makes such a claim. However, it is reasonable to expect that the vendor can prevent predefined, known, and quantified events from occurring. If there is damage to the client's infrastructure, who is responsible for paying the cost of recovery? By considering these questions carefully, the client organization can use the possibility of breaches to provide incentives for the vendor to perform well.

In any contractual arrangement, the client is responsible for performing due diligence. The RFP and contract should spell out the standards of care that will be followed, and it will assign accountability for technical and management due diligence. This includes the requirements to maintain the confidentiality of protected information and for nondisclosure of sensitive, confidential, and secret information.

There may be legislative and regulatory issues that impact the outsourcing arrangement, and both the client and vendor should be aware of these. Organizations should be wary of outsourcing responsibilities for which it is legally responsible, unless it can legally assign these responsibilities to another party. In fact, outsourcing such services may be prohibited by regulation or law, particularly for government entities. Existing protections may not be automatically carried over in an outsourced environment. For example, certain requirements for

compliance with the Privacy Act or the Freedom of Information Act may not apply to employees of an MSSP or service provider.

Preparing a good RFP for security services is no different than preparing any RFP. The proposing vendors should be obligated to respond with clear, measurable responses to every requirement, including, if possible, client references demonstrating successful prior performance.

Phase 2: Select a Provider

During Phase 1, the organization defined the scope of work and the services to be outsourced. The RFP and specification were created, and the organization must now evaluate the proposals received and select a vendor. The process of selecting a vendor includes determining the appropriate characteristics of an outsourcing supplier, choosing a suitable vendor, and negotiating requirements and contractual terms.

Determine Vendor Characteristics

Among the most common security services outsourced are those that include installation, management, or maintenance of equipment and services for intrusion detection, perimeter scanning, VPNs and firewalls, and anti-virus and content protection. These arrangements, if successfully acquired and managed, tend to be long-term and ongoing in nature. However, shorter-term outsourcing arrangements might include testing and deployment of new technologies, such as encryption services and PKI in particular, because it is often difficult and expensive to hire expertise in these arenas. Hiring an outside provider to do one-time or short-term tasks such as security assessments, policy development and implementation, or audit, enforcement, and compliance monitoring is also becoming popular.

One factor to consider during the selection process is the breadth of services offered by the prospective provider. Some vendors have expertise in a single product or service that can bring superior performance and focus, although this can also mean that the vendor has not been able to expand beyond a small core offering. Other vendors sell a product or set of products, then provide ongoing support and monitoring of the offering. This, too, can mean superior performance due to focus on a small set of offerings; but the potential drawback is that the customer becomes hostage to a single technology and is later unable to change vendors. One relatively new phenomenon in the MSSP market is to hire a vendor-neutral service broker who can perform an independent assessment of requirements and recommend the best providers.

There are a number of terms that have become synonymous with outsourcing or that describe various aspects of the arrangement. *Insourcing* is the opposite of outsourcing, referring to the decision to manage services in-house. The term *midsourcing* refers to a decision to outsource a specific selection of services. *Smartsourcing* is used to mean a well-managed outsourcing (or insourcing) project and is sometimes used by vendors to refer to their set of offerings.

Choose a Vendor

Given that the MSSP market is relatively new and immature, organizations must pay particular attention to due diligence during the selection process, and should select a vendor that not only has expertise in the services to be performed but also shows financial, technical, and management stability. There should be evidence of an appropriate level of investment in the infrastructure necessary to support the service. In addition to assessing the ability of the vendor to perform well, the organization should consider less tangible factors that might indicate the degree to which the vendor can act as a business partner. Some of these characteristics are:

- *Business culture and management processes.* Does the vendor share the corporate values of the client? Does it agree with the way in which projects are managed? Will staff members be able to work successfully with the vendor's staff?
- *Security methods and policies.* Will the vendor disclose what these are? Are these similar to or compatible with the customer's?
- *Security infrastructure, tools, and technology.* Do these demonstrate the vendor's commitment to maintaining a secure environment? Do they reflect the sophistication expected of the vendor?
- *Staff skills, knowledge, and turnover.* Is turnover low? Does the staff appear confident and knowledgeable? Does the offered set of skills meet or exceed what the vendor has promised?
- *Financial and business viability.* How long has the vendor provided these services? Does the vendor have sufficient funding to remain in the business for at least two years?
- *Insurance and legal history.* Have there been prior claims against the vendor?

Negotiate the Arrangement

With a well-written specification, the negotiation process will be simple because expectations and requirements are spelled out in the contract and can be fully understood by all parties. The specific legal aspects of the arrangement will depend on the client's industry or core business, and they may be governed by regulation (for example, in the case of government and many financial entities). It is important to establish in advance whether the contract will include subcontractors, and if so, to include them in any final negotiations prior to signing a contract. This will avoid the potential inability to hold subcontractors as accountable for performance as their prime contractor.

Negotiation of pricing, delivery terms, and warranties should also be governed by the specification; and the organization should ensure that the terms and conditions of the specification are carried over to the resulting contract.

Phase 3: Manage the Arrangement

Once a provider has been selected and a contract is signed, the SLA will govern the management of the vendor. If the SLA was not included in the specification, it should be documented before the contract is signed and included in the final contract.

Address Performance Factors

For every service or resource being outsourced, the SLA should address the following factors:

- The expectations for successful service delivery (service levels)
- Escalation procedures
- Business impact of failure to meet service levels
- Turnaround times for delivery
- Service availability, such as for after-hours
- Methods for measurement and monitoring of performance

Use Metrics

To be able to manage the vendor effectively, the customer must be able to measure compliance with contractual terms and the results and benefits of the provider's work. The SLA should set a baseline for all items to be measured during the contract term. These will by necessity depend on which services are provided. For example, a vendor that is providing intrusion detection services might be assessed in part by the number of intrusions repelled as documented in IDS logs.

To motivate the vendor to behave appropriately, the organization must measure the right things — that is, results over which the provider has control. However, care should be taken to ensure that the vendor cannot directly influence the outcome of the collection process. In the example above, the logs should be monitored to ensure that they are not modified manually, or backup copies should be turned over to the client on a regular basis.

The SLA metrics should be reasonable in that they can be easily measured without introducing a burdensome data collection requirement. The frequency of measurement and audits should be established in advance, as should the expectations for how the vendor will respond to security issues and whether the vendor will participate in disaster recovery planning and rehearsals. Even if the provider is responsible for monitoring of equipment such as firewalls or intrusion detection devices, the organization may want to retain control of the incident response process, particularly if the possibility of future legal action exists. In these cases, the client may specify that the provider is to identify, but not act on, suspected security incidents. Thus, they may ask the provider for recommendations but may manage or staff the response process itself. Other organizations distinguish between internal and external threats or intrusions to avoid the possibility that an outside organization has to respond to incidents caused by the client's own employees.

Monitor Performance

Once the contract is in place and the SLA is active, managing the ongoing relationship with the service provider becomes the same as managing any other contractual arrangement. The provider is responsible for performing the work to specifications, and the client is responsible for monitoring performance and managing the contract.

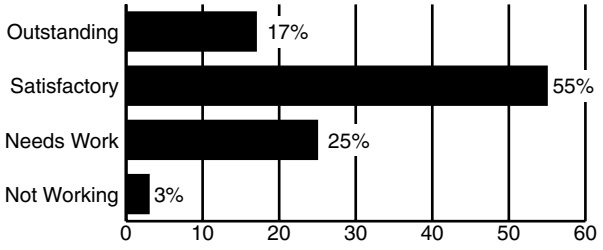


EXHIBIT 87.3 Customer satisfaction with security outsourcing.

Monitoring and reviewing the outsourced functions are critically important. Although the accountability for success of the arrangement remains with the client organization, the responsibility for monitoring can be a joint responsibility; or it can be done by an independent group inside or outside the organization.

Throughout the life of the contract, there should be clear single points of contact identified by the client and the vendor; and both should fully understand and support provisions for coordinating emergency response during a security breach or disaster.

Phase 4: Transition Out

In an ideal world, the outsourcing arrangement will continue with both parties to their mutual satisfaction. In fact, the client organization should include provisions in the contract for renewal, for technical refresh, and for adjustment of terms and conditions as the need arises. However, an ideal world rarely exists, and most arrangements end sooner or later. It is important to define in advance (in the contract and SLA) the terms that will govern the parties if the client decides to bring the work in-house or to use another contractor, along with provisions for penalties should either party not comply.

Should the arrangement end, the organization should continue to monitor vendor performance during the transition out. The following tasks should be completed to the satisfaction of both vendor and client:

- All property is returned to its original owner (with reasonable allowance for wear and tear).
- Documentation is fully maintained and up-to-date.
- Outstanding work is complete and documented.
- Data owned by each party is returned, along with documented settings for security controls. This includes backup copies.
- If there is to be staff turnover, the hiring organization has completed the hiring process.
- Requirements for confidentiality and nondisclosure continue to be followed.
- If legally required, the parties are released from any indemnities, warranties, etc.

Conclusion

The growth of the MSSP market clearly demonstrates that outsourcing of security services can be a successful venture both for the client and the vendor. While the market is undergoing some consolidation and refocusing as this chapter is being written, in the ultimate analysis, outsourcing security services is not much different than outsourcing any other IT service, and the IT outsourcing industry is established and mature. The lessons learned from one clearly apply to the other, and it is clear that organizations that choose to outsource are in fact applying those lessons. In fact, as Exhibit 87.3 shows, the majority of companies that outsource their security describe their level of satisfaction as outstanding or satisfactory.[7]

Outsourcing the security of an organization's information assets may be the antithesis of the ancient "security through obscurity" model. However, in today's networked world, with solid planning in advance, a sound rationale, and good due diligence and management, any organization can outsource its security with satisfaction and success.

References

1. Gary Kaiser, quoted by John Makulowich, in Government outsourcing, in *Washington Technol.*, 05/13/97; Vol. 12 No. 3, http://www.washingtontechnology.com/news/12_3/news/ 12940-1.html.
2. George Hulme, Security's best friend, *Information Week*, July 16, 2001, http://www.informationweek.com/story/IWK20010713S0009.
3. Jaikumar Vijayan, Outsources rush to meet security demand, *ComputerWorld*, February 26, 2001, http://www.computerworld.com/cwi/story/0,1199,NAV47_STO57980,00.html.
4. Chris King, META report: are managed security services ready for prime time?, *Datamation*, July 13, 2002, http://itmanagement.earthweb.com/secu/article/0,,11953_ 801181,00.html.
5. Glen Bruce and Rob Dempsey, *Security in Distributed Computing*, Hewlett-Packard Professional Books, Saddle River, NJ, 1997.
6. V. Govindarajan and R.N. Anthony, *Management Control Systems*, Irwin, Chicago, 1995
7. Forrester Research, cited in When Outsourcing the Information Security Program Is an Appropriate Strategy, at http://www.hyperon.com/outsourcing.htm.

88

Considerations for Outsourcing Security

Michael J. Corby, CISSP

Outsourcing computer operations is not a new concept. Since the 1960s, companies have been in the business of providing computer operations support for a fee. The risks and challenges of providing a reliable, confidential, and responsive data center operation have increased, leaving many organizations to consider retaining an outside organization to manage the data center in a way that the risks associated with these challenges are minimized.

Let me say at the onset that there is no one solution for all environments. Each organization must decide for itself whether to build and staff its own IT security operation or hire an organization to do it for them. This discussion will help clarify the factors most often used in making the decision of whether outsourcing security is a good move for your organization.

History of Outsourcing it Functions

Data Center Operations

Computer facilities have been traditionally very expensive undertakings. The equipment alone often cost millions of dollars, and the room to house the computer equipment required extensive and expensive special preparation. For that reason, many companies in the 1960s and 1970s seriously considered the ability to provide the functions of an IT (or EDP) department without the expense of building the computer room, hiring computer operators, and, of course, acquiring the equipment. Computer service bureaus and shared facilities sprang up to service the banking, insurance, manufacturing, and service industries. Through shared costs, these outsourced facilities were able to offer cost savings to their customers and also turn a pretty fancy profit in the process.

In almost all cases, the reasons for justifying the outsourcing decision were based on financial factors. Many organizations viewed the regular monthly costs associated with the outsource contract far more acceptable than the need to justify and depreciate a major capital expense.

In addition to the financial reasons for outsourcing, many organizations also saw the opportunity to offload the risk of having to replace equipment and software long before it had been fully depreciated due to increasing volume, software and hardware enhancements, and training requirements for operators, system programmers, and other support staff.

The technical landscape at the time was changing rapidly; there was an aura of special knowledge that was shared by those who knew how to manage the technology, and that knowledge was shared with only a few individuals outside the "inner circle."

Organizations that offered this service were grouped according to their market. That market was dictated by the size, location, or support needs of the customer:

- Size was measured in the number of transactions per hour or per day, the quantity of records stored in various databases, and the size and frequency of printed reports.

- Location was important because in the pre-data communications era, the facility often accepted transactions delivered by courier in paper batches and delivered reports directly to the customer in paper form. To take advantage of the power of automating the business process, quick turnaround was a big factor.
- The provider's depth of expertise and special areas of competence were also a factor for many organizations. Banks wanted to deal with a service that knew the banking industry, its regulations, need for detailed audits, and intense control procedures. Application software products that were designed for specific industries were factors in deciding which service could support those industries. In most instances, the software most often used for a particular industry could be found running in a particular hardware environment. Services were oriented around IBM, Digital, Hewlett-Packard, NCR, Burroughs, Wang, and other brands of computer equipment. Along with the hardware type came the technical expertise to operate, maintain, and diagnose problems in that environment. Few services would be able to support multiple brands of hardware.

Of course, selecting a data center service was a time-consuming and emotional process. The expense was still quite a major financial factor, and there was the added risk of putting the organization's competitive edge and customer relations in the hands of a third party. Consumers and businesses cowered when they were told that their delivery was postponed or that their payment was not credited because of a computer problem. Nobody wanted to be forced to go through a file conversion process and learn how to deal with a new organization any more than necessary. The ability to provide a consistent and highly responsive "look and feel" to the end customer was important, and the vendor's perceived reliability and long-term capabilities to perform in this area were crucial factors in deciding which service and organization would be chosen.

Contracting Issues

There were very few contracting issues in the early days of outsourced data center operations. Remember that almost all applications involved batch processing and paper exchange. Occasionally, limited file inquiry was provided, but price was the basis for most contract decisions.

If the reports could be delivered within hours or maybe within the same day, the service was acceptable. If there were errors or problems noted in the results, the obligation of the service was to rerun the process.

Computer processing has always been bathed in the expectation of confidentiality. Organizations recognized the importance of keeping their customer lists, employee ranks, financial operations, and sales information confidential; and contracts were respectful of that factor. If any violations of this expectation of confidentiality occurred in those days, they were isolated incidents that were dealt with privately, probably in the courts.

Whether processing occurred in a contracted facility or in-house, expectations that there would be an independent oversight or audit process were the same. EDP auditors focused on the operational behavior of servicer-designed specific procedures, and the expectations were usually clearly communicated. Disaster recovery planning, document storage, tape and disk archival procedures, and software maintenance procedures were reviewed and expected to meet generally accepted practices. Overall, the performance targets were communicated, contracts were structured based on meeting those targets, companies were fairly satisfied with the level of performance they were getting for their money, and they had the benefit of not dealing with the technology changes or the huge capital costs associated with their IT operations.

Control of Strategic Initiatives

The dividing line of whether an organization elected to acquire services of a managed data center operation or do it in-house was the control of their strategic initiatives. For most regulated businesses, the operations were not permitted to get too creative. The most aggressive organizations generally did not use the data center operations as an integral component of their strategy. Those who did deploy new or creative computer processing initiatives generally did not outsource that part of their operation to a shared service.

Network Operations

The decision to outsource network operations came later in the evolution of the data center. The change from a batch, paper processing orientation to an online, electronically linked operation brought about many of the same decisions that organizations faced years before when deciding to "build or buy" their computer facilities.

The scene began to change when organizations decided to look into the cost, technology, and risk involved with network operations. New metrics of success were part of this concept. Gone was the almost single focus on cost as the basis of a decision to outsource or develop an inside data communication facility. Reliability, culminating in the concept we now know as *continuous availability,* became the biggest reason to hire a data communications servicer. The success of the business often came to depend on the success of the data communications facility. Imagine the effect on today's banking environment if ATMs had a very low reliability, were fraught with security problems, or theft of cash or data. We frequently forget how different our personal banking was in the period before the proliferation of ATMs. A generation of young adults has been transformed by the direct ability to communicate electronically with a bank — much in the same way, years ago, that credit cards opened up a new relationship between consumers and retailers.

The qualification expected of the network operations provider was also very different from the batch-processing counterpart. Because the ability to work extra hours to catch up when things fell behind was gone, new expectations had to be set for successful network operators. Failures to provide the service were clearly and immediately obvious to the organization and its clients. Several areas of technical qualification were established.

One of the biggest questions used to gauge qualified vendors was bandwidth. How much data could be transmitted to and through the facility? This was reviewed on both a micro and macro domain. From the micro perspective, the question was, "How fast could data be sent over the network to the other end?" The higher the speed, the higher the cost. On a larger scale, what was the capacity of the network provider to transfer data over the 24-hour period? This included downtime, retransmissions, and recovery. This demand gave rise to the 24/7 operation, where staples of a sound operation like daily backups and software upgrades were considered impediments to the totally available network.

From this demand came the design and proliferation of the dual processor and totally redundant systems. Front-end processors and network controllers were designed to be failsafe. If anything happened to any of the components, a second copy of that component was ready to take over. For the most advanced network service provider, this included dual data processing systems at the back end executing every transaction twice, sometimes in different data centers, to achieve total redundancy.

Late delivery and slow delivery became unacceptable failures and would be a prime cause for seeking a new network service provider.

After the technical capability of the hardware/software architecture was considered, the competence of the staff directing the facility was considered. How smart, how qualified, how experienced were the people that ran and directed the network provider? Did the people understand the mission of the organization, and could they appreciate the need for a solid and reliable operation? Could they upgrade operating systems with total confidence? Could they implement software fixes and patches to assure data integrity and security? Could they properly interface with the applications software developers without requiring additional people in the organization duplicating their design and research capabilities?

In addition to pushing bits through the wires, the network service provider took on the role of the front-end manager of the organization's strategy. Competence was a huge factor in building the level of trust that executives demanded.

Along with this swing toward the strategic issues, organizations became very concerned about long-term viability. Often, huge companies were the only ones that could demonstrate this longevity promise. The mainframe vendor, global communications companies, and large well-funded network servicers were the most successful at offering these services universally. As the commerce version of the globe began to shrink, the most viable of these were the ones that could offer services in any country, any culture, at any time. The data communications world became a nonstop, "the store never closes" operation.

Contracting Issues

With this new demand for qualified providers with global reach came new demands for contracts that would reflect the growing importance of this outsourcing decision to the lifeblood of the organization.

Quality-of-service expectations were explicitly defined and put into contracts. Response time would be measured in seconds or even milliseconds. Uptime was measured in the number of nines in the percentage that would be guaranteed. Two nines, or 99 percent, was not good enough. Four nines (99.99 percent) or even five nines (99.999 percent) became the common expectation of availability.

A new emphasis developed regarding the extent to which data would be kept confidential. Questions were asked and a response expected in the contract regarding the access to the data while in transit. Private line networks were expected for most data communications facilities because of the perceived vulnerability of public telecommunications facilities. In some high-sensitivity areas, the concept of encryption was requested. Modems were developed that would encrypt data while in transit. Software tools were designed to help ensure unauthorized people would not be able to see the data sent.

Independent auditors reviewed data communications facilities periodically. This review expanded to include a picture of the data communications operation over time using logs and transaction monitors. Management of the data communication provider was frequently retained by the organization so it could attest to the data integrity and confidentiality issues that were part of the new expectations levied by the external regulators, reviewers, and investors. If the executives were required to increase security and reduce response time to maintain a competitive edge, the data communications manager was expected to place the demand on the outsourced provider.

Control of Strategic Initiatives

As the need to integrate this technical ability becomes more important to the overall organization mission, more and more companies opted to retain their own data communications management. Nobody other than the communications carriers and utilities actually started hanging wires on poles; but data communications devices were bought and managed by employees, not contractors. Alternatives to public networks were considered; microwave, laser, and satellite communications were evaluated in an effort to make sure that the growth plan was not derailed by the dependence on outside organizations.

The daily operating cost of this communications capability was large; but in comparison to the computer room equipment and software, the capital outlay was small. With the right people directing the data communications area, there was less need for outsourced data communications facilities as a stand-alone service. In many cases it was rolled into an existing managed data center; but in probably just as many instances, the managed data center sat at the end of the internally controlled data communications facility. The ability to deliver reliable communications to customers, constituents, providers, and partners was considered a key strategy of many forward-thinking organizations

Application Development

While the data center operations and data communications outsourcing industries have been fairly easy to isolate and identify, the application development outsourcing business is more subtle. First, there are usually many different application software initiatives going on concurrently within any large organization. Each of them has a different corporate mission, each with different metrics for success, and each with a very different user focus. Software customer relationship management is very different from software for human resources management, manufacturing planning, investment management, or general accounting.

In addition, outsourced application development can be carried out by general software development professionals, by software vendors, or by targeted software enhancement firms. Take, for instance, the well-known IBM manufacturing product Mapics®. Many companies that acquired the software contracted directly with IBM to provide enhancements; many others employed the services of software development organizations specifically oriented toward Mapics enhancements, while some simply added their Mapics product to the list of products supported or enhanced by their general application design and development servicer.

Despite the difficulty in viewing the clear picture of application development outsourcing, the justification was always quite clear. Design and development of new software, or features to be added to software packages, required skills that differed greatly from general data center or communications operations. Often, hiring the people with those skills was expensive and posed the added challenge in that designers were motivated by new creative design projects. Many companies did not want to pay the salary of good design and development professionals, train and orient them, and give them a one- or two-year design project that they would simply add to their resume when they went shopping for their next job.

By outsourcing the application development, organizations could employ business and project managers who had long careers doing many things related to application work on a variety of platforms and for a variety of business functions — and simply roll the coding or database expertise in and out as needed.

In many instances, also, outsourced applications developers were used for another type of activity — routine software maintenance. Good designers hate mundane program maintenance and start looking for new employment if forced to do too much of it. People who are motivated by the quick response and variety of tasks that can be juggled at the same time are well suited to maintenance tasks, but are often less enthusiastic about trying to work on creative designs and user-interactive activities where total immersion is preferred. Outsourcing the maintenance function is a great way to avoid the career dilemma posed by these conflicting needs. Y2K gave the maintenance programmers a whole new universe of opportunities to demonstrate their values. Aside from that once-in-a-millennium opportunity, program language conversions, operation system upgrades, and new software releases are a constant source of engagements for application maintenance organizations.

Qualifications for this type of service were fairly easy to determine. Knowledge of the hardware platform, programming language, and related applications were key factors in selecting an application development firm. Beyond those specifics, a key factor in selecting an application developer was in the actual experience with the specific application in question. A financial systems analyst or programmer was designated to work on financial systems; a manufacturing specialist on manufacturing systems, and so on.

Word quickly spread about which organizations were the application and program development leaders. Companies opened offices across the United States and around the world offering contract application services. Inexpensive labor was available for some programming tasks if contracted through international job shops, but the majority of application development outsourcing took place close to the organization that needed the work done.

Often, to ensure proper qualifications, programming tests were given to the application coders. Certifications and test-based credentials support extensive experience and intimate language knowledge. Both methods are cited as meritorious in determining the credentials of the technical development staff assigned to the contract.

Along with the measurable criteria of syntax knowledge, a key ingredient was the maintainability of the results. Often, one of the great fears was that the program code was so obscure that only the actual developer could maintain the result. This is not a good thing. The flexibility to absorb the application development at the time the initial development is completed or when the contract expires is a significant factor in selecting a provider. To ensure code maintainability, standards are developed and code reviews are frequently undertaken by the hiring organization.

Perhaps the most complicated part of the agreement is the process by which errors, omissions, and problems are resolved. Often, differences of opinion, interpretations of what is required, and the definition of things like "acceptable response time" and "suitable performance" were subject to debate and dispute. The chief way this factor was considered was in contacting reference clients. It probably goes to say that no application development organization registered 100 percent satisfaction with 100 percent of its customers 100 percent of the time. Providing the right reference account that gives a true representation of the experience, particularly in the application area evaluated, is a critical credential.

Contracting Issues

Application development outsourcing contracts generally took on two forms: pay by product or pay by production.

- Pay by product is basically the fixed-price contract; that is, hiring a developer to develop the product and, upon acceptance, paying a certain agreed amount. There are obvious derivations of this concept: phased payments, payment upon acceptance of work completed at each of several checkpoints — for example, payment upon approval of design concept, code completion, code unit testing, system integration testing, user documentation acceptance, or a determined number of cycles of production operation. This was done to avoid the huge balloon payment at the end of the project, a factor that crushed the cash flow of the provider and crippled the ability of the organization to develop workable budgets.
- Pay by production is the time-and-materials method. The expectation is that the provider works a prearranged schedule and, periodically, the hours worked are invoiced and paid. The presumption is that hours worked are productive and that the project scope is fixed. Failure of either of these factors most often results in projects that never end or exceed their budgets by huge amounts.

The control against either of these types of projects running amok is qualified approval oversight and audit. Project managers who can determine progress and assess completion targets are generally part of the organi-

zation's review team. In many instances, a third party is retained to advise the organization's management of the status of the developers and to recommend changes to the project or the relationship if necessary.

Control of Strategic Initiatives

Clearly the most sensitive aspect of outsourced service is the degree to which the developer is invited into the *inner sanctum* of the customer's strategic planning. Obviously, some projects such as Y2K upgrades, software upgrades, and platform conversions do not require anyone sitting in an executive strategy session; but they can offer a glimpse into the specifics of product pricing, engineering, investment strategy, and employee/partner compensation that are quite private. Almost always, application development contracts are accompanied by assurances of confidentiality and nondisclosure, with stiff penalties for violation.

Outsourcing Security

The history of the various components of outsourcing plays an important part in defining the security outsourcing business issue and how it is addressed by those seeking or providing the service. In many ways, outsourced security service is like a combination of the hardware operation, communications, and application development counterparts, all together. *Outsourced* is the general term; *managed security services* or MSS is the industry name for the operational component of an organization's total data facility, but viewed solely from the security perspective. As in any broad-reaching component, the best place to start is with a scope definition.

Defining the Security Component to be Outsourced

Outsourcing security can be a vast undertaking. To delineate each of the components, security outsourcing can be divided into six specific areas or domains:

1. Policy development
2. Training and awareness
3. Security administration
4. Security operations
5. Network operations
6. Incident response

Each area represents a significant opportunity to improve security, in increasing order of complexity. Let us look at each of these domains and define them a bit further.

Security Policies

These are the underpinning of an organization's entire security profile. Poorly developed policies, or policies that are not kept current with the technology, are a waste of time and space. Often, policies can work against the organization in that they invite unscrupulous employees or outsiders to violate the intent of the policy and to do so with impunity. The policies must be designed from the perspectives of legal awareness, effective communications skills, and confirmed acceptance on the part of those invited to use the secured facility (remember: unless the organization intends to invite the world to enjoy the benefits of the facility — like a Web site — it is restricted and thereby should be operated as a secured facility).

The unique skills needed to develop policies that can withstand the challenges of these perspectives are frequently a good reason to contract with an outside organization to develop and maintain the policies. Being an outside provider, however, does not lessen the obligation to intimately connect each policy with the internal organization. Buying the book of policies is not sufficient. They must present and define an organization's philosophy regarding the security of the facility and data assets. Policies that are strict about the protection of data on a computer should not be excessively lax regarding the same data in printed form. Similarly, a personal Web browsing policy should reflect the same organization's policy regarding personal telephone calls, etc. Good policy developers know this.

Policies cannot put the company in a position of inviting legal action but must be clearly worded to protect its interests. Personal privacy is a good thing, but using company assets for personal tasks and sending

correspondence that is attributed to the organization are clear reasons to allow some level of supervisory review or periodic usage auditing. Again, good policy developers know this.

Finally, policies must be clearly communicated, remain *apropos,* carry with them appropriate means for reporting and handling violations, and for being updated and replaced. Printed policy books are replaced with intranet-based, easily updated policies that can be adapted to meet new security demands and rapidly sent to all subject parties. Policy developers need to display a good command of the technology in all its forms — data communication, printed booklets, posters, memos, video graphics, and nontraditional means of bringing the policy to its intended audience's attention. Even hot air balloons and skywriting are fair game if they accomplish the intent of getting the policy across. Failure to know the security policy cannot be a defense for violating it. Selecting a security policy developer must take all of these factors into consideration.

Training and Awareness

Training and awareness are also frequently assigned to an outside servicer. Some organizations establish guidelines for the amount and type of training an employee or partner should receive. This can take the form of attending lectures, seminars, and conferences; reading books; enrolling in classes at local educational facilities; or taking correspondence courses. Some organizations will hire educators to provide specific training in a specific subject matter. This can be done using standard course material good for anyone, or it can be a custom-designed session targeted specifically to the particular security needs of the organization.

The most frequent topics of general education that anyone can attend are security awareness, asset protection, data classification, and recently, business ethics. Anyone at any level is usually responsible to some degree for ensuring that his or her work habits and general knowledge are within the guidance provided by this type of education. Usually conducted by the human resources department at orientation, upon promotion, or periodically, the objective is to make sure that everyone knows the baseline of security expectations. Each attendee will be expected to learn what everyone in the organization must do to provide for a secure operation. It should be clearly obvious what constitutes unacceptable behavior to anyone who successfully attends such training.

Often, the provider of this service has a list of several dozen standard points that are made in an entertaining and informative manner, with a few custom points where the organization's name or business mission is plugged into the presentation; but it is often 90 percent boilerplate.

Selecting an education provider for this type of training is generally based on their creative entertainment value — holding the student's attention — and the way in which students register their acknowledgment that they have heard and understood their obligations. Some use the standard signed acknowledgment form; some even go so far as to administer a digitally signed test. Either is perfectly acceptable but should fit the corporate culture and general tenor.

Some additional requirements are often specified in selecting a training vendor to deal with technical specifics. Usually some sort of hands-on facility is required to ensure that the students know the information and can demonstrate their knowledge in a real scenario. Most often, this education will require a test for mastery or even a supervised training assignment. Providers of this type of education will often provide these services in their own training center where equipment is configured and can be monitored to meet the needs of the requesting organization.

Either in the general or specific areas, organizations that outsource their security education generally elect to do a bit of both on an annual basis with scheduled events and an expected level of participation. Evaluation of the educator is by way of performance feedback forms that are completed by all attendees. Some advanced organizations will also provide metrics to show that the education has rendered the desired results — for example, fewer password resets, lost files, or system crashes.

Security Administration

Outsourcing security administration begins to get a bit more complicated. Whereas security policies and security education are both essential elements of a security foundation, security administration is part of the ongoing security "face" that an organization puts on every minute of every day and requires a higher level of expectations and credentials than the other domains.

First, let us identify what the security administrator is expected to do. In general terms, security administration is the routine adds, changes, and deletes that go along with authorized account administration. This can include verification of identity and creation of a subsequent authentication method. This can be a password,

token, or even a biometric pattern of some sort. Once this authentication has been developed, it needs to be maintained. That means password resets, token replacement, and biometric alternative (this last one gets a bit tricky, or messy, or both).

Another significant responsibility of the security administrator is the assignment of approved authorization levels. Read, write, create, execute, delete, share, and other authorizations can be assigned to objects from the computer that can be addressed down to the data item if the organization's authorization schema reaches that level. In most instances, the tools to do this are provided to the administrator, but occasionally there is a need to devise and manage the authority assignment in whatever platform and at whatever level is required by the organization.

A major responsibility of security administrators that is often overlooked is reporting their activities. If a security policy is to be deemed effective, the workload should diminish over time if the population of users remains constant. I once worked with an organization that had outsourced the security administration function and paid a fee based on the number of transactions handled. Interestingly, there was an increasing frequency of reassignment of authorizations, password resets, and adds, changes, and deletes as time went on. The rate of increase was double the rate of user population expansion. We soon discovered that the number of user IDs mushroomed to two or three times the total number of employees in the company. What is wrong with that picture? Nothing if you are the provider, but a lot if you are the contracting organization.

The final crucial responsibility of the security administrator is making sure that the procedures designed to assure data confidentiality, availability, and integrity are carried out according to plan. Backup logs, incident reports, and other operational elements — although not exactly part of most administrators' responsibilities — are to be monitored by the administrator, with violations or exceptions reported to the appropriate person.

Security Operations

The security operations domain has become another recent growth area in terms of outsourced security services. Physical security was traditionally separate from data security or computer security. Each had its own set of credentials and its own objectives. Hiring a company that has a well-established physical security reputation does not qualify them as a good data security or computer security operations provider. As has been said, "Guns, guards, and dogs do not make a good data security policy;" but recently they have been called upon to help. The ability to track the location of people with access cards and even facial recognition has started to blend into the data and operational end of security so that physical security is vastly enhanced and even tightly coupled with security technology.

Many organizations, particularly since September 11, have started to employ security operations specialists to assess and minimize the threat of physical access and damage in many of the same terms that used to be reserved only for data access and computer log-in authentication.

Traditional security operations such as security software installation and monitoring (remember ACF2, RACF, Top Secret, and others), disaster recovery and data archival (Comdisco, Sunguard, Iron Mountain, and others), and a whole list of application-oriented control and assurance programs and procedures have not gone away. Skills are still required in these areas, but the whole secure operations area has been expanded to include protection of the tangible assets as well as the data assets. Watch this area for more developments, including the ability to use the GPS location of the input device, together with the location of the person as an additional factor in transaction authentication.

Network Operations

The most recent articles on outsourcing security have looked at the security of the network operations as the most highly vulnerable and therefore the most sensitive of the security domains. Indeed, much work has been done in this area, and industry analysts are falling over themselves to assess and evaluate the vendors that can provide a managed security operation center, or SOC.

It is important to define the difference between a *network* operation center (NOC) and a *security* operation center (SOC). The difference can be easily explained with an analogy. The NOC is like a pipe that carries and routes data traffic to where it needs to go. The pipe must be wide enough in diameter to ensure that the data is not significantly impeded in its flow. The SOC, on the other hand, is not like the pipe but rather like a window in the pipe. It does not need to carry the data, but it must be placed at a point where the data flowing through the pipe can be carefully observed. Unlike the NOC, which is a constraint if not *wide* enough, the SOC will not be able to observe the data flow carefully enough if it is not *fast* enough.

Network operations have changed from the earlier counterparts described previously in terms of the tools and components that are used for function. Screens are larger and flatter. Software is more graphically oriented.

Hardware is quicker and provides more control than earlier generations of the NOC, but the basic function is the same.

Security operation centers, however, are totally new. In their role of maintaining a close watch on data traffic, significant new software developments have been introduced to stay ahead of the volume. This software architecture generally takes two forms: data compression and pattern matching.

- *Data compression* usually involves stripping out all the inert traffic (which is usually well over 90 percent) and presenting the data that appears to be *interesting* to the operator. The operator then decides if the interesting data is problematic or indicative of a security violation or intrusion attempt, or whether it is simply a new form of routine inert activity such as the connection of a new server or the introduction of a new user.
- *Pattern matching* (also known as data modeling) is a bit more complex and much more interesting. In this method, the data is fit to known patterns of how intrusion attempts are frequently constructed. For example, there may be a series of pings, several other probing commands, followed by a brief period of analysis, and then the attempt to use the data obtained to gain access or cause denial of service. In its ideal state, this method can actually predict intrusions before they occur and give the operator or security manager a chance to take evasive action.

Most MSS providers offer data compression, but the ones that have developed a comprehensive pattern-matching technique have more to offer in that they can occasionally predict and prevent intrusions — whereas the data compression services can, at best, inform when an intrusion occurs.

Questions to ask when selecting an MSS provider include first determining if they are providing a NOC or SOC architecture (the pipe or the window). Second, determine if they compress data or pattern match. Third, review very carefully the qualifications of the people who monitor the security. In some cases they are simply a beeper service. ("Hello, Security Officer? You've been hacked. Have a nice day. Goodbye.") Other providers have well-trained incident response professionals who can describe how you can take evasive action or redesign the network architecture to prevent future occurrences.

There are several cost justifications for outsourcing security operations:

- The cost of the data compression and modeling tools is shared among several clients.
- The facility is available 24/7 and can be staffed with the best people at the most vulnerable time of day (nights, weekends, and holidays).
- The expensive technical skills that are difficult to keep motivated for a single network are highly motivated when put in a position of constant activity. This job has been equated to that of a military fighter pilot: 23 hours and 50 minutes of total boredom followed by ten minutes of sheer terror. The best operators thrive on the terror and are good at it.
- Patterns can be analyzed over a wide range of address spaces representing many different clients. This allows some advanced warning on disruptions that spread (like viruses and worms), and also can be effective in finding the source of the disruption (perpetrator).

Incident Response

The last area of outsourced security involves the response to an incident. A perfectly legitimate and popular strategy is that every organization will at some time experience an incident. The ones that successfully respond will consider that incident a minor event. The ones that fail to respond or respond incorrectly can experience a disaster. Incident response involves four specialties:

1. Intrusion detection
2. Employee misuse
3. Crime and fraud
4. Disaster recovery

Intrusion Detection

Best depicted by the previous description of the SOC, intrusion detection involves the identification and isolation of an intrusion attempt. This can be either from the outside, or, in the case of server-based probes, can identify attempts by authorized users to go to places they are not authorized to access. This includes placing sensors (these can be certain firewalls, routers, or IDSs) at various points in the network and having those

sensors report activity to a central monitoring place. Some of these devices perform a simple form of data compression and can even issue an e-mail or dial a wireless pager when a situation occurs that requires attention.

Employee Misuse

Many attempts to discover employee abuse have been tried over the last several years, especially since the universal acceptance of Internet access as a staple of desktop appliances. Employees have been playing "cat and mouse" with employers over the use of the Internet search capabilities for personal research, viewing pornography, gift shopping, participation in unapproved chat rooms, etc. Employers attempt to monitor their use or prevent such use with filters and firewalls, and employees find new, creative ways to circumvent the restriction. In the United States, this is a game with huge legal consequences. Employees claim that their privacy has been violated; employers claim the employee is wasting company resources and decreasing their effectiveness. Many legal battles have been waged over this issue.

Outsourcing the monitoring of employee misuse ensures that independently defined measures are used across the board for all employees in all areas and at all levels. Using proper techniques for evidence collection and corroboration, the potential for successfully trimming misuse and dismissal or punishment of offenders can be more readily ensured.

Crime and Fraud

The ultimate misuse is the commission of a crime or fraud using the organization's systems and facilities. Unless there is already a significant legal group tuned in to prosecuting this type of abuse, almost always the forensic analysis and evidence preparation are left to an outside team of experts. Successfully identifying and prosecuting or seeking retribution from these individuals depends very heavily on the skills of the first responder to the situation.

Professionals trained in data recovery, forensic analysis, legal interviewing techniques, and collaboration with local law enforcement and judiciary are crucial to achieving success by outsourcing this component.

Disaster Recovery

Finally, one of the oldest security specialties is in the area of disaster recovery. The proliferation of backup data centers, records archival facilities, and site recovery experts have made this task easier; but most still find it highly beneficial to retain outside services in several areas:

- *Recovery plan development:* including transfer and training of the organization's recovery team
- *Recovery plan test:* usually periodic with reports to the executives and, optionally, the independent auditors or regulators
- *Recovery site preparation:* retained in advance but deployed when needed to ensure that the backup facility is fully capable of accepting the operation and, equally important, that the restored original site can resume operation as quickly as possible

All of these functions require special skills for which most organizations cannot justify full-time employment, so outsourcing these services makes good business sense. In many cases, the cost of this service can be recovered in reduced business interruption insurance premiums. Look for a provider that meets insurance company specifications for a risk class reduction.

Establishing the Qualifications of the Provider

For all these different types of security providers, there is no one standard measure of their qualifications. Buyers will need to fall back on standard ways to determine their vendor of choice. Here are a few important questions to ask that may help:

- What are the skills and training plan of the people actually providing the service?
- Is the facility certified under a quality or standards-based program (ISO 9000/17799, BS7799, NIST Common Criteria, HIPAA, EU Safe Harbors, etc.)?
- Is the organization large enough or backed by enough capital to sustain operation for the duration of the contract?
- How secure is the monitoring facility (for MSS providers)? If anyone can walk through it, be concerned.
- Is there a redundant monitoring facility? Redundant is different from a follow-the-sun or backup site in that there is essentially no downtime experienced if the primary monitoring site is unavailable.

- Are there SLAs (service level agreements) that are acceptable to the mission of the organization? Can they be raised or lowered for an appropriate price adjustment?
- Can the provider do all of the required services with its own resources, or must the provider obtain third-party subcontractor agreements for some components of the plan?
- Can the provider prove that its methodology works with either client testimonial or anecdotal case studies?

Protecting Intellectual Property

Companies in the security outsourcing business all have a primary objective of being a critical element of an organization's trust initiative. To achieve that objective, strategic information may very likely be included in the security administration, operation, or response domains. Protecting an organization's intellectual property is essential in successfully providing those services. Review the methods that help preserve the restricted and confidential data from disclosure or discovery.

In the case of incident response, a preferred contracting method is to have a pre-agreed contract between the investigator team and the organization's attorney to conduct investigations. That way, the response can begin immediately when an event occurs without protracted negotiation, and any data collected during the investigation (i.e., password policies, intrusion or misuse monitoring methods) are protected by attorney–client privilege from subpoena and disclosure in open court.

Contracting Issues

Contracts for security services can be as different as night is to day. Usually when dealing with security services, providers have developed standard terms and conditions and contract prototypes that make sure they do not commit to more risk than they can control. In most cases there is some "wiggle room" to insert specific expectations, but because the potential for misunderstanding is high, I suggest supplementing the standard contract with an easy-to-read memo of understanding that defines in as clear a language as possible what is included and what is excluded in the agreement. Often, this clear intent can take precedence over "legalese" in the event of a serious misunderstanding or error that could lead to legal action.

Attorneys are often comfortable with one style of writing; technicians are comfortable with another. Neither is understandable to most business managers. Make sure that all three groups are in agreement as to what is going to be done at what price.

Most activities involve payment for services rendered, either time and materials (with an optional maximum), or a fixed periodic amount (in the case of MSS).

Occasionally there may be special conditions. For example, a prepaid retainer is a great way to ensure that incident response services are deployed immediately when needed. "Next plane out" timing is a good measure of immediacy for incident response teams that may need to travel to reach the site. Obviously, a provider with a broad geographic reach will be able to reach any given site more easily than the organization with only a local presence. Expect a higher rate for court testimony, immediate incident response, and evidence collection.

Quality of Service Level Agreements

The key to a successfully managed security agreement lies in negotiating a reasonable service level agreement. Response time is one measure. Several companies will give an expected measure of operational improvement, such as fewer password resets, reduced downtime, etc. Try to work out an agreeable set of QoS factors and tie a financial or an additional time penalty for response outside acceptable parameters. Be prudent and accept what is attainable, and do not try to make the provider responsible for more than it can control. Aggressively driving a deal past acceptable criteria will result in no contract or a contract with a servicer that may fail to thrive.

Retained Responsibilities

Despite what domain of service is selected or the breadth of activities that are to be performed, there are certain cautions regarding the elements that should be held within the organization if at all possible.

Management

The first of these is management. Remember that management is responsible for presenting and determining the culture of the organization. Internal and external expectations of performance are almost always carried forth by management style, measurement, and communications, both formal and informal. Risk of losing that culture or identity is considerably increased if the management responsibility for any of the outsourced functions is not retained by someone in the organization ultimately accountable for their performance. If success is based on presenting a trusted image to partners, customers, and employees, help to ensure that success by maintaining close control over the management style and responsibility of the services that are acquired.

Operations

Outsourcing security is not outsourcing business operation. There are many companies that can help run the business, including operating the data center, the financial operations, legal, shipping, etc. The same company that provides the operational support should not, as a rule, provide the security of that operation. Keep the old *separation of duties* principle in effect. People other than those who perform the operations should be selected to provide the security direction or security response.

Audit and Oversight

Finally, applying the same principle, invite and encourage frequent audit and evaluation activities. Outsourced services should always be viewed like a yoyo. Whenever necessary, an easy pull on the string should be all that is necessary to bring them back into range for a check and a possible redirection. Outsourcing security or any other business service should not be treated as a "sign the contract and forget it" project.

Building an Escape Clause

But what if all this is done and it still looks like we made a mistake? Easy. If possible, build in an escape clause in the outsource contract that allows for a change in scope, direction, or implementation. If these changes (within reason) cannot be accommodated, most professional organizations will allow for an escape from the contract. Setup and equipment charges may be incurred, but those would typically be small compared to the lost time and expense involved in misunderstanding or hiring the wrong service. No security service organization wants a reference client that had to be dragged, kicking and screaming, through a contract simply because the name is on the line when everyone can agree that the service does not fit.

The Future of Outsourced Security

Industries Most Likely to Outsource

The first category of industries most likely to outsource security is represented by those companies whose key assets are the access to reliable data or information service. Financial institutions, especially banks, securities brokers, and insurance, health, or property claims operations, are traditional buyers of security services.

Recent developments in privacy have added healthcare providers and associated industries to that list. Hospitals, medical care providers, pharmaceuticals, and health-centered industries have a new need for protecting the privacy of personal health information. Reporting on the success of that protection is often a new concept that neither meets the existing operation nor justifies the full-time expense. HIPAA compliance will likely initiate a rise in the need for security (privacy) compliance providers.

The third category of industry that frequently requires outsourced security is the set of industries that cannot suffer any downtime or show any compromise of security. Railroads, cargo ships, and air traffic control are obvious examples of the types of industries where continuous availability is a crucial element for success. They may outsource the network operation or periodic review of their response and recovery plan. Internet retailers that process transactions with credit cards or against credit accounts fit into this category. Release of credit card data, or access to or changes made to purchasing history, is often fatal to continued successful operation.

The final category of industry that may need security services are those industries that have as a basis of their success an extraordinary level of trust in the confidentiality of their data. Taken to the extreme, this can include military or national defense organizations. More routinely, this would include technology research, legal, marketing, and other industries that would suffer severe image loss if it were revealed that their security was compromised or otherwise rendered ineffectual.

Measurements of Success

I once worked on a fairly complex application project that could easily have suffered from "scope creep." To offset this risk, we encouraged the user to continually ask the team, "How do we know we are done?" This simple question can help identify quite clearly what the expectations are for the security service, and how success is measured. What comes to my mind is the selection of the three milestones of project success: "scope, time, and cost — pick two out of three." A similar principle applies to measuring the success of security services. They are providing a savings of risk, cost, or effort. Pick two out of three. It is impractical to expect that everything can be completely solved at a low cost with total confidence. Security servicers operate along the same principles. They can explain how you can experience success, but only in two out of three areas. Either they save money, reduce risk, or take on the complexity of securing the enterprise. Only rarely can they do all three. Most can address two of these measures, but it lies to the buying organization to determine which of these are the two most important.

Response of MSS (Managed Security Service) Providers to New World Priorities

After September 11, 2001, the security world moved substantially. What was secure was no longer secure. What was important was no longer important. The world focused on the risk of personal safety and physical security and anticipated the corresponding loss of privacy and confidentiality. In the United States, the constitutional guarantee of freedom was challenged by the collective need for personal safety, and previously guaranteed rights were brought into question.

The security providers have started to address physical safety issues in a new light. What was previously deferred to the physical security people is now accepted as part of the holistic approach to risk reduction and trust. Look for an integration of traditional physical security concepts to be enhanced with new technologies like digital facial imaging, integrated with logical security components. New authentication methods will reliably validate "who did what where," not only when something was done on a certain device.

Look also for an increase in the sophistication of pattern matching for intrusion management services. Data compression can tell you faster that something has happened, but sophisticated modeling will soon be able to predict with good reliability that an event is forming in enough time to take appropriate defensive action.

We will soon look back on today as the primitive era of security management.

Response of the MSS Buyers to New World Priorities

The servicers are in business to respond quickly to new priorities, but managed security service buyers will also respond to emerging priorities. Creative solutions are nice, but practicality demands that enhanced security be able to prove itself in terms of financial viability.

I believe we will see a new emphasis on risk management and image enhancements. Organizations have taken a new tack on the meaning of *trust* in their industries. Whether it is confidentiality, accuracy, or reliability, the new mantra of business success is the ability to depend on the service or product that is promised. Security in all its forms is key to delivering on that promise.

Summary and Conclusions

Outsourced security, or managed security services (MSS), will continue to command the spotlight. Providers of these services will be successful if they can translate technology into real business metrics. Buyers of that service will be successful if they focus on the measurement of the defined objectives that managed services can provide. Avoid the attraction offered simply by a recognized name and get down to real specifics.

Based on several old and tried methods, there are new opportunities to effectively use and build on the skills and economies of scale offered by competent MSS providers. Organizations can refocus on what made them viable or successful in the first place: products and services that can be trusted to deliver on the promise of business success.

89

Outsourcing Security

James S. Tiller, CISA, CISSP

Unquestionably, security is complex. Whether one likes it or not, agrees or not, security permeates every aspect of today's business — security can, and does, exist at every layer within an environment. From physical security in the form of locks, barbed wire, metal detectors, and exotic plants, such as the formidable Dendrocnide,[1] to social and cultural demands on security operations, security — or the lack thereof — is everywhere. Given the convoluted reality of security, managing the required aspects of security can become overwhelming for many organizations, not to mention costly.

Planning, creating, and managing the various characteristics of security, which may include technology, operations, policy, communications, and legalese, requires a great deal of experience, time, and investment — investment in technology as well as people, development, and organizational commitment to the security posture defined and sanctioned through accepted policies and procedures. Unfortunately, it is difficult to associate these investments to actual returns. Yes, security can provide cost savings when planned and integrated compressively, but seldom has a direct impact on revenue for traditional businesses. This can be attributed to several reasons. Large, diverse firms that have complicated financial structures introduce a level of difficultly in pinning down a monetary return on security-related investments. On the other end of the spectrum, small companies operate on margins that are sensitive to business elements that have difficultly realizing measurable advantages through information security. For example, a bolt and nut manufacturer makes $0.0001 on each bolt and has the potential to lose substantial revenue if quality management misses a crossed thread on a batch of 100,000 units. Where does information security fit given that risk? For many, security is seen as an insurance policy; a risk mitigation contract written by technologists for business managers to ensure the stability of the network during an attack, or its resistance to attacks. Although this is not entirely true — security can be a differentiating business enabler — the fact remains that the majority of business owners view security as a cost of doing business.

Security, or insurance, is a nonprofit generating part of business (unless you are an insurance company) and represents the cost of mitigating one's exposure to threats — fire, hurricane, flood, hackers, etc. Additionally, security can require huge implementation costs, but that is only the beginning. Supporting and managing the constant updates, service packs, and patches, combined with the continual monitoring of logs, reports, and vulnerability warnings, are simply too much for many businesses.

Imagine a medium-sized company that designs, produces, and sells boats. This company might use the Internet for market research, VPNs to suppliers and resellers, and commodity management to make sure it is getting the best price for resin and fiberglass. With the sharing of critical logistics and financial information, this company is at risk without a sound security solution to control, or at least maintain, awareness of the threats to its business' success. However, the cost of secure operations may simply outweigh the risks; therefore, security becomes something of limited focus to the company. Boat manufacturing can be very competitive and mistakes are costly; the last thing the company may want to do is invest in people to manage its security, for which there is no foreseeable financial justification related directly to making boats better and faster. They know they need it, but today's technology, threats, and limitless exposures are sometimes too great to fully digest and make critical investment decisions that may impact the business for years to come.

Simply stated, businesses have difficultly rationalizing the costs of security controls where there is little or no measurable effect on the direct revenue-generating dealings of their core business. In many cases, security

is not ignored. A firewall is installed, configured based on the implementer's knowledge of operations passed down, and then left to rot on the technical vine.

Enter the security provider — typically referred to as a Managed Security Service Provider (MSSP) — an organization that assumes the responsibility of managing a company's security. Of course, there are several variations on this theme and each is fraught with its own share of complexities, advantages, and costs. This chapter investigates the role of MSSPs, the various solutions that can be found, and the implications of leveraging them for outsourcing security. Additionally, it is assumed that the focal audience is the traditional enterprise organization. For businesses that rely heavily on E-business between organizations, partners, and customers, the use of managed security is exponentially more involved and proportional to the criticality of E-business to the core revenue-generating functions.

The chapter continues by investigating the role security plays in business, the implications of technology, company culture, the commodity security has become, and outsourcing's involvement. Finally, this chapter discusses how outsourcing security can be a double-edged sword depending on how security is viewed within an organization — an enabler or an insurance policy.

The Business of Security

In the beginning, when security was a router with an access control list, it was much more simple to point to technology as the answer. Security practitioners at the time knew the threats to business were nothing that had ever been seen in traditional networks prior to the adoption of the Internet. However, with the neck-breaking pace of technology advancement and adoption, getting companies to simply acknowledge the massive threats presented by the Internet was difficult, much less getting them to invest in proper security management. At the time security became the inhibiter of technology and the Internet just when organizations were looking to expand their use of capabilities the Internet promised. It took time and a couple of legendary attacks, but many companies began to see the value of security. This is when the firewall was born; a system that one could point to and use to communicate one's organization's commitment to sound security — technology appeared to be the answer that was truly tangible. Of course, firewalls became larger, more complicated, and introduced dynamics into the infrastructure that were typically the result of demands for greater access to Internet resources in a secure manner. It became a give-and-take between functionality and security, for which we have yet to truly evolve.

Today, administrators, management, and entire companies are coming to the realization that security is much more than technology — albeit that technology is a critical and necessary component of a security program. It is fair to say that without security technology we would have little hope of realizing anything that could be mildly confused with information security. However, technology is only one of the many components of a secure posture and today that technology has become the focal point for management and comes with a substantial price. The investment in security technology, once again, is difficult to apply to the realization of true revenue — or even, in some cases, with cost savings. Cost savings are typically associated with streamlining a process to make it more efficient, therefore saving money. Whereas security processes do not have the luxury of being considered time-saving, rather the contrary is typically viewed of security. It is important to also recognize that traditional security measures do not "make money" and are usually associated with cost incurred for simply doing business in the Internet age — a toll for the information highway.

Security technology has become the focus for many companies, and requires investment, time, and constant management, but it remains difficult to justify to the CFO responsible for stock valuation. Security technology has become a commodity: something to sell or trade, or outside the realm of your primary focus, but a critical necessity to your survival. Therefore, pay someone else to deal with it but remain conscious of the risks. Security may not be one's core business, but the lack of security will become the core concern when an incident occurs.

A Judgment Call

Security, as mentioned above, can be as much of an enabler as a disabler of business, depending on the perspective of the decision makers. Defining risk is complicated. Determining what is of value to the business weighted against the perception of value to your customers. For a research organization, the decision is relatively simple — protect the proprietary data and invest in a security program that is relatively parallel to the tangible value placed on the information and its confidentiality.

On the other hand, one may allow an attack because the security breach will cause less damage in the short term than stopping all services that are providing $100,000 worth of transactions an hour. This is where business meets security. Generating revenue may be more important than the impacts the attack will have on the immediate term. This is seen in some E-commerce sites and the exposure of credit cards. To stop the attack and fix the hole may cause service disruption, leading to huge losses in revenue. Unfortunately for organizations that make this determination, they usually end up paying in the long run through loss of credibility.

It is necessary for any organization to truly investigate their perception and culture of security to realize the proportional inclusion of outsourcing security and the depth (or business impact) of that service.

To accomplish this, a risk analysis must be performed to identify digital assets, their value, the threats to those assets, and the impact of loss if the opposing threats were to be exploited. By performing an analysis, the organization can create multiple levels of security associated with different types and forms of data, ultimately defining proportional measures for controls. Once a risk is identified and measured against the impacts, the cost of the loss can be compared to the cost of remediation. It should not be immediately assumed that if the cost of remediation is greater than what the threat represents, the risk is simply accepted. Other risks and benefits can be realized by an investment originally destined to accommodate a single risk. Conversely, it also cannot be assumed that a risk will be mitigated when the associated costs are much less than the possible loss. Nevertheless, when a risk is identified and costs are determined, there must be a decision to address, accept, or transfer the risk. *Addressing* the risk is deciding to take action, either by people, technology, money, relocation, or anything that will mitigate the risk. *Accepting* risk is simply assuming the risk presented and hoping that one does not fall victim to an exploitation. Finally, *transferring* risk is where MSSPs come into play. By investing in another firm whose core business provides the protection one needs to cover those risks that are beyond the core focus, one achieves true insurance. Car insurance pays the tens of thousands of dollars that one would have to pay in the event of an accident. The cost of transferring the financial risk is a monthly payment to the insurance firm.

The Segmentation of Security

Security is primarily associated with technology — firewalls, intrusion detection systems, scanners, content filters, etc. — and rightfully so; this is to be expected. Technology is the tool by which we can realize digital information security; however, the security provided by technology is only as good as its owner. The people who plan, design, implement, support, and use the technology have to appreciate the security tools by understanding their role in the complex web of a security program and use them accordingly. Otherwise, the reality of security is lost and only a feeling of security remains. A firewall may provide ample security when it is implemented; but as each second passes, more vulnerabilities and exploits appear, requiring tuning and changes on the firewall to accommodate the dynamics of the environment. A firewall is a very simple example, but apply the analogy to all the security solutions, policies, organization, procedures, etc., and you get a very challenging proposition.

Why is this an important topic? If an organization embraces the concept that security technology can be a commodity and ultimately maintained by someone else, it will release the organization to focus on its business and the other side of security — culture. Culture is the use and understanding of security in our actions within the framework of business objectives. It is accepting security processes into the business process — where it should reside. Ultimately, the result is that the part of security that can consume time, money, and attention is left to others, while other portions of security (which could certainly include other versions of security technology combined with culture) do not burden organization personnel, so that they are free to enable the business to be more competitive in their industry.

Essentially, when outsourcing security, one must determine the scope of the involvement with the provider, what is expected, the relationship, and the depth the outsourcer needs to be within one's company. It is up to the company to determine what it considers the commodity and then associate that against services offered. For many organizations, MSSPs provide the "holy grail" of security solutions, the proverbial monkey off their back. For others, it represents the ultimate exposure of privacy and the inclusion of an unknown in their deepest inner workings.

Risk Management

As soon as you have anything of value, you are at risk of losing it — it is just that simple. Additionally, the risk is not always proportional to the perception of value by the owner. For example, you may have a junk car that barely gets you to the store, and life with the car is nearing greater pain with than without. However, someone who does not have a car — or the option to buy one — sees not only potential in obtaining something you spend little in protecting, but could use it to get something of greater value.

Risk is a measure of the loss of what you consider valuable, the impact of losing it, the threats to those assets, and how often those threats could be successful. Managing risk is continually reinvestigating and adjusting these measurements in accordance with business changes and the dynamics of technology and the environment.

Volcanoes are a formidable threat and the risk to your assets is directly proportional to the proximity of the volcano. You can mitigate this risk in several ways, each with its own costs. Build a firewall to slow the lava and buy time to escape with your assets; do not keep all your assets near the volcano, or move farther away. However, what is the potential of the volcano erupting? Every millennium or so, Mt. Vesuvius may erupt — so what is the real risk, and what investment should you make given these variables?

The moral of the examples is that you must determine what is of value to you, weigh it against the exposure to threats, and make an informed decision on how to mitigate the vulnerability. Performing a risk analysis is critical in determining if outsourcing security is best considering your core business processes. If the cost of mitigation outweighs the true value to the business, but the form of mitigation is ultimately part of your security posture, it may be very feasible to transfer that risk and mitigation to someone else. The result is that your security posture is satisfied, core business operations are not consumed by ancillary events and decision making, and portions of security that remain your focus can be aligned closer to business objectives — thus enabling business.

Depth and Exposure

It is one thing to determine that you could benefit from outsourcing some or all of your security needs; it is another to associate your specific needs with the concept of third-party involvement. Simply stated, security is layers — similarly, technology is expressed in layers (e.g., OSI model, security architecture) — and the more security is desired and applied, the greater the depth into the layers of technology, architecture, and process a provider must dive into your business. With the integration of managed security, there is an element of exposure and the inherent reliance placed on the shoulders of this, hopefully trusted, entity.

The depth requirement of integrating managed security services truly depends on the type and scope of the services being provided. An example is firewall management. You may only review the logs produced by the firewall to make various determinations, such as penetration attempts, errors, or unscrupulous activities. The depth required is very limited. There is usually no need for MSSP equipment to be installed on the customer's premises, and the logs can be posted regularly or streamed to the MSSP.

In contrast, given the same scenario, the MSSP has the authority to make modifications to the firewall configuration to accommodate changes in the environment, such as making rule additions to thwart an attack. To further the example, there may be proprietary tools or traditional applications to monitor the state of an application. Therefore, the MSSP can make decisions based on several pieces of information collected from many layers and take action at each layer for which it has influence.

To expound on the previous description, envision a router and firewall pair controlling traffic. Behind the firewall is a Web server running on Trusted Solaris (a Trusted Operating System [TOS]) providing application services supported by a DB2 database running on a S/390 deep in the environment segmented by another firewall. Between the information available from the firewall, Web server application, TOS, and the S/390, there are many points to make incisive judgments on the security of the service being provided. A potentially simple application can provide unparalleled access into the heart of a business's network if not properly controlled. An MSSP, if prepared to provide such support, can rationalize the collected information and compare it to external data, such as vulnerability notices, to quickly make determinations on the state of security in the event of an anomaly anywhere between the router and the back-office system.

However, as you can see, if an MSSP were to attempt to provide this scope of service to an organization, the access and control privileges required to bring value to the service (i.e., response time, use of the information collected, decision-making process) would commit the customer to trusting the MSSP implicitly.

If a database object was corrupted by the Web application, who is to blame? Was it a hacker? If it was, should the MSSP not have detected it and taken the appropriate precautions? Or, was it a change the MSSP performed without knowledge of the customer to mitigate the threat of a monitored attack? What happens when developers and administrators make changes to accommodate a new application, and the MSSP perceives the use of the new application as an attack or is simply not notified? It is possible the application will not function, causing some confusion.

As one can see, the requirements and obligations from the MSSP and the customer can become complex. Depending on the service demanded by the customer and the requirements those demands place on the MSSP, the service level agreements (SLA) can become legally intense documents. With this much sharing of responsibility of risk and threat mitigation, it would be easy to stamp the SLA as an insurance policy — this concept is further investigated later in this chapter.

Characteristics of Outsourced Security

There are several options when considering outsourcing security. Fundamentally, an organization must decide what areas it would feel comfortable relinquishing control over. Additionally, it is necessary to investigate the kind of change management that would be employed. The ability to easily address the security someone else is providing and correlate that with business objectives verified against a security policy can be critical in some fast-moving and dynamic companies.

To better understand what can be considered "outsource-able," it is necessary to discuss the types of services that are typically offered. Essentially, these are all very simple and somewhat obvious. However, what is not so obvious are the nuances of the services, their impact on an organization's infrastructure, and the needs placed on that infrastructure.

Managed Services

Managed services are when third-party companies monitor the condition of a device or system and make the appropriate adjustments based on customer, technical, or environmental demands. For example, a managed firewall service might make rule modifications on behalf of its customers to permit, deny, or simply modify the rules to accommodate a specific need. For small organizations, this is not time consuming because there are typically few rules. However, what if the same small company that chose to manage its own firewall with possibly limited resources was not aware of a security hole and the associated available patch? Or did not have sufficient experience to know that the patch might cause unrelated or obscure issues that could cause even greater havoc, if not another security vulnerability?

Managed security services represent the bulk of the concept of the MSSP definition. They manage the technology in varying degrees of complexity. Some use proprietary software to collect logs from systems and post them back to a security operations center (SOC) to perform an analysis.

They typically monitor the system's general functions, look for signs of performance issues, and review new vulnerabilities and patches that may need to be applied. This primarily revolves around the security application being monitored, as in the case with Checkpoint running on Windows NT. More effort is typically spent on the status of the application rather than on the hardware or operating system. Of course, this is a good example of the depth of the service — the MSSP's depth or range of offering and capability, and the associated requirements placed on the customer's infrastructure.

As one can see, there are several options. The following sections explain these options in greater detail.

Appliance or System

As briefly introduced above, there are different concepts of management based on the equipment involved. An appliance is a dedicated system to perform a specific task. Appliances are differentiated by a dedicated operating system uniquely created or modified to accommodate the security service. In contrast, a security service may be provided by an application installed on a general operating system (OS) that was not specifically designed for that application.

Using an appliance, the MSSP has more options available for a greater range of service capability. This is due to the packaged solution providing single access to most, if not all, of the critical layers of the device. This is a substantial point to consider. With a dedicated system, the MSSP can manage several characteristics of the

system without additional and possibly unacceptable access to the customer's network. Granted, this is not for all scenarios. For example, if a Nokia CheckPoint solution were in use, the OS is designed specifically to support CheckPoint and provide other network options. The MSSP can easily manage CheckPoint and take advantage of features in the IP Security Operating System (IPSO) to promote further management capabilities.

If the system is based on a traditional operating system, there may be a greater requirement placed on the customer to get the service. Additionally, this service may be necessary, in the customer's eyes, as a significant portion of the MSSP's services. An example is a customer wants the status and health of the disk drive system to ensure stability in the system. To accomplish this on a traditional operating system running on a server platform would usually require supplementary technology and access rights. On a single platform (e.g., an appliance), the options are usually greater due to the assumed architecture by the vendor. An example would be that a Solaris system running CheckPoint[2] will require more attention to the operating system because it was not specifically designed to only support a firewall application.

CPE Ownership

This may appear to be an oversimplification, but the owner of the customer premise equipment (CPE) can have impacts on the services offered, the scope of capability of the MSSP, the type of service, and the cost. Another aspect of CPE is when an MSSP requires its own systems to reside on the customer's network to perform various services. An example might be a syslog system that collects logs securely from many devices and compiles them to be sent to the SOC for analysis. On some large implementations, the logs are reviewed for anomalies at the collection point and only the items that appear suspicious are forwarded.

There are many situations that must be considered for a company that is investigating outsourcing security services. If an organization owns 20 firewalls and wishes to have them managed by a third party, beyond the obvious vendor platform supported by the MSSP, there is the version of the application or appliance that must be considered. A customer may be required to upgrade systems to meet the minimum requirements set forth by the MSSP.

Adding to the cost, some MSSPs will provide the equipment to manage the solution, but is the cost justifiable compared to purchasing the same equipment? There are many issues in this scenario, including:

- Will the MSSP upgrade and maintain the latest version of the system or software?
- Will the MSSP test patches to ensure the customer is not vulnerable to incompatibilities?
- Are the MSSP's systems properly integrated into the customer's environment to ensure the investment is reaching its potential? (This is especially interesting for intrusion detection systems.)
- In the event of a system failure, what is the repair timeframe and type? For example, does the customer simply receive a new system in the mail, or does someone from the MSSP come on-site to repair or replace the failed system?

Information Services

Information services collect all the information concerning security incidents, vulnerabilities, and threats, and provide a detailed explanation of what is impacted, and plausible remediation tactics. The information may include tools or other configuration options to determine if the customer is vulnerable and to provide links to patches or other updates to rectify vulnerablities. Additionally, information is processed to represent the specific environmental conditions of the infrastructure.

If a new virus is discovered that impacts Lotus Notes and not the more prominent target of hackers (i.e., Microsoft Exchange), the announcement may not get as much airplay but would be very important to a Lotus Notes administrator. The same situation applies in reverse. With all the Microsoft security vulnerabilities, which are seemingly endless, people using Linux, AIX, Solaris, Lotus, Apache, etc. have to review all the announcements to isolate what truly demands attention.

Information services can do many things for an organization. The following sections take a look at some examples.

Vulnerability Alerts

Staying in tune with vulnerability announcements and bugs can be a full-time job. In some cases, simply separating the valuable information from the load of indiscriminant data is very time consuming. There are information services designed specifically to collect information from many resources and compile a compre-

hensive list that pertains to a customer's specific situation. In many cases, one simply provides profile information and the information is sent back based on that profile.

Patches and Upgrades

There are several vendors that continually provide patches for software and systems that have a security component, if they are not dedicated to resolving a security issue. For many information services, communicating the vulnerability with information about an available patch is very helpful for customer organizations.

Heuristics

Collecting information is not all that complicated but reducing that information into a manageable compilation of data that generally applies to your environment can become time consuming. However, comparing dissimilar information that seemingly has little in common can reveal insights, thereby increasing the value of that information. A great deal can be determined from properly applied heuristic methods to gain more information than that collected.

Collecting data from many points to disclose more information is old hat for black hats. Hackers would collect small pieces of information that, on the surface, provided very few facts about the target. Using social engineering, dumpster diving, port scanning, and network sniffing, attackers can make very perceptive observations and determinations about a network. This ultimately gives them the advantage because few others have sought out the same heuristic opportunity.

Many MSSPs provide an excellent opportunity to glean information and filter data on the customer's behalf based on some preliminary rules and profiles established at the beginning of the service. In some cases, the information is presented on the Web and customers modify their profile to engineer the data dynamically, to refine the final presentation to their needs.

Of course, this is the last hurdle for information. As explained, there is update information, threat data, and news that may be applicable to one's industry, each presenting information from that industry domain. With the addition of a heuristic methodology that investigates relationships between the primary forms of information, one can come to decisions quickly and with reduced risk of making errors.

A good example would be a news report of a large ISP going down in a major metropolitan area near your home, causing issues for thousands of users. Later, there is a report of a DoS vulnerability about a widely used driver for a network card on Solaris systems. Solaris immediately provides a patch for the driver. Do you apply the patch? Not without more information, and certainly not without more information on how this patch will affect you. Does the patch take into consideration that you have 15 NICs in seven E10k's in two clusters running a modified kernel and custom application supporting 521 financial firms? That may be a somewhat extreme example. However, there may be other unrelated information about an application, network device, or operating system that may give pause to applying the patch, regardless of the amount of assurance the vendor and peers submit.

Monitoring Services

Monitoring services are an interesting twist on managed security offerings. Monitoring services are very specific in that they typically do not directly impact the network system's configurations. As described above, MSSPs usually perform modifications to critical systems that are responsible for infrastructure security, such as firewalls, routers, VPN systems, etc. In contrast, a monitoring service provider collects information from the network, makes various determinations, and contacts the customer, depending on the established communication protocol.

Monitoring service providers will identify events on the network and assign them to a security classification to ensure that the response and communication protocol are proportional to the severity of the measured event. In essence, this is founded on the heuristics of information management discussed above. By collecting data from many sources, a monitoring service provider can give substantial insight into the activities on one's network.

Communication Protocol

A communication plan is an established process that will be followed by the customer and MSSP to ensure an event is clearly communicated to all parties. This is a critical issue because the monitoring service provider

typically does nothing to thwart or mitigate the attack or event. Therefore, if an event is detected and classified, the customer needs to be made aware.

The protocol is directly related to the classification or severity level of the event. The following is a typical list of classifications:

- *Informational.* This classification refers to information collection activities, such as port scanning. Port scanning is a process many attackers use to seek out services running on systems with known vulnerabilities, identify operating systems with known weaknesses, or attempt to learn about the target architecture.
- *Warning.* An event is identified as a "warning" when suspicious activities are detected at a firewall and on the target system(s), but are not successful. A good example is a modified HTTP GET string sent to a Web server to gain information from the system. Although the firewall may allow a GET command, not aware of the malicious string contents, the Web server survived the attempted access.
- *Critical.* A "critical" classification is an event that is consistent, very specific, and requires immediate action to remedy. Usually, this is the sign of a committed attacker that is clear of the target's defenses and has the process for gaining the necessary access.
- *Emergency.* This classification indicates a security breach has occurred and mitigation and recovery procedures must begin immediately.

To assign an event, it is necessary to have human interaction with several levels of information from firewall and IDS logs to system logs and traces from the network track the event through the infrastructure.

Similarly, each event demands a certain level of communication:

- *Informational.* These are communicated in weekly reports to the customer, listing the events in order of volume, consistency, and which vulnerabilities or services are being searched for.
- *Warning.* An e-mail is usually sent to the administrators and management at the client, detailing the attack signature and recommendations for mitigation.
- *Critical and Emergency.* These demand direct communication to primary contacts at the customer. The major differences are the number of retries, duration between communication attempts, and the list of people to be notified.

Service Characteristics

To effectively monitor a network, it may be necessary to monitor entire network segments and dozens, if not hundreds, of systems. In contrast, if only the perimeter is monitored, the attacks that originate internally or get past the firewall may go undetected. Additionally, if only the IDSs are monitored, it will be difficult to correlate those warnings against other internal systems.

Inevitably, monitoring a network demands interfaces at several levels to collect information used to build a comprehensive image of the information flow. This will permit the MSSP to measure an attack's impact and penetration while learning the process and determining the criticality of the attack.

Some examples of elements that will need to be monitored to realize the service's full potential include:

- Internet-facing router(s)
- Firewall(s)
- VPN devices
- Intrusion detection systems (IDSs)
- Internet mail/relay servers
- Web servers
- Application servers
- Database servers
- Switches

By maintaining awareness of traffic flows not only for systems facing the Internet but also for applications and servers, it is possible to obtain a clear understanding of one's network and the picture of attacks as they flow in and out. This is especially valuable when an event is detected and one has the ability to logically trace the activity through the infrastructure to determine its impact.

Complexities begin to arise when faced with the types of logs or monitoring devices that are required by the MSSP and their relative exposure to proprietary information. For example, many internal e-mail systems

do not encrypt the authentication process — much less mail — as with traditional POP users. To expound upon this example, access to system logs may reveal activities on the system or application that an organization may not wish to share with a third party. Again, this represents the fine line between exposure of delicate information to an outside organization in an attempt to transfer monitoring, or management, responsibilities to another entity.

Outsourcing Conditions

We have discussed the business of security and the role it can play in the world of business. Additionally, the services have been outlined and some types of MSSP services presented. However, when is one supposed to use MSSPs? And once one determines that one should, what is the best way to approach the integration? There are many assumptions that can be made based on the above sections, but let us take the opportunity to scrutinize the decision-making process, the integration of an MSSP, and the impacts of such a decision.

Critical Characteristics

If considering outsourcing security, it is necessary to understand the personality of the organizations with which one seeks to partner. We have discussed the security and business ramifications to some degree, and the decisions will directly correlate with the type of vendor one ultimately will need to investigate.

Managed security is a moderately new concept and because many organizations possess the capability to provide services of this nature, many have risen to the top of the list for their respective type of offerings. Nevertheless, understanding their distinctiveness and how it maps to one's organizational culture is what should be measured — not the popularity or simply the cost.

Following are several examples of specific areas that should be reviewed to gauge the potential effectiveness of a managed services provider for one's organization.

Monitoring and Management

Essentially, understanding the impact of the MSSP's service will directly relate to which systems are affected by the MSSP's involvement within the environment.

For managing services, it is essential to understand the scope of products that the MSSP supports and will manage. Also, the degree to which the MSSP will interact with those systems will be the differentiator of the service and the demands of the customer. Changing firewall rules is dramatically different from managing the operating system or appliance. Clearly aligning customer requirements and demands to the scope of service is important.

Monitoring services are generally more simplistic but have greater involvement, as mentioned above, in the inner workings of the customer's environment. Nevertheless, to fully realize the service's potential for tracking and measuring attacks on the infrastructure, this is a necessary evil.

Adaptation

Probably the most discriminate measurement of the value of an MSSP service offering is its ability to adjust to changes in the security industry, tactics used by attackers, and the demands typically placed on security solutions by the organization. Maintaining awareness of publicized vulnerabilities is only one portion of a very complex formula used to manage security.

In many cases, the history and longevity of an MSSP can directly correlate with its ability to adapt to new attack strategies based on its experience in monitoring and gauging attacks. For example, an MSSP that has a great deal of experience in the industry can identify attack signatures that may not fit the traditional methodology reflected in the majority of documented attacks. The only way is to watch the flow of information to fully appreciate the risk posed by a questionable session. The only way to accomplish this is by pure human interaction. Once someone can visualize the event, it is possible to match whitehat to blackhat — more than any computer could accomplish through statistical analysis based on signatures.

Security is a maze of layers and an attacker can manipulate systems and processes within each layer that may appear to be normal operations within that layer. To provide a valuable service, the MSSP must be able to adapt to new methodologies and tactics in addition to understanding the traditional vulnerabilities.

Track Record

A good historical record and longevity in the marketplace are indications of successful operations. Additionally, the longer a company has successfully provided services of this type, the more likely that an organization's investment in the partnering will last long enough to establish a good relationship and evolve with their offerings. Unfortunately, the history is no promise that one will be able to maintain a close association. Mergers and acquisitions are a common reality in today's market, and a changing of the guard can be very painful.

One component that is typically overlooked is assessing the MSSP's customer base to determine how many, or what percentage of, customers are similar to one's organization and have the demands one places on the type of service being reviewed. Again, it is not simply a size or type comparison, but rather the successful merger of service and security posture maintained in accordance with customer business demands.

Can They Physically Perform

Depending on the scope of the investment, it is recommended that the security operations center (SOC) is visited and inspected for operational purposes. Basically, the ability to serve is directly proportional to the capacity of the systems and availability.

For example, if the SOC has one connection to the Internet, it is at risk of being severed — ultimately stopping the service. If the SOC has more than one service connection, is it in the same conduit? Are they to the same provider? There are endless amounts of redundancy issues that go well beyond the capacity of this chapter, but an MSSP's ability to survive a catastrophe will become your organization's fundamental concern.

By outsourcing security, one essentially trusts the people managing the systems (yours and theirs). Security awareness and involvement in the industry constitute what one is buying. The ability to commit resources to determining what is a concern and what is not, what is an attack, what are the latest vulnerabilities that have an impact, etc. constitute what one is buying. Anyone can set up a system to monitor logs, but valuable human interaction is the final layer of security. Therefore, what is the quality of the people the MSSP employs? It simply comes to a question of the type of people, and the rest is secondary.

Services

Possibly stating the obvious, the comprehensiveness of the service offerings is a dimension that can provide insight into what the MSSP feels is important. The scope and type of offerings can be a positive or a negative, depending on the perspective.

If an MSSP provides every possible service (e.g., managed VPN, managed firewall, content management, managed IDS, high availability, etc.), the impression can be a "Jack of all trades" — a perception of the commitment in filling the gaps of security but not the whole picture. In contrast, a provider may simply have a single service that it simply performs very well. Of course, if this is not the service one finds value in, their selection as the vendor of choice is in certain jeopardy.

Once again, this relates to the internal investigative process to determine what is critical to one's organization, what should be controlled internally, and what is the commodity of security services that are better left to people whose economy is structured to support that demand. It is merely economies of scale and one's core business purpose. Define what you do and are capable of doing within the bounds of your business directives and rely on others to perform the tasks that are their fundamental reason for existence in the industry.

Service Level Agreements and Repercussions

Service level agreements (SLAs) can be complex, and they can be tedious. Nevertheless, SLAs are incredibly important to define the service's expectations, especially with event-related offerings. Many SLAs refer to the time period it will take to respond to an event and the process for managing the event and recovering from it, which may include implementing procedures and processes to reduce the threat of the event from repeating.

However, defining the event clearly can elude most SLAs. This is where "buyer beware" and "Annie get your lawyer" begin to ring clearly in the background of contract negotiations. SLAs are where the service truly meets the expectations of the customer. It is highly recommended that the SLA be one of, if not the first item reviewed to save time in sales meetings. By the time an organization is investigating an MSSP, it should be very confident of its needs and the MSSP's offerings. Therefore, understanding the nuances of the service should be a primary and constant focus of discussions.

The SLA should cover every characteristic, from system deployment, to policy changes, incident response and handling, billing, responsible parties, action plans, upgrades, communication plans, and service acknowl-

edgment. (*Note:* Acknowledgment can be most critical. If one's expectations do not match the services rendered at any point, there is little value in the service.)

Finally, what could be considered the absolute is restitution and fault identification. That is, if an organization submits to a partnership with a managed security provider, it is shifting responsibility to that entity. The organization is paying for a defensive service that could become ingrained into its very business and could have a negative impact if not managed correctly. It can be much like a termite protection service. One invests in a company to visit one's home regularly to inspect for termites and check the bait. This is something very important but one does not have the equipment or expertise to facilitate comprehensive protection — so one delegates and shifts responsibility. If one's home suffers damage, the pest control company may be responsible for damages — depending on the contract or, in this case, the SLA.

In short, an organization should clearly understand its needs and the services that are offered by the MSSP, and then ensure that the SLA provides the necessary catalyst to successfully bind business objectives with service expectations.

The Future of Managed Security and Monitoring Services

What happens when the MSSP does not stop a virus from bringing an organization to a halt? Who is to blame? Today, in some cases, one may receive an official letter that basically states the public relations version of, "Oops! …Sorry about that, we'll do our best to stop that in the future. But there are no guarantees."

The subject for this section was alluded to in earlier discussions and represents a new direction in information security — insurance. It seems that few organizations are geared for addressing the real complexity of security. If a company is not in the business of providing security solutions, why should it invest in maintaining security? There are two answers to this: Based on business demands, the ability to provide and commit to secure operations is acceptable given the risks to revenue generating processes and assets. Of course, the other answer is that one simply cannot make that commitment. Nevertheless, in both cases, one must consider the risks associated with supporting security and outsourcing, and how much of each one pursues.

No matter what the degree or type of support for the security posture is chosen, risk is the common denominator and is inherently associated with money. How much of this money one is responsible for can be associated with who assumes the risk and to what level.

The natural conclusion is for MSSPs to become insurance providers or underwriters supported by larger firms willing to invest in the MSSP as risk-mitigation services. This will allow the insurance provider to pass on savings or incentives for using an MSSP.

This represents an interesting point. With the involvement of insurance companies in the support of services, they will undoubtedly become more efficient in not only measuring architectures for security but in being able to produce or certify standards currently available. This should come as no surprise. Insurance companies were the founders of fire regulations controlled and managed by the government today. By defining or sanctioning standards, insurance companies will enable MSSPs to address the market from another cost-saving avenue — once again leveraging their position as the economically engineered security trade to provide the necessary protection that eludes some companies.

Managed monitoring services are beginning to obtain market differentiation from their managed-security cousins and and also an identity of their own. Given the value-to-impact ratio, managed monitoring, when properly integrated and controlled, provides a strong argument for services that fill the gap in most environments.

As the breadth of monitoring capabilities increases and the correlation between disparate network elements becomes more refined, it seems obvious that monitoring services will continue to experience growth. In the future, one could imagine a firewall-like system that controls the information that the MSSP was providing. For example, e-mail content could be removed, thereby only allowing the MSSP to obtain the critical information in the header. Application monitoring could be limited to certain types of logs — not level or severity — but rather log content could be filtered for known log exposures.

Nevertheless, it is clear that monitoring services provide insight into network activities that can quickly relate to reducing risk, maintaining a measurable security posture, and reducing the exposure to threats.

Conclusion

Information security is challenging to manage in any environment, essentially due to the fundamental characteristics that information security represents. By virtue of its definition, security management is an intricate process comprised of technical issues, human interfaces, legal requirements, vigilance, and tenacity, all in balance with a constantly changing environment.

Any organization considering managed security, as an option for enhancing security through transferring risk or augmenting the existing security program, must be introspective and clearly realize its position on security operations versus exposure of the business. Once the core business objectives are weighed against performing the necessary duties required to maintain the desired security posture, an organization can begin to determine the type, scope, and depth of managed services that best fit its business.

Notes

1. The Dendrocnide is also known as the Australian stinging tree. Speaking from personal experience, this vicious plant will sting you with a crystal-like poison that is not only painful and lasts for days, but reactivates when water is introduced. A few stinging trees planted on the perimeter are a good deterrent.
2. The use of CheckPoint as an example is to support continuity between the examples given. The statements are not meant to insinuate that these options or challenges are only associated with CheckPoint. Additionally, there are many different firewalls available on the market and using CheckPoint's as an example seemed to be the most obvious to convey the necessary subject.

Domain 4
Applications and Systems Development Security

Applications and systems development security refers to the controls that are included within system and application software, and the steps used in their development. Applications are agents, applets, software, databases, data warehouses, and knowledge-based systems. These applications may be used in distributed or centralized environments.

The professional should fully understand the security and controls of the systems development life-cycle process. Included in this domain are application controls, change controls, data warehousing, data mining, knowledge-based systems, program interfaces, and concepts used to ensure data and application integrity, confidentiality, and availability. The security and controls that should be included within system and application software are discussed. The steps and security controls in the software life cycle and change control process and the concepts used to ensure data and software integrity, confidentiality, and availability are also discussed.

Contents

Contents

Section 4.2 Databases and Data Warehousing

Section 4.3 Systems Development Controls

Section 4.4 Malicious Code

Section 4.5 Methods of Attack

90

Security Models for Object-Oriented Databases

James Cannady

Object-oriented (OO) methods are a significant development in the management of distributed data. Database design is influenced to an ever-greater degree by OO principles. As more DBMS products incorporate aspects of the object-oriented paradigm, database administrators must tackle the unique security considerations of these systems and understand the emerging security model.

Introduction

Object-oriented (OO) programming languages and OO analysis and design techniques influence database system design and development. The inevitable result is the object-oriented database management system (OODBMS).

Many of the established database vendors are incorporating OO concepts into their products in an effort to facilitate database design and development in the increasingly OO world of distributed processing. In addition to improving the process of database design and administration, the incorporation of OO principles offers new tools for securing the information stored in the database. This chapter explains the basics of database security, the differences between securing relational and object-oriented systems, and some specific issues related to the security of next-generation OODBMSs.

Basics of Database Security

Database security is primarily concerned with the secrecy of data. Secrecy means protecting a database from unauthorized access by users and software applications.

Secrecy, in the context of database security, includes a variety of threats incurred through unauthorized access. These threats range from the intentional theft or destruction of data to the acquisition of information through more subtle measures, such as inference. There are three generally accepted categories of secrecy-related problems in database systems:

1. *The improper release of information from reading data that was intentionally or accidentally accessed by unauthorized users.* Securing databases from unauthorized access is more difficult than controlling access to files managed by operating systems. This problem arises from the finer granularity that is used by databases when handling files, attributes, and values. This type of problem also includes the violations to secrecy that result from the problem of inference, which is the deduction of unauthorized information from the observation of authorized information. Inference is one of the most difficult factors to control in any attempt to secure data. Because the information in a database is semantically related, it is possible to determine the value of an attribute without accessing it directly. Inference problems are most serious

in statistical databases where users can trace back information on individual entities from the statistical aggregated data.

2. *The improper modification of data.* This threat includes violations of the security of data through mishandling and modifications by unauthorized users. These violations can result from errors, viruses, sabotage, or failures in the data that arise from access by unauthorized users.
3. *Denial-of-service threats.* Actions that could prevent users from using system resources or accessing data are among the most serious. This threat has been demonstrated to a significant degree recently with the SYN flooding attacks against network service providers.

Discretionary versus Mandatory Access Control Policies

Both traditional relational database management system (RDBMS) security models and OO database models make use of two general types of access control policies to protect the information in multilevel systems. The first of these policies is the discretionary policy. In the discretionary access control (DAC) policy, access is restricted based on the authorizations granted to the user.

The mandatory access control (MAC) policy secures information by assigning sensitivity levels, or labels to data entities. MAC policies are generally more secure than DAC policies, and they are used in systems in which security is critical, such as military applications. However, the price that is usually paid for this tightened security is reduced performance of the database management system. Most MAC policies incorporate DAC measures as well.

Securing an RDBMS versus an OODBMS: Know the Differences

The development of secure models for OODBMSs has obviously followed on the heels of the development of the databases themselves. The theories that are currently being researched and implemented in the security of OO databases are also influenced heavily by the work that has been conducted on secure relational database management systems.

Relational DBMS Security

In traditional RDBMSs, security is achieved principally through the appropriate use and manipulation of views and and the SQL GRANT and REVOKE statements. These measures are reasonably effective because of their mathematical foundation in relational algebra and relational calculus.

View-Based Access Control

Views allow the database to be conceptually divided into pieces in ways that allow sensitive data to be hidden from unauthorized users. In the relational model, views provide a powerful mechanism for specifying data-dependent authorizations for data retrieval.

Although the individual user who creates a view is the owner and is entitled to drop the view, he or she may not be authorized to execute all privileges on it. The authorizations that the owner may exercise depend on the view semantics and on the authorizations that the owner is allowed to implement on the tables directly accessed by the view. To exercise a specific authorization on a view, the owner must possess the same authorization on all tables that the view uses. The privileges the owner possesses on the view are determined at the time of view definition. Each privilege the owner possesses on the tables is defined for the view. If, later on, the owner receives additional privileges on the tables used by the view, these additional privileges will not be passed on to the view. In order to use the new privileges within a view, the owner will need to create a new view.

The biggest problem with view-based mandatory access control is that it is impractical to verify that the software performs the view interpretation and processing. If the correct authorizations are to be assured, the system must contain some type of mechanism to verify the classification of the sensitivity of the information in the database. The classification must be done automatically, and the software that handles the classification must be trusted. However, any trusted software for the automatic classification process would be extremely complex. Furthermore, attempting to use a query language such as SQL to specify classifications quickly becomes convoluted and complex. Even when the complexity of the classification scheme is overcome, the view can do nothing more than limit what the user sees — it cannot restrict the operations that may be performed on the views.

GRANT and REVOKE Privileges

Although view mechanisms are often regarded as security "freebies" because they are included within SQL and most other traditional relational database managers, views are not the sole mechanism for relational database security. GRANT and REVOKE statements allow users to selectively and dynamically grant privileges to other users and subsequently revoke them if necessary. These two statements are considered to be the principal user interfaces in the authorization subsystem.

There is, however, a security-related problem inherent in the use of the GRANT statement. If a user is granted rights without the GRANT option, he should not be able to pass GRANT authority on to other users. However, the system can be subverted by a user by simply making a complete copy of the relation. Because the user creating the copy is now the owner, he can provide GRANT authority to other users. As a result, unauthorized users are able to access the same information that had been contained in the original relation. Although this copy is not updated with the original relation, the user making the copy could continue making similar copies of the relation, and continue to provide the same data to other users.

The REVOKE statement functions similarly to the GRANT statement, with the opposite result. One of the characteristics of the use of the REVOKE statement is that it has a cascading effect. When the rights previously granted to a user are subsequently revoked, all similar rights are revoked for all users who may have been provided access by the originator.

Other Relational Security Mechanisms

Although views and GRANT/REVOKE statements are the most frequently used security measures in traditional RDBMSs, they are not the only mechanisms included in most security systems using the relational model. Another security method used with traditional relational database managers, which is similar to GRANT/REVOKE statements, is the use of query modification.

This method involves modifying a user's query before the information is retrieved, based on the authorities granted to the user. Although query modification is not incorporated within SQL, the concept is supported by the Codd–Date relational database model.

Most relational database management systems also rely on the security measures present in the operating system of the host computer. Traditional RDBMSs such as DB2 work closely with the operating system to ensure that the database security system is not circumvented by permitting access to data through the operating system. However, many operating systems provide insufficient security. In addition, because of the portability of many newer database packages, the security of the operating system should not be assumed to be adequate for the protection of the wealth of information in a database.

Object-Oriented DBMS Characteristics

Unlike traditional RDBMSs, secure OODBMSs have certain characteristics that make them unique. Furthermore, only a limited number of security models have been designed specifically for OO databases. The proposed security models make use of the concepts of encapsulation, inheritance, information-hiding, methods, and the ability to model real-world entities that are present in OO environments.

The object-oriented database model also permits the classification of an object's sensitivity through the use of class (or entities) and instance. When an instance of a class is created, the object can automatically inherit the level of sensitivity of the superclass. Although the ability to pass classifications through inheritance is possible in object-oriented databases, class instances are usually classified at a higher level within the object's class hierarchy. This prevents a flow control problem, where information passes from higher to lower classification levels.

OODBMSs also use unique characteristics that allow these models to control the access to the data in the database. They incorporate features such as flexible data structure, inheritance, and late binding. Access control models for OODBMSs must be consistent with such features. Users can define methods, some of which are open for other users as public methods. Moreover, the OODBMS may encapsulate a series of basic access commands into a method and make it public for users, while keeping basic commands themselves away from users.

Proposed OODBMS Security Models

Currently, only a few models use discretionary access control measures in secure object-oriented database management systems.

Explicit Authorizations

The ORION authorization model permits access to data on the basis of explicit authorizations provided to each group of users. These authorizations are classified as positive authorizations because they specifically allow a user access to an object. Similarly, a negative authorization is used to specifically deny a user access to an object.

The placement of an individual into one or more groups is based on the role that the individual plays in the organization. In addition to the positive authorizations that are provided to users within each group, there are a variety of implicit authorizations that may be granted based on the relationships between subjects and access modes.

Data-Hiding Model

A similar discretionary access control secure model is the data-hiding model proposed by Dr. Elisa Bertino of the Universitá di Genova. This model distinguishes between public methods and private methods.

The data-hiding model is based on authorizations for users to execute methods on objects. The authorizations specify which methods the user is authorized to invoke. Authorizations can only be granted to users on public methods. However, the fact that a user can access a method does not automatically mean that the user can execute all actions associated with the method. As a result, several access controls may need to be performed during the execution, and all of the authorizations for the different accesses must exist if the user is to complete the processing.

Similar to the use of GRANT statements in traditional relational database management systems, the creator of an object is able to grant authorizations to the object to different users. The "creator" is also able to revoke the authorizations from users in a manner similar to REVOKE statements. However, unlike traditional RDBMS GRANT statements, the data-hiding model includes the notion of protection mode. When authorizations are provided to users in the protection mode, the authorizations actually checked by the system are those of the creator and not the individual executing the method. As a result, the creator is able to grant a user access to a method without granting the user the authorizations for the methods called by the original method. In other words, the creator can provide a user access to specific data without being forced to give the user complete access to all related information in the object.

Other DAC Models for OODBMS Security

Rafiul Ahad has proposed a similar model that is based on the control of function evaluations. Authorizations are provided to groups or individual users to execute specific methods. The focus in Ahad's model is to protect the system by restricting access to the methods in the database, not the objects. The model uses proxy functions, specific functions, and guard functions to restrict the execution of certain methods by users and enforce content-dependent authorizations.

Another secure model that uses authorizations to execute methods has been presented by Joel Richardson. This model has some similarity to the data-hiding model's use of GRANT/REVOKE-type statements. The creator of an object can specify which users may execute the methods within the object.

A final authorization-dependent model emerging from OODBMS security research has been proposed by Dr. Eduardo B. Fernandez of Florida Atlantic University. In this model the authorizations are divided into positive and negative authorizations. The Fernandez model also permits the creation of new authorizations from those originally specified by the user through the use of the semantic relationships in the data.

Dr. Naftaly H. Minsky of Rutgers University has developed a model that limits unrestricted access to objects through the use of a view mechanism similar to that used in traditional relational database management systems. Minsky's concept is to provide multiple interfaces to the objects within the database. The model includes a list of laws, or rules, that govern the access constraints to the objects. The laws within the database specify which actions must be taken by the system when a message is sent from one object to another. The system may allow the message to continue unaltered, block the sending of the message, send the message to another object, or send a different message to the intended object.

Although the discretionary access control models do provide varying levels of security for the information within the database, none of the DAC models effectively addresses the problem of the authorizations provided to users. A higher level of protection within a secure OO database model is provided through the use of mandatory access control.

MAC Methods for OODBMS Security

Dr. Bhavani Thuraisingham of MITRE Corp. proposed in 1989 a mandatory security policy called SORION. This model extends the ORION model to encompass mandatory access control. The model specifies subjects, objects, and access modes within the system, and it assigns security/sensitivity levels to each entity. Certain properties regulate the assignment of the sensitivity levels to each of the subjects, objects, and access modes. In order to gain access to the instance variables and methods in the objects, certain properties that are based on the various sensitivity levels must be satisfied.

A similar approach has been proposed in the Millen–Lunt model. This model, developed by Jonathan K. Millen of MITRE Corp. and Teresa Lunt of SRI/DARPA (Defense Advanced Research Projects Agency), also uses the assignment of sensitivity levels to the objects, subjects, and access modes within the database. In the Millen–Lunt model, the properties that regulate the access to the information are specified as axioms within the model. This model further attempts to classify information according to three different cases:

1. The data itself is classified.
2. The existence of the data is classified.
3. The reason for classifying the information is also classified.

These three classifications broadly cover the specifics of the items to be secured within the database; however, the classification method also greatly increases the complexity of the system.

The SODA Model

Dr. Thomas F. Keefe of Pennsylvania State University proposes a model called Secure Object-Oriented Database (SODA). The SODA model was one of the first models to address the specific concepts in the OO paradigm. It is often used as a standard example of secure object-oriented models to which other models are compared.

The SODA model complies with MAC properties and is executed in a multilevel security system. SODA assigns classification levels to the data through the use of inheritance. However, multiple inheritance is not supported in the SODA model.

Similar to other secure models, SODA assigns security levels to subjects in the system and sensitivity levels to objects. The security classifications of subjects are checked against the sensitivity level of the information before access is allowed.

Polyinstantiation

Unlike many current secure object-oriented models, SODA allows the use of polyinstantiation as a solution to the multiparty update conflict. This problem arises when users with different security levels attempt to use the same information. The variety of clearances and sensitivities in a secure database system result in conflicts between the objects that can be accessed and modified by the users.

Through the use of polyinstantiation, information is located in more than one location, usually with different security levels. Obviously, the more sensitive information is omitted from the instances with lower security levels.

Although polyinstantiation solves the multiparty update conflict problem, it raises a potentially greater problem in the form of ensuring the integrity of the data within the database. Without some method of simultaneously updating all occurrences of the data in the database, the integrity of the information quickly disappears. In essence, the system becomes a collection of several distinct database systems, each with its own data.

Conclusion

The move to object-oriented DBMSs is likely to continue for the foreseeable future. Because of the increasing need for security in the distributed processing environments, the expanded selection of tools available for securing information in this environment should be used fully to ensure that the data is as secure as possible. In addition, with the continuing dependence on distributed data the security of these systems must be fully integrated into existing and future network security policies and procedures.

The techniques that are ultimately used to secure commercial OODBMS implementations will depend in large part on the approaches promoted by the leading database vendors. However, the applied research that has been conducted to date is also laying the groundwork for the security components that will in turn be incorporated in the commercial OODBMSs.

91

Web Application Security

Mandy Andress, CISSP, SSCP, CPA, CISA

It is possible to do almost everything on the Web these days: checking stock quotes, requesting a new service, and buying just about anything. Everyone, it seems, has a Web application. But what exactly does that mean?

Web applications are not distinguishable, finite programs. They include many different components and servers. An average Web application includes a Web server, application server, and database server. The Web server provides the graphical user interface for the end user; the application server provides the business logic; and the database server houses the data critical to the application's functionality.

The Web server provides several different ways to forward a request to an application server and send back a modified or new Web page to the end user. These approaches include the Common Gateway Interface (CGI), Microsoft's Active Server Page (ASP), and Java Server Page (JSP). In some cases, the application servers also support request brokering interfaces such as Common Object Request Broker Architecture (CORBA) and Internet Inter-ORB Protocol.

Web Application Security

Not all applications are created, or implemented, equal, however. The lack of Web application security is quickly becoming a fast and easy way into a company's network. Why? All Web applications are different, yet they are all the same. They all run on the same few Web servers, use the same shopping cart software, and use the same application and database servers, yet they are different because at least part of the application includes home-grown code. Companies often do not have the time or resources to properly harden their servers and perform a thorough review of the application code before going live on the Internet.

Additionally, many programmers do not know how to develop secure applications. Maybe they have always developed stand-alone applications or intranet Web applications that did not create catastrophic results when a security flaw was discovered. In most cases, however, the desire to get a product out the door quickly precludes taking the time to properly secure an application.

Subsequently, many Web applications are vulnerable through the servers, applications, and in-house developed code. These attacks pass right through a perimeter firewall security because port 80 (or 443 for SSL) must be open for the application to function properly. Web application attacks include denial-of-service attacks on the Web application, changing Web page content, and stealing sensitive corporate or user information such as credit card numbers.

Just how prolific are these issues? Well, in the last few months of 2000, the following stories made headlines (and these are just the reported stories). A hacker broke into Egghead.com, potentially exposing its 3.7 million customer accounts. It was not until several weeks later that the company said the hacker did not gain access to customer credit card numbers. By this point, many of the credit cards had been canceled and the damage to Egghead's reputation had already been done. Creditcards.com was the victim of an extortion attempt by a hacker who broke into its site and stole more than 55,000 credit card numbers. The hacker posted the card

numbers on a Web site and demanded money from the company to take them offline. A bug in Eve.com's Web application allowed customers to view other people's orders by simply changing a number in the URL. The bug exposed customer names and addresses, products, and the dates on which they were ordered, the types of credit cards customers used, and the last five digits of the card numbers. Another bug in IKEA's Web application for its catalog order site exposed customer order information. Finally, a bug in Amazon.com's Web application exposed the e-mail addresses of many of its affiliate members. Web application attacks are such a threat that CERT issued an advisory on the subject in February 2000 (see Exhibit 91.1 or go to www.cert.org/advisories/CA-2000-02.html).

Web application attacks differ from typical attacks because they are difficult to detect and can come from any online user — even authenticated ones. To date, this area has been largely neglected because companies are still grappling with securing their networks using firewalls and intrusion detection solutions, which do not detect Web attacks.

How exactly are Web applications vulnerable to attack? The major exploits include:

- Known vulnerabilities and misconfigurations
- Hidden fields
- Backdoor and debug options
- Cross-site scripting
- Parameter tampering
- Cookie poisoning
- Input manipulation
- Buffer overflow
- Direct access browsing

Known Vulnerabilities and Misconfigurations

Known vulnerabilities include all the bugs and exploits in both operating systems and third-party applications used in a Web application. Microsoft's Internet Information Server (IIS), one of the most widely used Web servers, is notorious for security flaws. A vulnerability released in October 2000, the Extended Unicode Directory Traversal vulnerability (Security Bulletin MS00-078), takes advantage of improper Unicode handling by IIS and allows an attacker to enter a specially formed URL and access any file on the same logical drive as the Web server. An attacker can easily execute files under the IUSR_machinename account. IUSR_machinename is the anonymous user account for IIS and is a member of the Everyone and Users groups by default. Microsoft has released a patch for this issue, available for download at www.microsoft.com/technet/security/bulletin/MS00-078.asp.

This topic also covers misconfigurations, or applications that still contain insecure default settings or are configured insecurely by administrators. A good example is leaving one's Web server configured to allow any user to traverse directory paths on the system. This could potentially lead to the disclosure of sensitive information such as passwords, source code, or customer information if it is stored on the Web server (which itself is a big security risk). Another situation is leaving the user with execute permissions on the Web server. Combined with directory traversal rights, this could easily lead to a compromise of the Web server.

Hidden Fields

Hidden fields refers to hidden HTML form fields. For many applications, these fields are used to hold system passwords or merchandise prices. Despite their name, these fields are not very hidden; they can be seen by performing a View Source on the Web page. Many Web applications allow malicious users to modify these fields in the HTML source, giving them the opportunity to purchase items at little or no cost. These attacks are successful because most applications do not validate the returning Web page. They assume the incoming data is the same as the outgoing data.

Backdoor and Debug Options

Developers often create backdoors and turn on debugging to facilitate troubleshooting in applications. This works fine in the development process, but these items are often left in the final application that is placed on

EXHIBIT 91.1 CERT Advisory CA-2000-02 Malicious HTML Tags Embedded in Client Web Requests

This advisory is being published jointly by the CERT Coordination Center, DoD-CERT, the DoD Joint Task Force for Computer Network Defense (JTF-CND), the Federal Computer Incident Response Capability (FedCIRC), and the National Infrastructure Protection Center (NIPC).

Original release date: February 2, 2000
Last revised: February 3, 2000

Systems Affected

- Web browsers
- Web servers that dynamically generate pages based on unvalidated input

Overview

A Web site may inadvertently include malicious HTML tags or script in a dynamically generated page based on unvalidated input from untrustworthy sources. This can be a problem when a Web server does not adequately ensure that generated pages are properly encoded to prevent unintended execution of scripts, and when input is not validated to prevent malicious HTML from being presented to the user.

I. Description

Background

Most Web browsers have the capability to interpret scripts embedded in Web pages downloaded from a Web server. Such scripts may be written in a variety of scripting languages and are run by the client's browser. Most browsers are installed with the capability to run scripts enabled by default.

Malicious Code Provided by One Client for Another Client

Sites that host discussion groups with Web interfaces have long guarded against a vulnerability where one client embeds malicious HTML tags in a message intended for another client. For example, an attacker might post a message like

```
Hello message board. This is a message.
<SCRIPT>malicious code</SCRIPT>
This is the end of my message.
```

When a victim with scripts enabled in their browser reads this message, the malicious code may be executed unexpectedly. Scripting tags that can be embedded in this way include <SCRIPT>, <OBJECT>, <APPLET>, and <EMBED>.

When client-to-client communications are mediated by a server, site developers explicitly recognize that data input is untrustworthy when it is presented to other users. Most discussion group servers either will not accept such input or will encode/filter it before sending anything to other readers.

Malicious Code Sent Inadvertently by a Client for Itself

Many Internet Web sites overlook the possibility that a client may send malicious data intended to be used only by itself. This is an easy mistake to make. After all, why would a user enter malicious code that only the user will see?

However, this situation may occur when the client relies on an untrustworthy source of information when submitting a request. For example, an attacker may construct a malicious link such as

```
<A HREF="http://example.com/comment.cgi? mycomment=<SCRIPT>malicious code</
SCRIPT>"> Click here</A>
```

When an unsuspecting user clicks on this link, the URL sent to example.com includes the malicious code. If the Web server sends a page back to the user including the value of mycomment, the malicious code may be executed unexpectedly on the client. This example also applies to untrusted links followed in e-mail or newsgroup messages.

Abuse of Other Tags

In addition to scripting tags, other HTML tags such as the <FORM> tag have the potential to be abused by an attacker. For example, by embedding malicious <FORM> tags at the right place, an intruder can trick users into revealing sensitive information by modifying the behavior of an existing form. Other HTML tags can also be abused to alter the appearance of the page, insert unwanted or offensive images or sounds, or otherwise interfere with the intended appearance and behavior of the page.

Abuse of Trust

At the heart of this vulnerability is the violation of trust that results from the "injected" script or HTML running within the security context established for the example.com site. It is, presumably, a site the browser victim is interested in enough to visit and interact with in a trusted fashion. In addition, the security policy of the legitimate server site example.com may also be compromised.

EXHIBIT 91.1 CERT Advisory CA-2000-02 Malicious HTML Tags Embedded in Client Web Requests (continued)

This example explicitly shows the involvement of two sites:

```
<A HREF="http://example.com/comment.cgi? mycomment=<SCRIPT SRC='http://bad-
site/badfile'></SCRIPT>"> Click here</A>
```

Note the SRC attribute in the <SCRIPT> tag is explicitly incorporating code from a presumably unauthorized source (bad-site). Both of the previous examples show violations of the same-source origination policy fundamental to most scripting security models:

- Netscape Communicator Same Origin Policy
- Microsoft Scriptlet Security

Because one source is injecting code into pages sent by another source, this vulnerability has also been described as "cross-site" scripting.

At the time of publication, malicious exploitation of this vulnerability has not been reported to the CERT/CC. However, because of the potential for such exploitation, we recommend that organization CIOs, managers, and system administrators aggressively implement the steps listed in the solution section of this document. Technical feedback to appropriate technical, operational, and law enforcement authorities is encouraged.

II. IMPACT

Users may unintentionally execute scripts written by an attacker when they follow untrusted links in Web pages, mail messages, or newsgroup postings. Users may also unknowingly execute malicious scripts when viewing dynamically generated pages based on content provided by other users.

Because the malicious scripts are executed in a context that appears to have originated from the targeted site, the attacker has full access to the document retrieved (depending on the technology chosen by the attacker), and may send data contained in the page back to the site. For example, a malicious script can read fields in a form provided by the real server, then send this data to the attacker.

Note that the access that an intruder has to the Document Object Model (DOM) is dependent on the security architecture of the language chosen by the attacker. Specifically, Java applets do not provide the attacker with any access to the DOM.

Alternatively, the attacker may be able to embed script code that has additional interactions with the legitimate Web server without alerting the victim. For example, the attacker could develop an exploit that posted data to a different page on the legitimate Web server.

Also, even if the victim's Web browser does not support scripting, an attacker can alter the appearance of a page, modify its behavior, or otherwise interfere with normal operation.

The specific impact can vary greatly, depending on the language selected by the attacker and the configuration of any authentic pages involved in the attack. Some examples that may not be immediately obvious are included here.

SSL-Encrypted Connections May Be Exposed
The malicious script tags are introduced before the Secure Socket Layer (SSL) encrypted connection is established between the client and the legitimate server. SSL encrypts data sent over this connection, including the malicious code, which is passed in both directions. While ensuring that the client and server are communicating without snooping, SSL makes no attempt to validate the legitimacy of data transmitted.

Because there really is a legitimate dialog between the client and the server, SSL reports no problems. Malicious code that attempts to connect to a non-SSL URL may generate warning messages about the insecure connection, but the attacker can circumvent this warning simply by running an SSL-capable Web server.

Attacks May Be Persistent through Poisoned Cookies
Once malicious code that appears to have come from the authentic Web site is executing, cookies may be modified to make the attack persistent. Specifically, if the vulnerable Web site uses a field from the cookie in the dynamic generation of pages, the cookie may be modified by the attacker to include malicious code. Future visits to the affected Web site (even from trusted links) will be compromised when the site requests the cookie and displays a page based on the field containing the code.

Attacker May Access Restricted Web Sites from the Client
By constructing a malicious URL, an attacker may be able to execute script code on the client machine that exposes data from a vulnerable server inside the client's intranet.

EXHIBIT 91.1 CERT Advisory CA-2000-02 Malicious HTML Tags Embedded in Client Web Requests (continued)

The attacker may gain unauthorized Web access to an intranet Web server if the compromised client has cached authentication for the targeted server. There is no requirement for the attacker to masquerade as any particular system. An attacker only needs to identify a vulnerable intranet server and convince the user to visit an innocent-looking page to expose potentially sensitive data on the intranet server.

Domain-Based Security Policies May Be Violated
If your browser is configured to allow execution of scripting languages from some hosts or domains while preventing this access from others, attackers may be able to violate this policy.

By embedding malicious script tags in a request sent to a server that is allowed to execute scripts, an attacker may gain this privilege as well. For example, Internet Explorer security "zones" can be subverted by this technique.

Use of Less-Common Character Sets May Present Additional Risk
Browsers interpret the information they receive according to the character set chosen by the user if no character set is specified in the page returned by the Web server. However, many Web sites fail to explicitly specify the character set (even if they encode or filter characters with special meaning in the ISO-8859-1), leaving users of alternate character sets at risk.

Attacker May Alter the Behavior of Forms
Under some conditions, an attacker may be able to modify the behavior of forms, including how results are submitted.

III. Solution

Solutions for Users

None of the solutions that Web users can take are complete solutions. In the end, it is up to Web page developers to modify their pages to eliminate these types of problems.

However, Web users have two basic options to reduce their risk of being attacked through this vulnerability. The first, disabling scripting languages in their browser, provides the most protection but has the side effect for many users of disabling functionality that is important to them. Users should select this option when they require the lowest possible level of risk.

The second solution, being selective about how they initially visit a Web site, will significantly reduce a user's exposure while still maintaining functionality. Users should understand that they are accepting more risk when they select this option, but are doing so in order to preserve the functionality that is important to them.

Unfortunately, it is not possible to quantify the risk difference between these two options. Users who decide to continue operating their browsers with scripting languages enabled should periodically revisit the CERT/CC Web site for updates, as well as review other sources of security information to learn of any increases in threat or risk related to this vulnerability.

Web Users Should Disable Scripting Languages in Their Browsers
Exploiting this vulnerability to execute code requires that some form of embedded scripting language be enabled in the victim's browser. The most significant impact of this vulnerability can be avoided by disabling all scripting languages.

Note that attackers may still be able to influence the appearance of content provided by the legitimate site by embedding other HTML tags in the URL. Malicious use of the <FORM> tag in particular is not prevented by disabling scripting languages.

Detailed instructions to disable scripting languages in your browser are available from our Malicious Code FAQ:

http://www.cert.org/tech_tips/malicious_code_FAQ.html

Web Users Should Not Engage in Promiscuous Browsing
Some users are unable or unwilling to disable scripting languages completely. While disabling these scripting capabilities is the most effective solution, there are some techniques that can be used to reduce a user's exposure to this vulnerability.

Since the most significant variations of this vulnerability involve cross-site scripting (the insertion of tags into another site's Web page), users can gain some protection by being selective about how they initially visit a Web site. Typing addresses directly into the browser (or using securely stored local bookmarks) is likely to be the safest way of connecting to a site.

Users should be aware that even links to unimportant sites may expose other local systems on the network if the client's system resides behind a firewall, or if the client has cached credentials to access other Web servers (e.g., for an intranet). For this reason, cautious Web browsing is not a comparable substitute for disabling scripting.

With scripting enabled, visual inspection of links does not protect users from following malicious links, since the attacker's Web site may use a script to misrepresent the links in the user's window. For example, the contents of the Goto and Status bars in Netscape are controllable by JavaScript.

EXHIBIT 91.1 CERT Advisory CA-2000-02 Malicious HTML Tags Embedded in Client Web Requests (continued)

Solutions for Web Page Developers and Web Site Administrators

Web Page Developers Should Recode Dynamically Generated Pages to Validate Output

Web site administrators and developers can prevent their sites from being abused in conjunction with this vulnerability by ensuring that dynamically generated pages do not contain undesired tags.

Attempting to remove dangerous metacharacters from the input stream leaves a number of risks unaddressed. We encourage developers to restrict variables used in the construction of pages to those characters that are explicitly allowed and to check those variables during the generation of the output page.

In addition, Web pages should explicitly set a character set to an appropriate value in all dynamically generated pages.

Because encoding and filtering data is such an important step in responding to this vulnerability, and because it is a complicated issue, the CERT/CC has written a document that explores this issue in more detail:

http://www.cert.org/tech_tips/malicious_code_mitigation.html

Web Server Administrators Should Apply a Patch from Their Vendor

Some Web server products include dynamically generated pages in the default installation. Even if your site does not include dynamic pages developed locally, your Web server may still be vulnerable. For example, your server may include malicious tags in the "404 Not Found" page generated by your Web server.

Web server administrators are encouraged to apply patches as suggested by your vendor to address this problem. Appendix A contains information provided by vendors for this advisory. We will update the appendix as we receive more information. If you do not see your vendor's name, the CERT/CC did not hear from that vendor. Please contact your vendor directly.

the Internet. Backdoors that allow a user to log in with no password, or a special URL that allows direct access to application configuration, are quite popular.

The existence of this type of Web application vulnerability is caused by a lack of formal policies and procedures that should be followed when taking a system live. A key step in that process should be removing backdoors and disabling debugging options. This simple step will greatly reduce the number of vulnerabilities in any application. This step is often skipped, however, because time constraints on getting the application up and running prevent a formalized approach from being followed.

Cross-Site Scripting

Cross-site scripting is difficult to define because it has many meanings. In general, it is the process of inserting code into pages sent by another source. One way to exploit cross-site scripting is through HTML forms. Forms allow a user to type any information and have it sent to the server. Often, servers take the data input in the form and display it back to the user in an HTML page to confirm the input. If the user types code, such as a JavaScript program, into a form field, the code will be processed by the client's browser when the page is displayed.

Cross-site scripting breaches trust. A user trusts the information sent by the Web server and does not expect malicious actions. With cross-site scripting, a user can place malicious code on the server that will be executed on a different user's machine. Posting messages on a bulletin board is a good example of cross-site scripting. A malicious user completes a form to post a message on a bulletin board. The posting includes some malicious JavaScript code. When an innocent user looks at the bulletin board, the server will send the HTML to be displayed along with the malicious user's code. The code will be executed by the client's browser because it thinks it is valid code from the Web server.

Parameter Tampering

Parameter tampering involves manipulating URL strings to retrieve information the user should not see. Access to the back-end database of the Web application is made through SQL calls that are often included in the URL. Malicious users can manipulate the SQL code to potentially retrieve a listing of all users, passwords, credit card numbers, or any other data stored in the database. The Eve.com flaw discussed previously was the result of parameter tampering.

Cookie Poisoning

Cookie poisoning refers to modifying the data stored in a cookie. Web sites often store cookies on user systems that include user IDs, passwords, account numbers, etc. By changing these values, or poisoning the cookie, malicious users could gain access to accounts that are not their own.

Attackers can also steal users' cookies and gain access to accounts. A large percentage of commercial Web applications, such as Web-based e-mail and online banks, use cookie data for authentication. If the attackers can gain access to the cookie and import it into their own browsers, they can access the user's account without having to enter a user IDs and password or other form of authentication. Granted, the account is only accessible until the session expires (as long as the Web application does provide session timeouts), but the damage is already done. In just a few minutes, the attacker can easily drain a customer's bank account or send malicious, threatening e-mails to the president.

Input Manipulation

Input checking involves the ability to run system commands by manipulating input in HTML forms processed by a Common Gateway Interface (CGI) script. For example, a form that uses a CGI to mail information to another user could be manipulated through data entered in the form to mail the password file of the server to a malicious user or delete all the files on the system.

Buffer Overflow

A buffer overflow is a classic attack technique in which a malicious user sends a large amount of data to a server to crash the system. The system contains a set buffer in which to store this data. If the data received is larger than the buffer, parts of the data overflow onto the stack. If this data is code, the system would execute any code that overflowed onto the stack. An example of a Web application buffer overflow attack again involves HTML forms. If the data in one of the fields on a form is large enough, it could create a buffer overflow condition. Specially malformed form data could cause the server to execute arbitrary code, allowing an attacker to potentially gain complete control of the system.

To learn more about buffer overflows, take a look at "Tao of a Buffer Overflow" by Dildog, available at http://www.cultdeadcow.com/cDc_files/cDc-351/. Other good references include "A Look at the Buffer-Overflow Hack" located at http://www2.linuxjournal.com/lj-issues/issue61/2902.html and "UNIX Security: The Buffer Overflow Problem" at http://www.miaif.lip6.fr/willy/security/.

Direct Access Browsing

Direct access browsing refers to accessing a Web page directly that should require authentication. Web applications that are not properly configured allow malicious users to directly access URLs that could contain sensitive information or cause the company to lose revenue if the page normally requires a fee for viewing.

Web application attacks can cause significant damage to a company's assets, resources, and reputation. Although Web applications increase a company's risk of attack, many solutions exist to help mitigate this risk.

Prevention

The best way to prevent Web application attacks is through education and vigilance. Developers should be educated in secure coding practices, and management should be educated in the risks involved with taking a system live before it has been thoroughly tested. Additionally, administrators and security professionals should be constantly monitoring vendor Web sites, security Web sites, and security mailing lists for new vulnerabilities in the applications and servers used in their Web application. Securityfocus.com, securityportal.com, ntsecurity.com, and linuxsecurity.com are some top security sites that provide excellent information. It does not matter how secure the in-house developed application is if an attacker can gain access to everything through a vulnerability in the database server.

First and foremost with developer education, they should learn never to trust incoming data. A heightened distrust of the end user goes a long way in developing a secure Web application; they should only trust what they control. Because they cannot control the end user, they should view all data input as potentially hostile. Never assume that what was sent to the client's browser is returned unchanged or that the data entered into a Web form is what it should be. Does a form field asking for a customer's address really need to contain a < symbol? Such symbols usually indicate code. Adding filters and input checks significantly reduce the risk of a majority of Web application attacks.

Developers should also include all security measures in the application as they are coding it. Using the anonymous Web server account during development to save time, although each user will authenticate to the application with a username and password, can cause some problems. Bugs might exist in the authentication code, but this will not be discovered until a few days before the application goes live or even after it goes live. Finding bugs at the last minute means the application launch will be delayed or it will be launched with bugs. Neither choice is optimal, so include everything throughout the development process.

If possible, do not use admin or superuser accounts to run the application. Although it may be appealing to run everything as root to save the time of dealing with access rights and permissions, that is asking for trouble. Running everything under a superuser account, the Web application user will have write access to all database tables. By modifying a few URLs with SQL code, a malicious user can easily wipe out the entire database. Following the security principle of least privilege is a must. Least privilege means giving a user the lowest level of permissions necessary to perform a certain task. The user can still enjoy the Web application and the company can feel safe from malicious users, knowing they cannot easily perform illegal operations; their access does not allow it.

Using HTTP GET requests to send sensitive data from the client to the server introduces numerous security holes and should be avoided. GET requests are logged by the Web server in cleartext for the world to read. A credit card number sent to the server by a GET request will be sitting in the Web server logs in cleartext. Using database encryption to protect credit card numbers is useless if all an attacker needs to do is gain access to the Web server logs. SSL does not prevent this issue, either. SSL just encrypts the data during transmission; the GET request will still be logged in cleartext on the Web server. The request might also be stored in the customer's browser history file.

The HTTP POST command should be used instead to send data between the client and the Web server. The POST command uses the HTTP body to pass information, so it is not logged by the Web server. The information is still sent in cleartext, so SSL should be used to prevent network sniffing attacks.

JSP and ASP (*SP) are frequently used in Web application development and often contain hard-coded passwords for connection to directories, databases, etc. Some might think this is okay because the server should process the code and display only the resulting Web page, but numerous vulnerabilities exist that prove this is not always the case. One of the simplest exploits to prove this is the IIS bug that showed the source code of an ASP when ::$DATA was appended to the end of a URL. For example, submitting http://www.site.com/page.asp::$DATA would display the page's source code and all the juicy secrets it contain.

Developers should always be cognizant of HTML code comments and error messages that might leak information. While this will not directly lead to an attack, an attacker can learn enough about the application's architecture to launch a successful attack. For example, including a commented-out connection string that was once part of a server script can give an attacker valuable information.

Error messages also need to be looked at. Some error messages can provide information on the physical path of the Web server that can be used to launch an attack on the system. Other error messages may provide information on the specific database or application servers being used. Overall, error messages do not pose any specific danger, but like commented code, the information gleaned from them can be used to learn the architecture of the application and fine-tune an attack.

Cross-site scripting is a very effective attack that is difficult to defend. The current consensus is to use HTML encoding. With HTML encoding, special characters, such as < and >, are assigned a descriptor: < is &lt and > is &gt. When sent to the browser, the encoded characters will be displayed instead of executed. To prevent the bulletin board attack described previously, input data needs to be encoded. Some products provide tools for this. In IIS, for example, the Server object has HTMLEncode that takes an input string and outputs the data in encoded format.

Secure coding is only one of many components needed to develop a secure Web application. Ideally, security should be discussed, planned for, and included in all phases of application development. When this occurs, the end result will be a stable, secure Web application. Procedures for ongoing monitoring and maintenance

of the Web application should also be developed to help ensure that the security of the application is maintained.

Technology Tools and Solutions

Secure coding practices will help secure the Web application, but it may not be enough. Several tools and applications exist to help audit and secure Web applications.

If a Web application uses CGI scripts, one should scan it with rfplabs' whisker.pl script. This Perl script scans a site for known CGI vulnerabilities. It is freely available at www.wiretrip.net/rfp.

Complete source code reviews are also critical. While it may be too costly to hire a consultant for a full-blown review, several tools exist to help with the process in-house. NuMega (www.numega.com), L0pht (www.l0pht.com/slint.html), ITS4 (www.rstcorp.com/its4), and Lclint (lclint.cs.virginia.edu) all provide source code review programs.

Several products specifically address Web application security (and that number is growing rapidly). Sanctum, Inc.'s AppShield™ product (www.sanctuminc.com) protects Web sites from all the vulnerabilities discussed in this chapter. AppShield acts like a firewall for the Web application, allowing only approved data and requests to be passed to the application. They also have a product, AppScan™, that can be used to test applications for vulnerabilities.

SPI Dynamics' (www.spidynamics.com) WebInspect application scans Web pages, scripts, proprietary code, cookies, and other Web application components for vulnerabilities. WebDefend, like Sanctum's AppShield, provides real-time detection, alert, and response to Web application attacks.

A few other products on the market help protect Web applications from some Web attacks. Entercept and the open-source Saint Jude are new intrusion prevention applications that stop attacks at the operating system level before they occur. These products can protect Web applications from buffer overflow attacks or cross-site scripting that try to invoke processes at the operating system level. Additionally, SecureStack from SecureWave (http://www.securewave.com/products/securestack/index.html) provides buffer overflow protection for Windows NT and 2000 servers.

Summary

Exploiting Web application holes is quickly becoming the attack method of choice to gain access to sensitive information and servers. Numerous methods exist in both commercial and home-grown applications that allow attackers to read information they should not have access to and, in some cases, even allow the attacker to gain complete control of the system.

Many of these holes exist because programmers and application developers are not adequately trained in secure programming practices. Those who are adequately trained do not always implement these practices because the time constraints set to get the product to market quickly preclude taking the time necessary to adequately secure the application.

The main Web application security holes include known vulnerabilities and misconfigurations, hidden fields, backdoor and debug options, cross-site scripting, parameter tampering, cookie poisoning, input manipulation, buffer overflow, and direct access browsing.

To prevent and protect applications from these vulnerabilities, developer education is key. Additionally, a few commercial tools and products exist to help find vulnerabilities and protect applications from being exploited by these vulnerabilities.

In conclusion, Web application attacks, or Web perversion as Sanctum, Inc., calls this phenomenon, are a rapidly growing threat. Education and vigilance are key to protecting the data and resources made accessible to the world by a Web application.

The Perfect Security: A New World Order

Ken Shaurette

A fool does not learn from his mistakes nor the mistakes of others.

OUR FUTURE IS LARGELY A FUNCTION OF OUR PAST, OUR PRESENT, AND THE CHOICES WE MAKE. The past gives us the knowledge and wisdom to know which choices to make, what works, what does not, and what is still unproven. IS security professionals who can look into their crystal balls will see that the future is simply an updated representation of the past. It is not possible to predict the future of technology and its use by our companies, competitors, suppliers, and customers, but one can understand how the issues one deals with today are not all that different from what was being addressed in the past. It is called "planning" rather than "soothsaying."

Regardless of what it is called, forecasting the future of information security is documented and written about in trade magazines and security journals by experts of all kinds. Consider the sampling of past headlines and quotes from various trade magazines in Exhibit 28-1. Contrast the headlines and quotes in Exhibit 28-1 with more recent ones from the past few years in Exhibit 28-2. These may not speak volumes, but as far as being a predictor of the future, notice how the statements in Exhibit 28-1 from 8 or more years ago are not all that different from the ones in Exhibit 28-2 that occurred in the past few years.

Take as an example the 1989 statement in *Computers and Banking* regarding passwords as a defense; it does not say anything about locking the car, just taking the keys. Now in *PC World,* June 2000, experts are asking if the days of the password are numbered. Does that mean that in ten more years one might see a headline like: Headline 2010 — Computers for Everyone Magazine — "Biometrics, smart cards and two factor authentication which last year made archaic password authentication extinct is now seeing its days numbered as DNA and genetic testing begin to become less expensive."

Exhibit 28-1. Headlines and quotes: yesterday.

April 20, 1981	*BusinessWeek*: "Computer Crime — The spreading danger to business"
July 7, 1985	*Express New San Antonio*, *Texas*: "Computer Bandits Hit Banks"
February 4, 1987	*Computerworld*: "Take a Byte Out of Crime, because data is a strategic resource, MIS must learn how to safeguard this precious commodity"
March 1989	*Computers in Banking*: "The password defense is equivalent to taking the car keys with you when you park your car"
February 12, 1990	*Computerworld*: "And the password is obsolete. Are memorized computer passwords passé? Quite a few computer security scientists and security experts think so"
October 14, 1990	*The Independent,* London, England: "Hackers blackmail five banks"
December 1992	*Networking Management*: "Experts warn that network security is not improving"

Exhibit 28-2. Headlines and quotes: today.

September 23, 1996	*Web Week*: "Cyberbanking Pioneers Fight Security, Financial Barriers"
December 7, 2000	*AP*: "Global Cybercrime Laws Lacking, Study Says Few Countries Found to Have Updated Legislation"
June 29, 2000	*PC World*: "Are days of the password numbered? In the future, you'll have no need to remember passwords or PIN numbers"
April 10, 2000	*The Industry Standard*: "Business Under Attack, cyber protest groups reach a new level of aggression and sophistication in their anticorporate campaigns"
December 11, 2000	*APBnews.com*: "A 21-year-old aspiring actor is charged with computer fraud and theft for allegedly hacking into a Hollywood talent agency's Web site, stealing private audition listings and reselling them on the Internet"
December 22, 2000	*ComputerWorld*: "… Hacker breaks Egghead's security shell…hacker had managed to penetrate its computer systems, potentially including the customer databases in which the company stores credit card numbers"

What does all this have to do with building a perfect security world? Only a fool makes the same mistake multiple times. With an understanding of the past, by investigating what has worked and what has not, one can get a fairly accurate representation of what the future may hold. It is often said that history repeats itself, so use it to your advantage like a crystal ball. Many companies have moved away from mainframes, but the security concerns did not go away. In fact, the concerns only became "distributed" to more places.

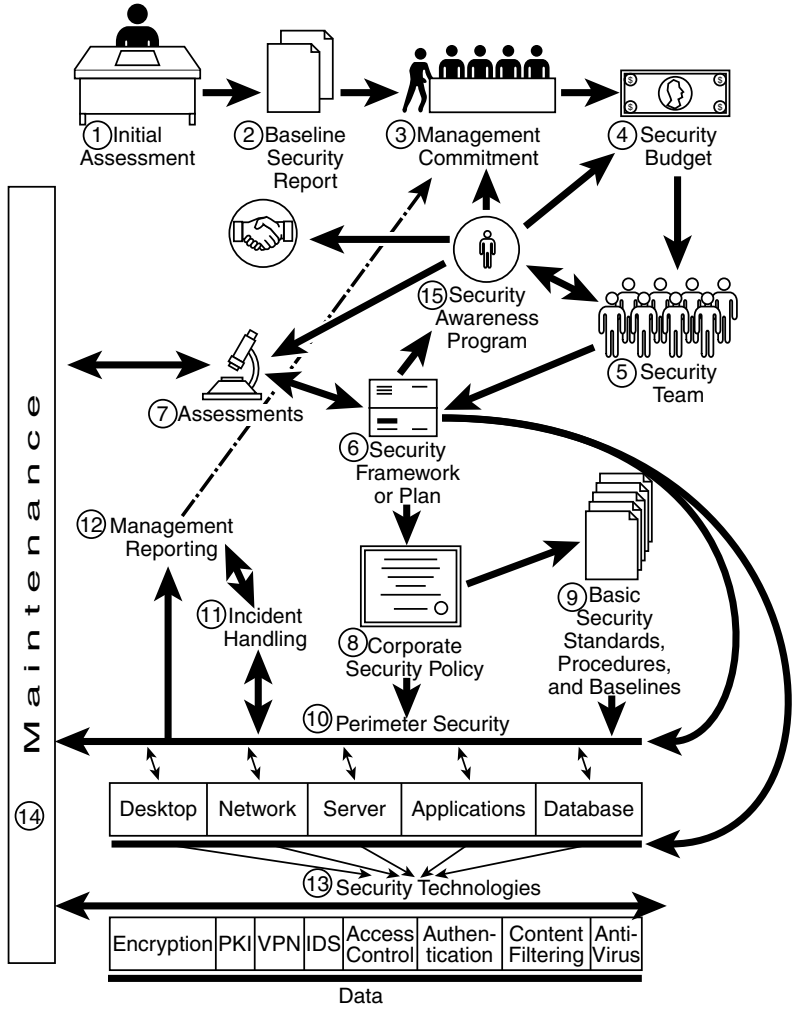
To the security professional, destruction, disclosure, use, and modification (DDUM) of data are all very critical considerations. Data confidentiality remains an issue, as does integrity and availability (CIA triad: confidentiality, integrity, and availability). Of equal importance is the timeliness and validity of data. As Donn B. Parker, retired consultant at Atomic Tangerine, points out, stealing copies of data can make the data of minimal value, but does not impact the integrity of the original data. For example, stealing a trade secret that has not yet been marketed. The owner may still have the original trade secret, but because the information has been stolen and released, that data it is no longer valued the same. How many companies still use copies of production data in test environments? Are the same protections afforded to the test environment as in production? Are the same people who are authorized in production the only ones able to access it in test? Simply possessing the original copy of data may not be enough. Should time-sensitive data such as the company earnings projections be stolen and used to purchase stock or leak to the media, the data could be considered intact, unchanged although invalid. The author would contend that this does not necessarily invalidate the CIA framework, but rather reinforces it. The data was proprietary or confidential to the company. The fact that it was stolen and likely caused the company harm or lost opportunity reinforces the value of protecting it.

It took 20-plus years for the mainframe to reach a point where it was reasonably secure and stable. Then along came the client/server model and moved some of the data and processing closer to the user, reintroducing the same concerns and new vulnerabilities. If we did not learn from what was done during those 20 years of hardening the mainframe, it will take another 20 years to build equal stability for our distributed environments.

SHAPING THE MOLD OF A PERFECT SECURITY WORLD

What has the past taught? By analyzing the past and seeing what has worked and what has not, one can begin to mold future security structures. The mold begins to take form when a self-evaluation of business processes is completed. This consists of investigating corporate business process, how computer processing makes it effective or not, and where new technology can increase productivity and provide new revenue.

While different in that new protocols such as TCP/IP and technologies such as the Internet are replacing the communication medium of Frame Relay, the mainframe, and SNA networks, it is still necessary to protect the data at basically four points: (1) at the point of origin; (2) storage (in memory, in a database or file on disk, or in long-term storage such as backups); (3) at the point of processing (the application); and (4) while it is in transit or on the wire from point to point (the network).



(1) Baseline Assessment
(2) Baseline Security Report
(3) Management Commitment
(4) Security Budget
(5) Security Team
(6) Corporate Security Framework or Plan
(7) Formal Assessment(s)
(8) Corporate Security Policy
(9) Basic Security Standards, Procedures, and Baselines
(10) Perimeter Security
(11) Incident Handling
(12) Management Reporting
(13) Security Technologies
(14) Maintenance
(15) Security Awareness Program

Exhibit 28-3. The perfect security world.

Use Exhibit 28-3 as a reference for the following discussion. Various discussions are numbered and referenced below by a matching number in a circle located in the exhibit.

By performing an initial baseline assessment of the company's computer-processing environment, including business processes and controls, a company will have the basic ingredients for the recipe to their secure world. This baseline assessment (1) is accomplished by asking a series of basic questions (refer to Exhibit 28-4) to provide a baseline of the company's security posture. After completing the initial questions, a company can quickly assess whether it fully understands all of the risks and exposures to the corporate information assets and business-processing environment. This can be accomplished by generating a short concise baseline security report (2) to document the findings and set into motion a security operational plan that will lay the groundwork for minimizing exposures to the business-processing environment.

CAUTION: The initial baseline assessment is an abbreviated version of a more full-blown "risk or security assessment/analysis." Be careful not to be misled that the company is not less secure than it appears. The assessment is only as good as the honesty and knowledge of the people who answer the questions and the experience and knowledge of the person(s) interpreting the answers. For example, just because a company has policies, it does not mean that the policies are being followed or even enforced. It is still necessary to assess at a more detailed level by testing a policy to see if people are in compliance with it.

After the report is complete, a company must deal with the number-one issue to a successful security program: management commitment (3). Each organization will find the level of management commitment very different. It may be easy to get the needed buy-in because of an incident causing financial loss, or it may be difficult because management does not understand all the risks, as the baseline report likely points out. Presenting them in a business context will help management understand. In either case, be prepared by understanding management's business expectations and use the questions (Exhibit 28-4) to educate management to the security concerns.

Until security matters as much to management as the bottom line, the rank-and-file users will not make security policies, guidelines, and procedures a priority. As the security program grows, it will be equally important to have management's buy-in filter throughout all levels of the organization — from executives to line managers.

Remember, security will be cast in the same light as insurance. Security, like insurance, minimizes what one has at risk. A company spends money to have security, because it is *not* willing to accept the risk associated with all of the vulnerabilities that put the business at risk. Security does not increase business profitability unless a company can show that its security provides an advantage over its competition. For most companies, security

Exhibit 28-4. Baseline assessment of company security posture.

1. Are company policies defined to address business use of company resources, covering such things as explicit and appropriate e-mail privacy or Internet usage policy? Are they enforced consistently, if at all?
2. Are the company's operating systems up-to-date with the most current security patches to prevent exposure to known hacking vulnerabilities? Do you know which vulnerabilities can be exploited to access your system?
3. Is your company able to detect a computer crime, and can you gather evidence that can prove to the court, media, or stockholders how the crime was perpetrated and who committed the crime?
4. Does your company allow remote access from home or wireless? Are employees working only from the corporate office? What methods do employees use to access the network? Have they created any methods you are not aware of, such as remote control or modems on a desktop?
5. What is sent across the company network? Do the transmissions include vital or confidential information?
6. Do the information processing safeguards extend protection for the PBX and other telephone attacks?
7. Is there a definition of "incident"? Has an incident response plan been created to handle critical incidents? Does management want to have ability to criminally prosecute on incidents, making it necessary for evidence to stand up in the legal system?
8. Have federal legislation and guidelines such as the 1991 Federal Sentencing Guidelines, the 1996 HIPAA (Health Insurance Portability and Accountability Act), or 1999 Gramm, Leach, Bliley Financial Systems Modernization Act been reviewed for how they apply to the business?
9. Are all users authenticated and authorized to use the company network?
10. Are all of the entry points into the company known and documented? Does that include the ones that exist because of technology, such as modems, personal Internet connections, extranet connectivity, and any others?

does not generate revenue. It is a cost of doing business. Security will be viewed as an expense but must be seen as a necessary cost of doing business. With the dependency today on data, it is no longer an issue of whether a company can afford to provide security measures, but whether the company can afford not to.

Next is a budget (4) to back the efforts of the security program, including appropriate salaries to hire security professionals or the necessary security consultants that can assist in continuing management education, technology evaluation, and can help to complete the building of the security infrastructure. The budget should provide for a team that will coordinate and see to a successful project. The team (5) will build the corporate security framework or plan (6) and present it to management for continued commitment and potential additional budget needs. A security awareness program (15) begins to take shape at this point, simply to keep management

informed of security architecture and funding needs. This communication could be formal or informal. Making it more formal and taking advantage of beginning to form a security awareness program will make the process of keeping management informed consistent and timely. The security awareness program is required throughout the security programs lifecycle, regardless of whether the process is made formal or not. The security awareness program may find it necessary to illustrate examples to management of recent incidents and legislation or regulations to help understand the importance and justify continued budgetary support for security.

The plan should include prioritization of activities to build the perfect security world. Depending on the organization, it may be necessary to use formal assessment(s) (7) to help prioritize actions, build support (management commitment using the security awareness program), or to identify additions or changes to the framework. Initial management commitment and budget to perform the assessment are still required. Enterprisewide risk assessments can be very labor intensive. It is very important to set expectations and a goal for the assessment. This can be difficult, especially if no other assessments have ever been done.

Assessments come in many forms: from the formal enterprisewide assessment that covers the entire corporation and its processing environment to smaller targeted assessments of selected platforms. For example, penetration tests or vulnerability scans can be performed against the company's external access points to find exposures to unauthorized entry. Another example might be an analysis of host operating systems to determine their status and whether they are missing security patches or are improperly configured.

A formal corporate risk assessment could arguably be identified as the number-one requirement before continuing to build a security program. How can a company identify what needs to be done, where the framework is incomplete, what to prioritize, what is missing from policy, essentially what to tell management, without one? It is true that each element in the infrastructure and the risks that pertain to them will affect other elements, and each risk will in turn affect how the complete framework should be managed. However, many companies do not have the luxury of time, money, or commitment to get into an enterprisewide risk assessment. Smaller targeted assessments with a specific goal in mind can be pursued first to get a security process off the ground.

Smaller, less formal assessments can identify gaps in basic security components such as application development, servers, or the network. The simple assessment can help identify basic best practices that are missing but, as a matter of due diligence, should be followed. This gives

the plan a place to start without needing the more complex formal or enterprisewide assessment first. In such a situation, the more formal complete enterprisewide risk assessment can be prioritized for a later date.

LAW AND ORDER: POLICIES, PROCEDURES, STANDARDS, AND GUIDELINES

Every world needs some form of law and order. Corporate security policy (8) provides the backbone, the roadmap or recipe for this new-world order. It defines where a company is and where it wants to go. It establishes baselines to which business processing must adhere. The baselines are the prescribed security controls specified for each component (hardware/software) in the data processing environment in order to achieve a reasonable and consistent level of security throughout the organization. Guidelines are documented in such places as the Common Criteria, BS7799 (British Standard 7799) or in attempts to adopt an international standard modeled after BS7799 (ISO 17799).

Policy and procedures are living documents that change constantly as technology evolves or as business needs change. There are differing layers of policy. The higher-level policy should be reasonably generic and cover such items as "It is the policy of Company X that all computer systems will maintain virus scanning tools with up-to-date virus signatures." This is a management statement of direction. At a lower level are more technical statements or standards that spell out the specific virus scanning software on which the company has standardized. This is the company virus scanning standard. Procedures are the step-by-step actions to support policy and will identify the specifics of how to maintain the virus signatures or use the standard virus tool. These lower-level policies must be maintained and must evolve, always having the support of management and company commitment for consistent enforcement. Higher-level policy is less likely to change but, nonetheless, must be regularly reviewed and even tested to see if it is still applicable to the organization's business model. Policy, just like program code, should have version control, with old versions archived for future reference, management review, and authorizations (sign-off) for implementation. These are the essential components of basic change management.

PERIMETER SECURITY (10)

> The foolish man ignores the desktop or workstation; the wise man considers it one of his toughest challenges.

The surface of this perfect security world is covered with layers of base security solutions native to each platform (a platform for the purposes of this chapter is described as any processing environment providing access to data). There can be layers of security or vulnerability, whichever is

preferred, found at each of the seven layers of the OSI Reference Model: physical, data link, network, transport, session, presentation, and application. The OSI model is a set of protocol layers that enable different computers to interface. Security is aligned with physical, technical or logical, and administrative components. Each different flavor of operating system, each database, and the different network architectures all provide platforms for processing data. Their structures are unique, and each has a different weakness and may require special design and configuration or third-party technology to maximize protection.

Maintaining (14) the perimeter is one of the most overlooked security vulnerabilities, not the maintenance in and of itself, but not keeping software or hardware current or applying known security patches. In general, the tools used by attackers search for known vulnerabilities in a platform. If they are known by vulnerability scanning software, most often there is also a fix for them and simple maintenance can eliminate these exposures. Management reporting, represented in Exhibit 28-3 as (12), provides feedback to management to keep them informed and aware of security from budgetary needs for maintenance to information on incidents. This level of reporting sustains management commitment. If management never hears anything about where the money committed in the budget is going, they are less likely to support additional budget dollars when needed. This restarts the whole cycle over again. Upgrades could be hardware, the latest software release, security-specific patches, or a new database or enhanced routers and other technology for the network.

THE CRUST: DESKTOP

The desktop is nearest the eighth vulnerability, often considered the weakest point in any network, "people." Surprisingly, it is often the last vulnerability point considered for improved security. This is most often because the desktop is most numerous and represents a largely uncontrollable entry point. In the past few years, more and more security technologies and third-party vendor solutions have become dependent on touching the desktop. Security awareness (15) is the primary solution in the security process that directs attention to the people vulnerability.

Consider authentication, authorization, and accounting as the main components of any security framework. Authorization depends on proper authentication of the person(s) that use(s) the desktop. Access control systems, such as biometrics, smart cards, tokens, or even PKI, all depend on client implementations and often dependent on client interaction that touches the desktop. PKI, for example, without other controls does not provide authentication of a user; it authenticates a workstation or laptop or whatever location the key has been stored on, not necessarily determining whether the key is in an authorized person's hands. Actual authentication

routines can be performed at a server, but often client code is required to link with the desktop applications or operating system to create the required integration. As noted earlier, because there often are so many desktops, the cost of implementation can be high, both because of sheer volume and because successful implementation often requires cooperation from the end user.

It has historically *not* been acceptable in many companies to expect an end user to perform any actions to help improve security. This is slowly changing as end users get more computer-savvy and as technology becomes friendlier. This can also be improved using a solid security awareness program (15) to keep them informed and aware of policy, standards, and procedures, as well as educated on technology and proper use of the platforms.

THE HIGHWAYS AND BYWAYS: NETWORK

Roads provide connectivity between communities. Roads provide a good analogy to networks in many ways. In the data processing world, the network provides communication (connectivity) between the user and data. Regardless of where the data is stored or where the user is physically located, the network can provide the user access.

In northern states, it is often said that there are only two seasons: winter and road construction. Like roads, the network requires continual monitoring and maintenance to keep it healthy and running smoothly and securely. Similar to the way a properly constructed road provides safe transportation, a network that is properly constructed and configured can provide safe and secure data delivery. However, a smooth functioning network does not necessarily represent a secure one. In general, it is the function of the network to provide transmission of bits from one point to another.

Technologies (13) such as IDS (intrusion detection system), VPN (virtual private network), and general network encryption all enhance the security of the network. An IDS does this by alerting appropriate personnel on incidents or reporting on suspicious activity. A VPN allows the use of a public network — the Internet — to connect networks together in a secure way. To put it in simple terms, a "virtual tunnel" using encryption protocols such as 3DES or IPSec is established between the networks allowing secure transmission between them. This prevents packet sniffing or password theft, provided it is properly configured. Essentially, a VPN provides a trusted tunnel using encryption between two points over the unsecured Internet Protocol. Link encryption by itself will camouflage the data while it is on the wire from storage to the user.

CITIES, VILLAGES, AND TOWNS: THE SERVERS AND HOSTS

Today, hosts or servers come in many flavors or operating environments. In the past, there was MVS, VM, DEC-VAX, and other mainframe-oriented systems. The mainframe had the power to house all the functionality of server and desktop, as well as provide the services for the other "platforms" that today are often spread (distributed) across multiple servers and even on different operating systems. Today's mainframes, now better described as enterprise servers, and some of the more powerful servers provide virtual separation by platform, but all reside on the same physical box. Each platform provides specialty services such as database server, Web server, application server, and even security server. Each might be on physically different hardware with differing operating systems. This is often done for performance tuning, definitely not for enhancing security controls.

Hosts or servers have basically four layers of vulnerability:

1. hardware, (the physical box, including the internal components, memory, and CPU, which can have special configurations
2. basic I/O (input/output) or firmware, which provides the CPU with data to process or puts the information out to storage or printer devices
3. kernel or nucleus of the operating system, which like the city is the downtown area, a very critical part of the city; it makes the rest of the city run
4. the operating system interfaces or shells, such as command line interfaces or the graphical user interfaces that provide user friendliness

The physical hardware and operating systems are like the buildings in the city. Different buildings have structures to provide services that match their architecture. The bank, built extra strong, protects the money with a vault; the restaurant drive-up has a special window and microphone to allow ease of access from the auto. Servers with special operating systems can be hardened to provide the protection of a firewall, or can have an open architecture that supports public data and easy access for students or to public information.

Piecing these layers together into a seamless, appropriately secure computing platform represents the challenge. Security must be considered to protect the internal components against physical theft or tampering. Alarm devices, physical locking mechanisms, or, more simply, a controlled computer room with proper environmental and access controls can protect the hardware. Access control systems from various third-party vendors can be purchased to enhance the base security of the kernel or access to the peripheral devices and restrict access using control lists of who has permission to execute commands or be able to select from a menu.

RUNNING THE CITIES, TOWNS, AND VILLAGES: APPLICATIONS

The application is the layer of the platform that is like the processes a city uses to make it run, such as holding civil court to collect fees or billing landowners for taxes. It is the accounts payable, the receivable, billing, inventory handling, shipping, etc. part of the environment. The application is heavily dependent on the operating system and database and often designed with those layers in mind in order to provide a seamless and secure processing environment. An application that integrates with the operating system (OS) or is tightly integrated to the database (DB) tends to be the most flexible and can leverage the unique features of the specific OS or DB. This would be in place of the application providing its own authentication and audit. An integrated application overall becomes the one that a user will prefer to use because it does not introduce additional authentication layers or complexity to the business process.

If the application depends on its own authentication, it introduces additional exposures and, most likely, another authentication routine for the end user. Some might consider another password and additional authentication more security, but the added complexity and human natures involvement with yet another password scheme will generally render this layer a waste of time because the user will choose poor passwords or write them down. The operating system already provides authentication; why not trust it and avoid authenticating again?

The special business function provided by the application often requires application-level security specific to the functions built into the application. These functions may be necessary to control which user can perform what specific functions. Rather than provide unique authentication unto itself to identify the user, the application should trust that the user has already authenticated at the operating system level. The application can have its own authorization structure to control what functions a user can perform, but should interface with the operating system or the database to perform the authentication. The methods used to allow this are APIs (application programming interfaces) in the application or in the operating system. These provide customized functionality such as integration between the application and the operating system or with third-party vendor technologies such as smart cards, tokens, public encryption keys, or biometrics.

LIBRARIES AND SCHOOLS: THE DATABASES

The database is the holder of the data or the processed data (information). It is the point closest to the entity that most companies are trying to protect, the data. The database can hold the security information,

application controls, metadata or data about data, and simply the basic data itself.

The database, like the application, depends on authentication to identify who a user is so that the proper association can be made to what they are authorized to access. Authorization then identifies the user's access to the data elements (tables, rows, fields, columns); what level of access they have (update, insert, delete); as well as determining any access to database utilities (import, export, load, unload, compress). Even if application security is tightly implemented, without carefully controlling authentication and authorization in the database to only proper users with access to the application functions, the database services provide direct access to the data. This creates a "backdoor" or vulnerability, which is often found by auditors. These services "go around" the application security and do not have any of the edits or controls that might have been built into the application. Because of this, the person desiring access to query or modify the data is not likely to use the application; they will take a more direct route and access the data using other tools. Most relational database management systems (RDBMS) allow languages such as SQL or other direct access reporting tools to manipulate the data directly. The PeopleSoft 7.0 application system, for example, has very sound security built into the application, including authentication and authorization; but unless the RDBMS accounts that the PeopleSoft architecture depends on are properly configured and these database accounts are appropriately managed, access to all PeopleSoft data can be compromised via the database directly. Another example is a poorly designed Web application. The application may require a single generic database account for all access by any user of the application. It might require a fixed password that can be hardcoded in the application programs in order to connect to the database. Compromising that one account compromises the entire application.

One mechanism to address this weakness is to use restricted shells (UNIX) within the functionality of the operating system to control what an account can do at the operating system level. For example, the DB2UDB database in a UNIX environment counts on the native operating system to perform the authentication (password checking) and manage which user account is in which groups. The account never actually logs into the operating system. The account can actually be disabled from performing any functions at that level by assigning a "dummy" or null shell for the accounts default shell. Doing this causes no UNIX shell to be opened and any session with the operating system to terminate, but the password checking will still occur and connectivity to the database will still work. This is commonly used by many UNIX system services or daemons.

Base ORACLE, another popular RDBMS, can have quite weak native security. Third-party technologies such as SQLSECURE from Braintree Systems

(Pentasafe) can enhance the database security authentication, authorization, and auditing features. These tools, for the database or access control systems for the operating system, antivirus tools for desktop or server, encryption (public key infrastructures) or virtual private networks (VPNs), and intrusion detection systems (IDSs), whether host or network based, are all security technologies (13) that enhance the level of security for the base environment and help base platforms improve on their weaknesses. However, improperly maintained (14) technologies introduce not only added complexity, but also new places for vulnerabilities — just like a poorly maintained operating environment.

THE CYCLE OF SECURITY: SUMMARY

> Learn from the mistakes of others. You will not live long enough to make all of them yourself.

What has the past taught us? One needs to learn from past mistakes. Not patching or performing maintenance on hardware and software leaves them vulnerable to the same unauthorized access that befell those before us. Known vulnerabilities are a primary cause of unauthorized access and jeopardize the stability of the processing environment.

Many companies have moved the processing of data from the mainframe to distributed systems, but the security controls did not go with it. The new environment requires the same attention to controls and audit as was available on the mainframe. Use the concepts that were perfected in the environment of old to construct new processing environments so that it does not take so long to get it right.

There are eight layers of vulnerability. These layers fit neatly into physical, technical, and administrative layers. Detail vulnerabilities can be found in each layers of the OSI Reference Model: physical, data link, network, transport, session, presentation, and application, plus the toughest to control layer of vulnerability, the operator or user, who is probably the greatest exposure.

Creating a perfect security world requires attention to all of the layers that make up a business-processing model. Each layer can introduce unique vulnerabilities. The complete solution is not just about technology. Administration, management, and process are all important parts of the security solution. Understanding the overall security process can help build a comprehensive security program. The total program will have management's commitment, an adequate budget, and a roadmap called policy with a security awareness program that educates, communicates, and ties everything together by providing feedback to the operator as well as management to keep the cycle of security flowing.

92

Security for XML and Other Metadata Languages

William Hugh Murray, CISSP

When the author was a beardless boy, he worked as a punched-card machine operator. These were primitive information processing machines in which the information was stored in the form of holes punched in paper cards. Although paper was relatively cheap by historical standards, by modern standards it was very expensive storage. For example, a gigabyte of storage in punched paper would fill the average room from floor to ceiling, wall to wall, and corner to corner. It was dear in another sense; that is, there was a limit to the size of a record. A "unit record" was limited to 80 characters when recorded in Hollerith code. This code in this media could be read serially at about 10 to 15 characters per second. In parallel, it might be read at 8 to 12 thousand characters per minute.

As a consequence, application designers often used very dense encoding. For example, the year in a date was often stored as a single digit; two digits when the application permitted it. This was the origin of the famous Y2K problem. As the Y2K problem resolved, it was often thought of as a programming logic problem. That is, the program would not process years stored as four digits and might interpret 2000 as being earlier than 1999 rather than later. However, it was also a quality of data problem. When the year was encoded as one or two digits, information was often permanently lost. In fixing the problem, one often had to guess as to what the real data was.

The meaning of a character in a punched-card record was determined by its position in the record. For example, an account number might be recorded in columns 1 to 8 of the card. Punched-card operators of large stable applications could often understand the records from that application by looking at the color of the card and determine what information was stored in which columns by looking at the face of the card where the fields were delineated and identified. When dealing with small or novel applications, one often had to refer to a "card layout" recorded on a separate piece of paper and stored in a binder on the shelf. Because this piece of paper was essential in understanding the data, its loss could result in loss of the ability to comprehend the data.

The name of the file was often encoded in the color of the card, and the name of the field in its position in the card. The codebook might have been printed on the face of the card or it might have been stored separately. In any case, it was available to the operators, but not to the machine. That is, the data about the data was not machine-readable and could not be used by it.

This positional encoding of the meaning of information and separate recording of its identity on a piece of paper carried over into early computer programming. Therefore, when starting to resolve the Y2K problem, one could not rely on the machine to identify where instances of the problem might appear, but had to refer to sources external to the programs and the data.

MetaData

In modern parlance, this data about the data is called metadata. Metadata is used to permit communication about the data to take place between programs that do not otherwise know about each other. Database schemas, style sheets, tagged languages, and even the data definition section of COBOL are all examples of metadata. Because storage is now both fast and cheap, modern practice calls for the storage of this metadata with the data that it describes. In many applications and protocols, the metadata is transmitted with the data. A good example is electronic data interchange (EDI), in which fields carry their meaning or intended use in tags.

Good practice says that one never stores or moves the data without the metadata. Preferred security practice says that the metadata should be tightly bound to the data, as in a database, so as to resist unintended change and to make any change obvious. In object-oriented computing, the data, its meaning, and all of the operations that can be performed upon it may be bound into a single object. This object resists both arbitrary changes and misunderstanding.

Tagged Languages

One form of metadata is the tag. A tag is a specially formatted field that contains information about the data. It is associated with the data to which it refers by position; that is, the tag precedes the data. Optionally but often, the tag refers to everything after it and before a corresponding end tag.

XML is a tagged language. In this regard, it is similar to HTML, EDI, and GML. A tag is a variable that carries information about the data with which it is contextually associated. A tag is metadata. To a limited degree, tags are reserved words. Only limited reservation is required because, as in these other tagged languages, tags are distinguished from data by some convention. For example, tags can be distinguished by bracketing them with the left and right pointing arrows, <tagname>, or beginning them with the colon, :tagname. Each tag has an associated end tag that is similarly distinguished; for example, by beginning the end tag with the left pointing arrow followed by a slash, </tagname> or the colon followed by the letter "e," :etagname. The use of end tags eliminates the need for a length attribute for the data. Tags are often nested. For example, the tags for name and address may appear inside a tag for name and address.

A tagged language is a set of tag definitions. Such a set, language, dialect, or schema is defined in a Document Type Definition object. This schema can be encapsulated in the object that it describes, or it can be associated with it by reference, context, or default. These language definitions can be, and usually are, nested. This provides maximum functionality and flexibility but may cause confusion.

The concept of "markup" comes from editing and publishing. The author submits a document to the editor who "marks up" the text to communicate with both the author and the printer or composer. One early tagged language was the Generalized Markup Language, perhaps the prototypical markup language. However, the concept of markup suggests something that is done in a separate step to add value or information to the original. Many of the tagged languages called markup languages are really not markup languages in that special sense.

As with most languages, tagged languages provide for special usage. They provide for special vocabularies that may be meaningful only in a special context. For example, the meaning of the word "security" is different when used in financial services than when it is used in information technology. Similarly, EDI uses a number of different vocabularies, including X12, EDIFACT, TRADACOMS, that are applicable only in their intended applications.

The eXtensible Markup Language

XML is a language for describing data elements. It describes the attributes of the data and identifies its intended meaning and use. It consists of a set of tags that are associated with each data element and a description that decodes the tag. Keep in mind the analogies of a database schema and a record layout. Also keep in mind the limitations of these languages. And think of the analogy of HTML; as HTML says this is how to display or print it, XML says these are its attributes and this is what it means. XML is not magic.

XML is an open language. That is why it is called extensible. Of course, all programming languages are extensible to some degree or another. The dynamic HTML bears only a family resemblance to the HTML of a decade ago. Current browsers are dynamically extensible through the use of plug-ins and the Dynamic Object

Model (DOM). Modern HTML is dynamically extensible, extensible on-the-fly. The capabilities of the interpreter are dynamically extended through the use of plug-ins, applets, and similar mechanisms.

The owner of the object in which XML is used is permitted to define arbitrary tags of his or her own choice and embed their definition in the object. The meaning and attributes of a new tag are described in old tags. XML is a dialect of the Standard Generalized Markup Language, developed by IBM and adopted as an ISO standard. XML is the parent of a number of dialects, including cXML (Commerce XML), VXML (Voice XML), and even MSXML (Microsoft XML). There can be dialects for industries, applications, and even services. However, the value of any dialect is a function of the number of parties that speak it.

XML is a global language. That is to say, it has global schemas that go across enterprises, industries, and even national boundaries. These schemas represent broad prior agreement between users and applications on the meaning and use of data. The scope of the vocabulary of XML can be contrasted to that of programming languages such as COBOL where the data description is usually limited to an enterprise and often to a single program; where the base set of verbs is common across enterprises but there are no common nouns.

XML implements the concept of namespaces. That is, it provides for more than one agreement between a name and its meaning. The intended namespace is indicated by the name of the space, followed by a colon in front of the tagname (<ns:tagname>). There can be broad agreement on a relatively small vocabulary with many special vocabularies used only in a limited context.

XML is a declarative language. It makes flat statements. These statements are interpreted; they are not procedural. It says what is rather than what to do. However, one must keep in mind that tagnames can encapsulate arbitrary definitions that are the equivalent of arbitrary procedures.

XML is an interpreted language. Like BASIC, Java, and HTML, it is interpreted by an application. However, to provide for consistency and to make XML-aware applications easier to build, most will use a standard parser and a standard definition or schema.

It is recursive. The XML schema, the object that defines XML, is written in XML. It can include definitions by reference. For example, it can reference definition by uniform resource locator (URL). Indeed, because it increases the probability that the intended definition of the tag will be found, this style of use is not only common, but also frequently recommended. Of course, from the perspective of the owner of the data, this is safe; it ensures the owner that the tags will be interpreted using the definitions that the owner intended. From the perspective of the recipient of the data, it may simply be one more level of indirection (i.e., sleight of hand) to worry about. The good thing about this is that URLs begin with a domain name. (Keep in mind that, while domain names are very reliable, they can be spoofed.) While it is possible, even usual, for the meaning of the metadata to be stored in a separate object, local definition may override the global definition.

It supports "typed" data, that is, data types on which only a specified set of operations is legal. However, as with all properties of XML-defined data, it is the application, not the language itself that prevents arbitrary operations on the data. For example:

```
<simpleType name="nameType">
 <restriction base="string">
 <maxLength value="32"/>
 </restriction>
</simpleType>
```

sets the maximum length of "nameType" equal to 32. Similar metadata could impose other restrictions or define other attributes such as character set, case, set or range of valid values, decimal placement, or any other attribute or restriction.

XML and other tagged metadata languages are not tightly bound to the data. That is to say, anyone who is privileged to change the data may be privileged to change the metadata. Anyone who is privileged to change the tag can separate it from the data. This loose binding can be contrasted with a database in which changing the metadata requires a different set of privileges than changing the data itself (see Exhibit 92.1).

XML Capabilities and Limitations

Every tool has both capabilities, things that it can do, and limitations. The limitations may be inherent in the very concept of the tool (e.g., screwdrivers are not useful for driving nails) or they may be implementation induced (e.g., the handle of the screwdriver is not sufficiently bound to the bit). The tool may not be suitable for the application (e.g., the screwdriver is too large or too small for the screw). One does not use Howitzers

EXHIBIT 92.1 The E-Wallet: An Example

A good example of the use of metadata in communication is the E-wallet application. Its owner uses the e-wallet to store and use electronic credentials. These include things such as name and address, user IDs and passwords, credit card numbers, etc. Because all of this information is sensitive to disclosure, it is usually stored in a database. The database can hide the data and associate it with its metadata, its intended meaning and use. Alternatively, the data could be stored in a flat file using tags for the metadata and file encryption to hide the data in storage when not in use.

The user employs the E-wallet application to present the credentials in useful ways. For example, suppose that the user has decided to make a purchase from an online merchant. After making a selection, the user presses the checkout button on the screen and is presented with the checkout screen. This screen asks for name and billing address, name and shipping address, and charge information. The user invokes the e-wallet application to complete this screen.

The E-wallet presents the data stored in it and the user clicks and drags it to the appropriate fields on the checkout screen. The user knows what information to put in what places on the screen because the fields are labeled. These labels are put on the screen using HTML. While they are visible to the user, they are not visible to the e-wallet application. Therefore, the user must do the mapping between the fields in the E-wallet and those on the checkout screen. Although this process is flexible, it is also time-consuming. Although it ultimately produces the intended results, it relies on feedback and some intermediate error correction. When the screen is completed to the user's satisfaction, the user presses the Submit button. At this point, the screen is returned to the merchant where the merchant's computer verifies it further and might initiate another round of error correction.

If, in addition to labeling the fields on the screen with HTML, the merchant also labeled them with XML, then an XML-aware E-wallet could automatically complete part of the checkout screen for the user. If the checkout screen requests billing information, the E-wallet will look to see if it has the information to complete that section. In the likely case that it has more than one choice, it will present the choices to the user and the user will choose one. When the screen is completed to the user's satisfaction, the user will press the Submit button. When the screen is returned to the merchant, the data is suitably labeled with his XML so that his XML-aware applications and those of his trading partners (e.g., his credit card transaction service) can validate the data.

The use of XML has not changed the application or its appearance to the user. It has not changed the data in the application or its meaning. It has simply facilitated the communication between XML-aware applications. It has made the communication between the applications more automatic. Data is stored where it is supposed to be, controlled as it is supposed to be, and communicated as it is supposed to be. The applications behave more automatically and the opportunity for error is reduced. Notice that the applications of some merchants, most notably Amazon, achieve the same degree of automation. However, they do it at the cost of replicating the data and storing it in the wrong place that is, user data is stored on the merchant system. This can and has led to compromises of that data. While one might argue that the data is better protected on the merchant's server than on the customer's client, the aggregation of data across multiple users is also a more attractive target.

Just as there are multiple browsers, there will be multiple E-wallet applications. As the requirement for the browser is that it recognizes HTML, the requirement for the E-wallet is to speak the same dialect of XML as the merchant's application. To make sure that it speaks the same dialect of XML as the merchant, the E-wallet may speak multiple XML dialects, similar to the way that browser applications speak multiple encryption algorithms.

Notice that the merchant's application could request information from the user's E-wallet that it does not display on the screen and which the user does not intend to provide. The user relies on the behavior of his application, the E-wallet, to send only what he authorizes.

As the merchant's application might attempt to exploit the E-wallet or its data, the user might attempt to alter the tags sent by the merchant in an attempt to dupe the merchant. The merchant relies on his application to protect him from such duping.

to kill flies. This section discusses the capabilities, uses, misuses, abuses, and limitations of XML and similar metadata languages.

XML is metadata. It is data about data. Its role is similar to that of the schema in a database. Its fundamental role is to carry the identity, meaning, and intent of the data. It is neither a security tool nor is it intrinsically a vulnerability. From a security point of view, its intrinsic role is to support communication and reduce error. The potentially hostile or threatening aspects of XML are not those unique to it, but rather those that it shares with other languages, metadata, tagged and otherwise; a language that usually communicates truth can be used to lie.

Exhibit 92.2 Web Mail: An Example

"Web mail" turns normal two-tier client/server e-mail into a three-tier client/server application. Perhaps the most well-known example is Microsoft's Hotmail. However, other portals such as Excite and Yahoo! have their own implementations. Many Internet service providers have an implementation that permits their mail users to access their post office from an arbitrary machine, from behind a firewall (that permits HTTP but restricts mail), or from a public kiosk.

In Web mail, the message is actually decoded and handled on the middle tier. Then the message is displayed to the user on his workstation by his Web browser. In one implementation, the middle tier failed to recognize the tags and simply passed them through to the Web browser. An attacker exploited this capability to use the browser to pop up a window labeled as the Web mail log-on window with prompts for the username and passphrase. Although mature users would not respond to a log-on prompt that they were not expecting, novice users did. Although all applications behaved as intended, the attacker used them to produce a result that duped the user. Web mail enabled the tags to escape the mail environment where they were safe, merely text, into the browser environment in which they were rendered in a misleading way.

This exploit illustrates an important characteristic of languages like XML that is easy to overlook when discussing them: they are transparent to the end user. The end user does not even know that they exist, much less what they say, how they carry meaning to his system, his application, or to himself.

People have been using and living with HTML for almost a decade. As XML is defined in XML, so is HTML 4.0, the vocabulary known as XHTML. (Recursion is often confusing and sometimes even scary.) People have been using EDI tags for almost a generation. Although they are now a subset of XML, all of our experience with them is still valid.

Perhaps the aspect of XML that is the source of most security concerns is that it is used with "push" technology; that is, the tags that describe the data come with the data. Moreover, the schema for interpreting the data may also be included. All of this often happens without very much knowledge or intent on the part of the recipient or user. However, the meaning will be interpreted on the receiving system. Although it causes concern, it is as it should be. Only the sender of the data knows the intended meaning.

The fundamental responsibility for security in XML rests with the interpreter. As the browser hides the file system from HTML, the application must hide it from XML. As the browser decides how the HTML tag is to be rendered, so the application decides on the meaning of the XML tag. However, in doing so, it may rely on a called parser to help it deal with the tags. To the extent that the application relies on the parser, it must be sure that the one that it is using is the one that it expects. While normal practice permits a program to rely on the environment to vouch for the identity of a called program, good security practice may require that the application validates the identity of the parser, even to the extent of checking its digital signature.

Similar to many interpreted languages, XML can call escape mechanisms that permit it to pass instructions to the environment or context in which the user or receiver expects it to be interpreted. This may be the most serious exposure in XML, but it is not unique to XML. Almost all programming or data description languages include such an escape mechanism. These escape mechanisms have the potential to convert what the user thinks of as data into procedure (see Exhibit 92.2.)

While most of the use of such mechanisms will be benign, they have the potential to be used maliciously. The escape mechanisms included in Word, Excel, and Visual Basic have been widely exploited by viruses to get themselves executed, to get access to storage in which to place replicas, and to display misleading information to the user.

World Wide Web Security

While XML will have many applications other than the World Wide Web, this is the application of both interest and importance. As discussed, XML does little to aggravate the security of the Web. It is true that it can be used to dupe both users and applications. However, the vulnerabilities that are exploited can as easily be exploited using other languages or methods. By making the intent and meaning of the data more explicit, it may facilitate intelligence gathering.

On the other hand, it has the potential to improve communication and reduce errors. XML is being used to extend the capabilities of Web clients and servers so as to increase the security of their applications. While these capabilities might be achieved in a variety of other ways, they are being implemented using XML. That they are being implemented using a metadata language demonstrates one value of such languages. These

implementations have the potential to bring to security many of the advantages of metadata languages, including interoperability that is both platform and transport independent. However, keep in mind that these definitions are about the use of XML for security rather than about the security of XML.

Control of Access to XML Objects

One such application is the control of access to documents or arbitrary objects stored on Web servers in a manner that is analogous to the control of access to database objects. In client/server applications, XML can be analogous to an SQL request. That is, it is used to specify the data that is being requested. As the database server limits access to the data that it stores and serves up, so the server responding to an XML request can control access to the data that it serves.

In SQL, the fundamental object of request and control is a table. However, most database servers will also provide more granular control. For example, they may provide for discretionary access control over rows, columns, or even cells. Many can exercise control over arbitrary combinations of data called views. Notice that discretionary access control over the data is a feature of the database manager rather than of the language or schema. Notice also that the data is bound to the schema only when it is in a database manager. Once the data is served up by the database manager, then trusted paths and processes may be required to preserve its integrity.

In XML, as in HTML, the fundamental object of access control is the document. For this purpose, the document is analogous to the database table. Almost all servers can restrict access to some pages. While this capability is rarely used, many provide discretionary access control to pages, that is, the ability to grant some users access to a page while denying it to others. For example, the Apache Web server permits the manager to grant or restrict access to named documents to specified users, user groups, IP addresses, or address/user pairs. Notice that as a database administrator can exercise more granular access control by naming multiple views of the same data, so too can the administrator of a server exercise more granular control by creating multiple documents.

However, tags are used to specify more granular objects than documents. This raises the possibility of more granular access control. As a database manager may provide more granular access control than a table, a server may provide more granular access control than a page. If it is going to do this at all, it can do it to the level of any tagged object. While administratively one might prefer large objects, from the perspective of the control mechanism, one tag looks pretty much like any other. Damiani et al.[1] have demonstrated such a mechanism.

Process-to-Process Authentication

On the Web, particularly in E-commerce applications, it is often necessary for a client process to demonstrate its identity to a server process. These *bona fides* are often obtained from a trusted third party or parties. Such a demonstration may involve the exchange of data in such a way that the credentials cannot be forged or replayed. The protocols for such exchanges are well worked out. These protocols lend themselves to being described in structured data. In XML, such exchanges involve two schemas: one for the credentials themselves and another for requesting them.

A dialect of XML, authXML, has been proposed for this application. It defines formats for data to assert a claim of identity and for evidence to support that claim.

Process-to-Process Integrity

Similarly, in E-commerce applications, it is necessary to be able to digitally sign transactions so as to demonstrate their origin and content. This requires tags for the transaction itself, the signature, and the certificate. S^{2}ML, the Security Services Markup Language, provides a common language for the sharing of security services between companies engaged in B2B and B2C transactions.

Recommendations

1. *Identify and tag your own data.* Keep tags with your data. Although useful and used for communication, metadata is primarily for the use of the owners of the data.

2. *Bind your metadata to your data.* Use database managers, access-controlled storage, encryption, trusted applications, trusted systems, and trusted paths.
3. *Verify what you rely on.* This is the fundamental rule of security in the modern networked world. If relying on an object description, then be sure that you are using that description. If relying on an object not to have a script hidden in it, then be sure to scan for scripts.
4. *Accept tags only from reliable sources.* Do not place more reliance on tags from a source than you would on any other data from that source. While you might reject data without tags from a source, do not accept data with tags where you might not accept the data without the tags.
5. *Reject data with unexpected tags.* Do not pass the tags on. Do not strip them off and pass the data on.
6. *Include tags in logs and journals.* Not only will this improve the integrity and usability of the logs and journals, but it will improve accountability.
7. *Use the security tags where indicated and useful.*
8. *Communicate these recommendations to application developers and managers in appropriate standards, procedures, and enforcement mechanisms.* Although these measures are essential to the safe use of metadata, their use and control is usually in the hands of those with other priorities.
9. *Focus on the result seen by the end user.* After all is said and done, the security of the application will reside in what the end user understands and does.

Conclusion

HTML and similar metadata languages have given us levels of interoperability that were not dreamed of a decade ago. As the number of interoperable systems on the Internet has risen linearly, the value to the users has risen exponentially. XML promises us another order-of-magnitude increase in that interoperability. Not only will it help create interoperability between clients and servers on the Internet, but it will also improve interoperability among arbitrary objects and processes wherever located. By conserving and communicating the meaning and intent of data, it will increase its utility and value. Not since the advent of COBOL has there been a tool with such promise; this promise is far more likely to be realized and may be realized on a grand scale.

However, as with any new tool, the value of XML will depend, in large part, on one's skill in using it. As with any idea, its value will depend on one's understanding of it. As with any new technology, its value may be limited by fear and ignorance.

As with any new tool, one must understand both its capabilities and its limitations. Few things in information technology have caused as many problems as using tools without proper regard for their limitations.

Although the use of XML will often be outside the purview of the information security professional, hardly anyone else will be concerned about its limitations, misuse, or abuse. If the enterprise suffers losses because of limitations, misuse, or abuse, it is likely to hold us accountable. If the fundamental idea should become tarnished because of such limitations, misuse, or abuse, we will all be poorer for it.

Note

1. http://www9.org/w9cdrom/419/419.html. Design and Implementation of an Access Control Processor for XML Documents. Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati.

93

XML and Information Security

Samuel C. McClintock

Information technology changes on a daily basis, and almost every year the world is presented with a new "holy grail" of the information age. Into this fray comes the eXtensible Markup Language (XML), one of our newest Holy Grails that promises everlasting life, or least ever-usable data. At its heart is a simple text-based language that can describe complex data structures. Because of its simplicity, almost any computer has the power to use XML and almost every type of network can transmit it. XML has also received very broad support from almost all the major vendors and many of the smaller ones, allowing almost any computer system to manipulate XML without major modifications to the existing infrastructure. So what are the problems?

Well, the basic problems have never changed — the Internet is as insecure as it ever was, technology moves at breakneck speeds, some people make mistakes, others steal or vandalize information, and garbage in-garbage out still applies to every computer system ever made. XML does not change any of this, but it does provide one more avenue of abuse. XML becomes one more consideration to integrate with ongoing security efforts, and XML manages to add a few more security wrinkles of its own.

Fortunately, the fact that many of the information security issues of XML are common to existing problems makes it easy to adapt our current security practices. XML, by its very nature, also allows us to create "extensions" of the language to specifically target different security solutions for XML, such as encryption. Major vendors have already designed security around XML and have proposed new standards for encryption and digital signatures in XML. However, the latest wave of solutions is by no means complete. Programmers, database administrators, and executives must pay attention to the fact that XML will make the data easier to read, organize, and disseminate, that XML does not effectively change any of the existing problems, and plan their security appropriately.

XML will continue to make rapid advances throughout all of our information technology. Not only will tomorrow's information security professionals have to protect resources that use XML, they will also see XML integrate into many of the security tools they use. Thus, information security professionals need to understand both XML and the security issues surrounding XML applications.

XML Basics

To understand XML, and the security issues of XML, a little background is in order. For the information security professional, this could be seen as getting to know thy enemy, getting to know thy friend, or for the truly advanced, one more step on the familiar road to technologically induced schizophrenia.

Why Not HTML?

HyperText Markup Language (HTML) is one of the foundations of the World Wide Web. HTML is extremely simple and easy to use and has become one of the most successful publishing languages in the world. Even non-programmers can learn the rudiments of HTML, the codes or "tags" that define what a document will

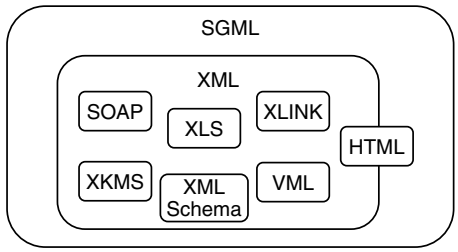


EXHIBIT 93.1 The structure of SGML and XML.

look like, and produce Web sites. But HTML has become a victim of its own success, and the ease of HTML use has come up against limitations born of the growth and expectations of the Web:

- HTML is not extensible so it is not possible to define tags for specific requirements. If this is not bad enough, different browser vendors invent their own extensions for new features in browsers, creating some abysmal headaches for developers.
- HTML only describes the appearance of documents, not the contents, thus making it more difficult to find specific content on the Web.
- HTML does not allow individual elements to be marked up semantically to indicate what each element means (e.g., the difference between one's home address and one's e-mail address).

These limitations of HTML are, in fact, slowing down the Web as the proliferation of Web-based information is becoming ineffectual because of our inability to sift through it all. At the same time, our "speed-of-light" network known as the World Wide Web is slowing to a crawl. It takes longer to find not only the specific site, but also the specific information within the site, such as the price or color of a product, because of the plethora of possible choices.

SGML: Where It All Began

It was not difficult to see the problems that HTML was causing. Thus, in 1996 the World Wide Web Consortium (W3C) went back to the mother tongue to find a solution — the Standard Generalized Markup Language (SGML). Most people are unaware that HTML is a very simple application of SGML. SGML is a universal standard supported by a large number of software vendors that describes the data itself, not just the way it is represented. SGML also provides for a more structured environment; any SGML document can be a container for another document, with arbitrary nesting, allowing complex documents to be constructed from simpler ones.

The only problem with SGML is that it is too general and far too complex for most Web browsers to process, with a specification (set of standards and requirements) of over 500 pages. And the answer was not expanding HTML, which would be limited and need constant adaptation. So a new language, XML, was derived by creating a subset of SGML, a streamlined metalanguage that enables users to build their own markup languages. XML's specification is limited to a much more manageable 50 pages than SGML's original 500. Yet XML consists of enough rules so that anyone can create a markup language from scratch, and is constructed in a way such that HTML fits into the new metalanguage (see Exhibit 93.1).

Benefits of XML

A large number of companies are jumping on the XML bandwagon, and for good reason. XML provides an array of benefits, many of which were not present with HTML, including:

- *Simplicity.* XML is usually easily readable and understandable to both people and computers, is easily processed by computers, and yet is still capable of representing complex data structures. It is much easier to learn than other distributed software technologies (such as CORBA and DCOM) and saves development time.
- *Open standard.* XML is an open, World Wide Web Consortium (W3C) standard, and almost every major software developer in the world endorses XML. Although Microsoft, Oracle, and IBM may never agree on where the sun rises, they all support the open standard for XML in their software products.

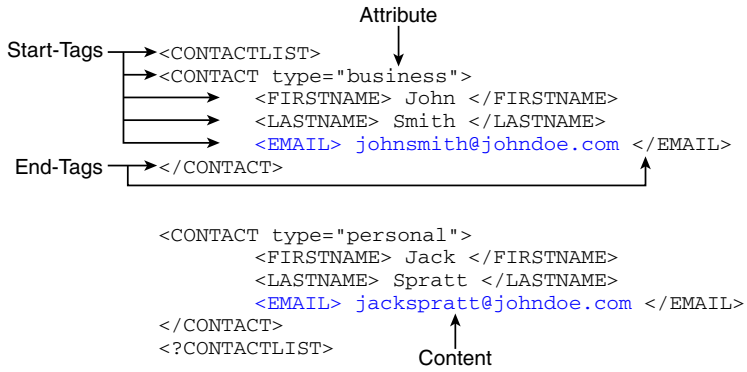


EXHIBIT 93.2 The basic syntax of XML.

- *Data description*. XML makes it easier to provide metadata, or descriptive data about the information. This in turn opens up possibilities in data mining or more efficient search engines, helping the consumer find information or find the information producer.
- *Publishing heaven*. One of the greatest benefits of XML is its ability to separate content from design, and vice versa. Content management has been a problem since the typewriter and has become more important as documents become interwoven with the digital infrastructure. XML provides a key solution allowing the look of the document to change without touching the content, and allowing the content to change without touching the design.

XML Nuts and Bolts

Basically, XML consists of rules and conventions allowing anyone to create a markup language from scratch. As a result, when creating an XML document, one creates one's own elements and assigns them any names one likes. In this way, XML can be used to describe almost any type of document, such as a list of car accessories or a contact list.

As evidenced in Exhibit 93.2, the syntax in XML is so easy that even nonprogrammers can develop tags in a matter of hours. This example also demonstrates the basic rules for creating a well-formed XML document. A well-formed document is one that conforms to the minimal set of rules that allows the document to be processed. The example in Exhibit 93.2 conforms to the following rules for XML:

- *Document element*. Each document must have only one top-level element; the document element or root element in the example is "CONTACTLIST."
- *Element nesting*. If an element starts within another element, it must also end within that same element. In the example, if one of the lines was written as:

  `<EMAIL>johnsmith@johndoe.com</CONTACT></EMAIL>`

  it would not be considered valid because the end tag for </EMAIL> must be placed before the end tag for </CONTACT>.
- *Start and end tags*. Each element must have both a start tag and an end tag, and the element name must exactly match the name in the corresponding end tag. Element names are case sensitive.

This example demonstrates that although XML is very simple, it is also very rigid in many ways. However, this is not a problem, but one of the real unifying powers of XML — everybody has to adhere to the rules for it all to work.

Document Type Definition

Even understandable tags only make sense if they are known to everyone who needs them. Groups of users who want to have a common document type have another valuable tool available to them in XML: the

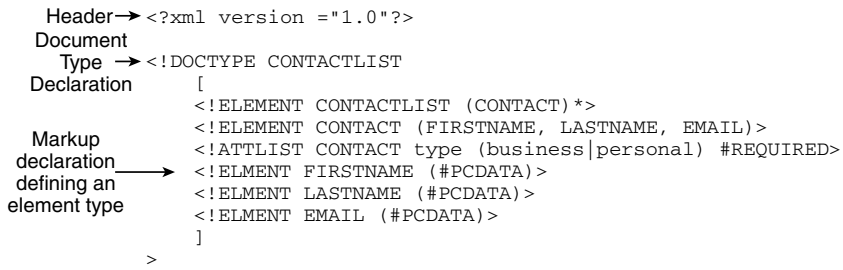


EXHIBIT 93.3 DTD with an XML header.

Document Type Definition (DTD). This aspect of XML facilitates the definition of industry-specific standards for information exchange. Thus, the example in Exhibit 93.2 could be preceded by a DTD, as shown in Exhibit 93.3.

The use of DTDs is also a very powerful validation tool. In the DTD in Exhibit 93.3, using commas between the elements that make up the element CONTACT indicates the "sequence" form for the subsequent (child) elements. So, if one tries to add an element such as:

```
<!--Invalid element -->
<CONTACT>
      <LASTNAME> Doe </LASTNAME>
<FIRSTNAME> Jane </FIRSTNAME>
      <EMAIL> janedoe@johndoe.com </EMAIL>
</CONTACT>
```

it would not be considered valid because the order of the child elements is not as declared in the DTD. Omitting a child element or including the same child element type more than once would also be considered invalid.

Because XML is both simple and capable of defining document types, it has the potential to solve significant programming problems for building interactive business applications. A general-purpose set of XML elements and document structure is known as an XML application, or XML vocabulary. Industry groups such as the finance, health, chemical, and newspaper industries have already made large inroads into creating their own XML applications for their industry members; for example, CML (Chemistry Markup Language) and OFX (Open Financial Exchange).

Other XML Tools

In addition to creating XML applications for a specific industry group, or class of documents, XML applications or standards are constantly being developed that can be used within any type of XML document. These applications can make it easier to produce, format, or secure XML documents. Some examples include:

- *XLink*. The new XML Linking Language allows multiple link targets and is significantly more powerful than the HTML linking mechanism.
- *XSL*. The eXtensible Stylesheet Language enables the creation of powerful document stylesheets using XML syntax.
- *XML Schema*. The formalized concepts for XML Schema were published by the W3C in March 2001. XML Schema is a more powerful alternative to writing DTDs.

Security Issues of XML

As with the Internet, information security was not the first, or even second, area of concern when XML was designed. The word "security" barely made a token appearance in the initial recommendation for XML — as a programming example. Yet, XML promises to make data easier to read, organize, and disseminate — you can almost hear the sales pitch:

Oh, you wanted *security* with your new XML and the <autoaccessory> leather seats</autoaccessory>? Well sir, that is going to cost you extra.

XML as a Disruptive Technology?

One of the key problems with any new technology is its potential for disruptive influence. Information security professionals tend to like mature products and are most comfortable in stable, unchanging environments. XML is by no means mature and new standards are introduced on an almost-monthly basis. XML also brings change not only to the landscape of the Internet, but also to many other business and database applications.

By and large, the greatest change lies with the technologies and protocols based on HTML. These technologies and the related infrastructures have shortcomings, but they were shortcomings that were understood by the system administrator or information security professional. The existing protocols for these infrastructures work fairly well, up to a point. XML goes well beyond that point and thus becomes a serious problem of relearning the rules and of pushing the boundaries of infrastructure that were not designed for the flexible content that XML brings.

Probably the biggest example of the type of impact XML is having is that HTML is no longer being considered for any further work on its own, but rather as a reformulation within XML. In essence, XML has ended the development of HTML as its own domain, and reduced HTML to the status of a vocabulary — albeit an important one.

Verbosity and File Size

XML markup can be incredibly verbose. XML uses a text format and uses tags to delimit the data. Because of this, XML files are almost always larger than comparable binary formats. In the previous examples, the XML tags easily tripled the size of the file. Proponents of XML point out that disk space is not as expensive as it used to be and that there are many ways to compress and transmit data accurately and quickly.

Although this new aspect to the bloat in file size can be compensated for, it should be well planned for and not assumed as some minor performance factor. Some companies will be transferring terabytes and larger complex data structures to XML. Even minimal file size expansions of 40 or 50 percent can have a large, somewhat expensive impact on these large databases. Information technology workers and managers at all levels must factor in the space and bandwidth issues for these larger systems as the transition to XML continues.

That Internet Thing Again

XML is fast becoming a *lingua franca* among business applications using the Internet. XML should provide for easy and seamless purchasing, banking, and other functions as it matures. But the Internet is as insecure as ever, and XML will do nothing to improve it. In fact, XML purposely moves us in the direction of making all the data transmitted over the Internet easier to understand and read.

Almost all the major vendors, along with the W3C, saw this problem waiting to lay waste to all their efforts in adopting XML. The problem essentially boils down to two well-known security problems: confidentiality and authentication. Encryption is needed to keep the more important or private data confidential, a problem that could occur on a very granular level. For example, users pulling information out of a document may have access to information that they do not need to see. Digital signatures are needed to provide authenticity, integrity, and non-repudiation.

At first, major vendors supplied their own security solutions to provide encryption and digital signatures for XML applications. Since then, major vendors and various working groups have been fast-tracking proposals for new encryption and digital signature requirements in XML:

- *Encryption*. In March 2001, the W3C published the requirements specification for XML encryption. According to the specification, the mission of the W3C working group was "to develop a process for encrypting/decrypting digital content (including XML documents and portions thereof) and an XML syntax used to represent the (1) encrypted content and (2) information that enables an intended recipient to decrypt it."
- *Digital signatures*. XML signature requirements (now considered a second recommendation by the W3C) are being addressed concurrently with the XML Key Management Specification (XKMS). The

XKMS requirements were submitted in March 2001 by several major software vendors, including VeriSign, Microsoft, Baltimore Technologies, Citigroup, Hewlett-Packard, IBM, IONA Technologies, PureEdge, and Reuters Limited.

DTDs and New Security Issues

As with the introduction of any new technology, the integration of XML will result in security holes that will be hacked, cracked, and abused. Probably the largest security threat will come from the intentional and unintentional change of XML Schema, DTDs, and even XSL stylesheets. The creation of an XML application, or vocabulary among industry groups, assumes that there will be one XML application upon which all else will be built. It is also logical to assume that companies will use, and in many cases require, "master" DTDs or stylesheets for internal and external usage. A small change could produce a fatal error in a DTD and could halt XML processing on a large scale. And an attack of this nature need not be sophisticated. A cracker could change an attribute from optional to required, and get a big laugh as a company spends hours trying to find this small, "innocuous" error.

What if one, the consummate security professional, relies on a default attribute or DTD for the security of data? A small change could expose enormous quantities of privileged data. What if one relied on XML in various security products for access control? A small error could lock out one's entire company from the network, or provide access to the very people one would like to exclude from network services.

DTDs could also be exploited in other ways. If the header of an XML document contained a URL to establish a path to the DTD elsewhere on the network, the client must have access to the DTD to evaluate XML objects. If the DTD host server is behind the firewall, then once communication is established between the client and server, the firewall could be defeated.

All of these attacks or problems are very simple relative to other ways computer systems are cracked. Although subsequent solutions will undoubtedly be published, and new security included in various XML tool sets, the very open nature of XML ensures that these less-sophisticated attacks will continue to be a problem, especially for the more naïve companies that fail to take adequate steps to protect their data.

The XML Family, Step-Children, and Bastards

XML is definitely a family of technologies, but the continuous development of modules and applications for specific tasks is far from over, creating a large number of uncertainties. Some of the new specifications for XML encryption, or XSL, or Xlink, are now in place, but the community of vested interests, from major software vendors to financial institutions, still has a lot of debating to do. Other specifications and recommendations are just now surfacing, and many more will be developed over the next few years. Of course, there is the long line of software vendors all ready to support XML. And as certain as taxes, there is also the long line of software upgrades to support the new additions to XML as each new module or application becomes "official."

As new software for XML is developed, and as XML is added to existing products, security holes will develop because of the push to get "enhanced" applications to market as quickly as possible. For example, consider the security problems that have developed with a browser application and a database application after the integration of XML. This trend is likely to continue in the near future.

With all these new requirements, modules, and applications going around for XML, the entire field is becoming confusing, adding just a little more risk to the entire endeavor. Again, this has not gone unnoticed by the W3C or various industry groups. RosettaNet, an industry consortium of over 400 members, has made a recent plea for XML convergence among the various applications. But 400 members do not make a world, and the world is assured a slightly tortuous route to this convergence as all the vested interests weigh in.

Some Conclusions

While there is currently a lot of work under way on various standards, requirements, and modules for XML, this work is maturing at a rapid pace. Despite the ongoing development, make no mistake — XML is already here. It is proliferating throughout information technology on corporate, industry-specific, and global scales. And XML is making large impacts on electronic publishing, database storage, the exchange of electronic

documents, and application integration. It is therefore important that executives at all levels, including those involved in information security, understand the nature of the Holy Grail known as eXtensible Markup Language.

One of the odd aspects of the proliferation of XML is that to enjoy the benefits of drinking from this Holy Grail requires that everyone, not just one person, drink from the Holy Grail. By and large, XML requires XML-based input by users in order to thrive and for everyone to see the promise of XML on the Web and in E-commerce. As XML becomes widely adopted, everyone should benefit from faster publishing of information, faster processing of orders, and faster document searches. Of course, a huge factor in this success will hinge on whether XML integration and use can be done securely.

XML as a Security Solution

In addition to all the security issues that must be addressed for XML, the astute security professional, programmer, or executive may start to realize a trend not previously considered: XML is being used as part of security solutions. Security is no different than healthcare or automobiles; it has its own distinct vocabulary and ways of organizing data. XML will be used not only to provide a common document framework for information security, but also to integrate the various security tasks among applications and computer systems.

One is already starting to see this trend in various aspects of security-related programs, such as Microsoft Exchange. As this trend continues, it will become more important for security professionals to understand the fundamentals of XML and how XML is used in various security solutions because XML may very well become a binding agent among various security components.

Where to Go from Here

The XML world is a demanding one, and this chapter presents just a broad summary regarding XML and XML security issues. To exploit XML to its fullest and to secure applications and data dependent on it, programmers, executives, and security professionals must be versed in a wide range of topics. Stylesheets, DTDs, data trees, and hyperlinked structures will all become common to a more robust and more usable infrastructure of the digital world. The defense lies not only with maintaining good security policies, but, as always, staying current with technology.

For more information, there are a variety of Web sites that provide up-to-the-minute information and news on XML. A good place to start is the Web site for the World Wide Web Consortium: www.w3.org. One can also look in any major search engine for "XML" and quickly become inundated by the amount of information one will find. One can only hope that XML will transform that one process of searching for more information faster and much more accurately as time goes on.

TESTING OBJECT-BASED APPLICATIONS

Polly Perryman Kuver

INSIDE

Object Properties; Object Methods; Classes; Object Events; The Testing Effort

INTRODUCTION

Buttons, icons, fields, menus, and windows are all objects. Each of them, by the very nature of being objects, possess properties, methods, and events. Properties describe the object. Methods state what the object can be told to do. Events are what the object does when it is invoked. For example, the print icon in Microsoft Word is usually one-quarter inch by one-quarter inch (property). It is gray and has a picture of a yellow printer with a piece of paper on it (property). It can be told to appear on the toolbar (method), be grayed out when it is not available (method), and recognize clicks from the left mouse button (method). When clicked, it will invoke print code, causing the document to print (event).

Because the print icon is a defined object, it can be used again in other applications. While nearly all software manufacturers today take advantage of code reuse, it is exemplified in the Microsoft products where the same print icon appears across all MS products from Office to Explorer and Exchange. Reusability is one benefit of object-oriented development. Maintenance is another, and the value of object-oriented development will continue to grow with technology because not many companies can stay competitive if they cannot build once, test thoroughly, and then use again and again and again.

As object-oriented development spreads and grows in the software community, techniques for testing object-based applications become more important. An understanding of

PAYOFF IDEA

The important thing to remember in testing object-based applications is that incremental development and user involvement make the process move along swiftly and more smoothly. When an object is created, it can be viewed by the user in a prototype. Changes can be made easily as the application moves from prototype to finished production system. When testing is managed and automated, it can be repeated and elaborated upon without starting from scratch because scripts are reusable and maintainable.

objects and object classes is the first step in understanding current testing techniques and developing the skill to invent proper testing techniques.

PROPERTIES

Object properties ensure that the use of one type of object is consistent throughout an application. Take buttons, for example. Whatever the application, it is easier to use and more appealing to the eye when all of the buttons available to the user are the same shape, color, and size. To accomplish this, button properties include the dimensions for width and height, color, and font properties. Each property is defined in one place for objects of a single type. When the developer wants to use this button in another software package with a green button instead of a gray button, the object properties for the button can be accessed and modified in one location, one time for the entire application. The gray button becomes a green button throughout the application with this one change. If the color of the button is to be selected by the user, the color property is made public, since any public property can be accessed by the user. In this example, the color property is made accessible to the user, allowing it to be changed by the user from an available color palette. When the object has been thoroughly tested in the first application, this type of change does not warrant or require retesting of the object at the object level.

Testing becomes a matter of checking to ensure that the correct version of the button object has been included in the new application. This testing includes checks to ensure that one of the developers did not define a button object somewhere within their code that was not affected by the single instance change. The tester will perform this as a black-box test. That is, from an end-user perspective: Does the button display when it is supposed to display? Is the button active when it is suppose to be active? This is especially needed when the button object or any other type of object is public. If a user opts to change a color, it must change everywhere or the help desk will be receiving a lot of unnecessary calls.

Modifying the text of an object is just as simple as changing dimensions, color, and fonts. One button object is created and defined. Since text is a unique property worded to be consistent with the action the specific button will perform, the property text can be changed to fit the function. When the object text property is changed, the object should be saved under a new name to which new methods and events will be assigned. For example, standard words for the buttons may advisedly be used throughout the application. "OK" is used instead of "Enter." In fact, "OK" has become somewhat of a *de facto* standard in the windows world, replacing what was at one time a specific command or series of commands to update records.

In some applications, "OK" does not mean update; rather, it is used to indicate continue, show me the next screen. In those cases, updating may

not occur until a button saying "Update" is located and clicked. For this reason, it is important to define the application text property standard and name the object appropriately. The text on a button is not generally a property that users are allowed to change. This property is hidden.

When it comes to testing an application, it is easier to identify defects in consistency and usage when object properties are defined in system specifications. This is because the specification implicitly explains what is supposed to be happening. However, the very nature of the object-oriented design often preempts the creation and publication of formal specifications. The standards used in defining object properties are in somebody's head or on little yellow Post-Its™ stuck on and around the developer's monitor. When the application moves into testing, the Post-Its do not move to testing with the software. That may be alright if the testing is to be limited to purely black-box function testing, but what about "look and feel"? In today's market, "look-and-feel" testing is critical. Consumers place a lot of emphasis on it. It cannot be ignored. So, how is it done?

It is done by creating business-based scenarios on which end-to-end testing is planned and documented in the test plan. This accomplishes three things: it documents the scenarios as well as the strategy and scheduling for testing the object properties and that all objects were addressed during testing; it documents how each object was addressed; and it documents the criteria used to determine if the objects throughout the application met a specific level of quality.

Addressing object properties in the test plan does not have to be involved. Simple bullets or sentences can be used. Toolbar objects will all be:

- gray in color
- have Times New Roman print
- in bold print

METHODS

The development of object properties and methods go hand-in-hand and are often discovered simultaneously during the design phase of the project. Properties characterize an object. Methods animate the object by defining what it can do. Think about it terms of action words. Methods equal actions the object can be told to do, such as display, show, move, get, calculate.

Methods can be defined and hidden from the user, or they can be public, allowing users to select the method from a list of options. An example of this is the selection of icons to be displayed on a toolbar or on a pull-down menu. In any Microsoft product, the user can go to View/Toolbars within the application and click each of the desired icons on or off; those with a check will appear on the toolbar. In reality, the

user is changing the method used by the object in the icon class from hidden to display.

Together, the properties and methods determine the boundaries or interface of an object and are defined as an instance of a class. Test scenarios that check methods, as well as how the class structure interprets the methods, must be planned for all object-based applications.

CLASSES

The combined properties and methods must be identified and recognizable to the class structure being used. That is, there must be an icon class or something equivalent to an icon class before an instance of an icon, say a print icon, can be used in an application.

The class structure is based on object linking embedded (OLE) technology and supported by the programming language used. Visual Basic, java, and C++ each have their own techniques for handling class structures. That is, they recognize specific types of object property–method combinations and, while they may each have an icon class, a button class, and a menu class, the rules governing the inclusion of an object within the class may vary.

The value of object-oriented design and development is in the adjustments that can be made at the object level, allowing developers to make the necessary changes without touching every screen and form in an application. Variances between class structures reduce portability and increase the maintainability of objects across platforms.

This is exemplified in the case of Microsoft Java versus SUN Java where standard Java objects had to be redefined to meet the unique requirements of Microsoft's Windows-optimized Java. Internet applications, test tools, and other products constructed in standard SUN Java to take advantage of the virtual machine capabilities it offers had to be redefined, reconstructed, and retested to run on Microsoft Windows platforms. This significantly increased the workload for many software manufacturers already stretching to meet customer requests in a highly competitive market.

When a class structure for a specific family of objects does not exist, Visual Basic and other programming languages allow for the coding of instructions for recognition.

While testing for class acceptance and recognition is an important part of testing object-based applications, it is perhaps even more important to have a predefined approach for reporting and debugging class-related defects during the testing process.

EVENTS

Once the right icons and buttons are defined, tested, and placed in an object container such as a spreadsheet, document frame, or form, the

code behind the scenes is connected to the objects, allowing intended application functions to occur. Test scenarios that will demonstrate "if this action, then that result" need to be developed and executed. If the user clicks on the print icon, then something will print; if the user clicks "OK," then the next form is displayed.

Testing is based on the intended object event, and object-based testing tools provide the power to exercise the event thoroughly. The reason for this is that the test criteria can be established and the steps of the test recorded in a single script. The script can then be copied and easily modified at various points to extend test coverage to any number of "what–if" conditions, based on the intended event of the object.

In traditional code or non-object-based applications, the events are actually the equivalent of program control points. Each control point triggers a subroutine, a macro, or another program. To test, each of the possible paths must be first identified and then tested. The number of "what–if" conditions is limited by the number of conditions the tester can perform in the allocated testing time.

Since testing of object-based applications can be more extensive using the testing tools available, more extensive testing can be performed in the same allocated testing time. As a result, more defects can be found, fixed, and retested. So while there may be little or no difference between object-based and traditional applications in the types of defects found, it can be faster and more effective to find the defects in an object-based system, and it is certainly more judicious to fix and retest the object-based application.

Use of a tool is not mandatory in testing object-based applications. Manual testing of object events can be conducted in the same way traditional program control points are exercised. Manual testing involves defining scenarios for all the possible paths of a program or possible paths that can occur, given various conditions for an event and exercising those paths using basic business-use scenarios. For example, if a program stores data in a database, the data can be entered from an updated payroll screen or the human resources screen, as might occur in an integrated system if an employee marries and changes the number of dependents on a W4 form and insurance coverage.

Manual testing would require separate tests for each of the data-entry screens in payroll and human resources to be defined and executed; whereas, an object-based automated testing tool could be scripted to recognize the variables of the different data-entry forms and test the object-event, which updates the database. Thus, a single script will permit multiple tests to be executed in less time.

THE TESTING EFFORT

A spiral approach to design, development, and testing is a good way to optimize the benefits of object-oriented design and development. It allows

for the quick turnaround required in what one executive at Sun Microsystems, Inc., termed, "Internet time." That is, keeping pace with the rapid changes in technology and meeting customer demand for products that can be easily installed, operated, and customized to fit their environments.

The spiral approach is based on a model originally developed by Barry Boehm for the U.S. Department of Defense. The model promotes and allows for the reconciliation of concurrent, related development efforts that are undertaken in the same timeframe. Thus, individual "production lines" for various objects, object-containers, and background code can be established and run at the same time. The objects, containers, and code converge during the integration phase.

When the spiral model is employed, traditional testing processes must be reviewed and revised to ensure that adequate testing occurs, but that testing does not become a bottleneck in the overall effort to complete development and get the software into production. The very first step for ensuring a successful testing effort is to invest in a software testing tool that provides object-based testing capability. The tool is essential unless an organization can really rationalize a tester using a little ruler to measure objects as they are displayed on the monitor or want to trust visual perception, judgment, and approximation as the basis for pass/fail.

Without a software testing tool, the organization would also have to be prepared to increase the testing budget by orders of magnitude because each time a change was made to an object, testing would have to begin all over again. Use of an object-based testing tool allows for the test script to be modified for reuse. The impact of development changes in the test environment is greatly minimized. The frontrunners in object-based test tools in today's market are: Rationale's Robot, Mercury's WinRunner, and Seque's Silk.

Each of these products can be purchased alone or in a suite of tools. The benefit of purchasing a suite of tools is that they contain applications that significantly help with the organization and management of the testing effort, which is the second consideration of the testing process. Rationale's SQA Manager is an excellent example of a group of tools that support the testing process.

SQA Manager allows test scripts sequences to be defined with dependencies and it keeps track of when, who, and which scripts are run. This ensures that tests can be run to verify object properties, then methods, then events as soon as the object is developed. The same scripts or a subset of them can be reused and be scheduled to rerun when the object is placed in the object-container and again when the application is integrated.

Having the tools selected up front in the testing process ensures that the capabilities they provide can be incorporated into the test plan, thereby maximizing the power of the tool, the reuse of scripts, and the level of quality built into the product.

The test plan, although listed in third place in the testing process, is essential in building a solid testing effort. It takes the testing from beginning to end in a logical, thorough process. A good plan will allow for testing to be performed in increments and keep pace with development.

THE PLAN

The use of a testing tool does not eliminate the need to plan. Rather, it ensures that a good plan can be implemented with better, more consistent results and repeated as modules are added, modified, or deleted. For example, using the automated test tool Rationale Robot to test at the object properties and methods level would be carried out by running Object Properties and Alphanumeric test scripts. The Object Properties test will capture and compare objects.

A Robot Alphanumeric script checks for case-sensitive or case-insensitive test, numbers, or a number within a range. It will also check to see if a field is blank and allow testers to tailor the test to specific values. Again, the description should specify how the test was set up and what values were used for verification.

Validating the objects in the containers might include Window Existence scripts that literally verify that the correct window exists in memory. For example, does a pushbutton (object) appear on the dialue box (container) as expected? These scripts can be followed by event tests that ensure that each object in the container performs as expected. The event scripts may include customized .DLL or EXE routines constructed by the development team. List scripts to determine if the alphanumeric contents of list boxes, combo boxes, and multi-line edit controls work properly. Event scripts can also be created to verify file existence, menu selections up to five submenus deep, and file comparisons.

The integration test or system test validates the functions of the application to see if they meet the end-user business needs. These scripts capture the keystrokes of the end user and can include the common wait state scripts that ensure that data populates a screen within a specified period of time or that an object is accessible when it is supposed to be during day-to-day operations. Scripts can also be set up to ensure that the edits are being performed correctly, that data has been entered in all required fields, and that pop-up windows and dialog boxes appear when that are supposed to with the correct information.

For example, one test for a purchase order application might be to ensure that the correct forms are accessed. When the type of purchase is designated as Fashion items, the series of frames, forms, or windows accessed will be different than when the type of purchase is for Staple items. The test is set up to enter all required data, including the type of purchase to be made, then click on the “OK” button. A wait state is established for the “OK” button by indicating that it is grayed out after it is

clicked, making it inactive and unusable until the next form is displayed. That is, the test tool will automatically check to see if the next form is displayed as a result of clicking "OK." The tester specifies how often the checks are made (e.g., every two seconds for up to 30 seconds). If the correct form is not found in the 30-second period, the test fails. If the correct form is identified in that time period, the test passes. The tool determines if the form is the correct form, based on tester-defined criteria for the forms; for example, in a linked test, the banner information of the correct form, Fashion or Staple, would be specified and verified by the tool.

When the type of script is selected, it is documented in the description, along with the values and other criteria used. This documentation can be created as comments within the script rather than as a separate word processing document.

What all of this means is that by the time tests are executed to verify that data is being saved correctly, and the right window pops-up when it is supposed to, it has already been proven that the windows all have a banner or header and that the label in every banner and header will present itself with the same color.

In other words, like tests, are done with like tests and those things that in days gone by were considered merely cosmetic are identified, cleaned up, and laid to rest before an application ever gets to system test. When the same objects are used to create each of the windows, it is only necessary to test that the windows were created using the approved objects. Objects need only to be tested when a revision is made to an existing object or a new object is created.

SUMMARY

The important thing to remember in testing object-based applications is that incremental development and user involvement make the process move along swiftly and more smoothly. When an object is created, it can be viewed by the user in a prototype. Changes can be easily made as the application moves from prototype to finished production system. When testing is managed and automated, it can be repeated and elaborated upon without starting from scratch because scripts are reusable and maintainable.

Testing the functionality of an application — whether it is object-based or traditional — requires the construction of business-use scenarios mapped to system requirements. The difference in testing the two types of application is in the approach used and type of automated testing tools available. To get started:

- Define the scope of the test.
- Get an understanding of what is supposed to happen when an object event or program control is triggered.

- Create single-event scenarios (based on the object event or the program control points).
- Cover as many "if-else" conditions as time allows.
- Build scenarios that exercise as many conditions as possible.
- If a testing tools is going to be used, determine what scripts need to be created and how they can be reused by defining variable or modifying specific lines in the script.

Notes

1. Microsoft, *Visual Basic 6.0, Programmer's Guide,* Microsoft Press, 1998.
2. Rationale, *SQA Suite Documentation,* Rationale University, 1996–1997.
3. Kaner, C., Falk, J., and Nguyen Hung Quoc, *Testing Computer Software, 2nd ed.,* International Thomson Computer Press, 1998.

Polly Perryman Kuver has more than 19 years of computer experience, including 12 years in management positions. As a process engineer, her areas of expertise are national and international software engineering and documentation standards, quality assurance, configuration management, and data management. Currently, she is a consultant in the Boston, MA, area.

Secure and Managed Object-Oriented Programming

Louis B. Fried

Payoff

Object-oriented programming has great promise for reducing maintenance and speeding development. It does, however, have its drawbacks concerning the management and security of object inventories. This article explains how to control and secure an object-oriented programming inventory so that the full benefits of the technology can be realized.

Problems Addressed

Software development has always been expensive. Those who pay the bill dream of obtaining results for lower cost and in less time. The search for tools to realize this dream has produced data base management systems, query systems, screen development tools, fourth-generation languages, graphic programming aids, and code generator. The ultimate tools, however, will free developers from programming altogether, and the best way to do this is to reuse existing code.

The various tools that developers already use are effective because they reuse code in some sense. For example, using data base management systems, programmers need not develop their own access routines as they were forced to do many years ago.

The developers of Object-Oriented Programming languages and tools promise to take the reuse of code to new levels, but there are ongoing debates about the benefits and the potential problems associated with object-oriented programming (OOP). For each argument there are various responses.

The Overriding Benefit: Reusable Software

One concern is that objects require continuous maintenance and enhancement to keep up with the changing needs of the business. However, software has always required maintenance.

Another concern is that the analysis task required to identify and define appropriate objects is formidable. Advocates of Object-Oriented Programming respond that the best software development efforts result from spending more time in the definition and specification phases; in addition, developers can reuse objects for long-term savings.

In fact, proponents of object-oriented programming (OOP) point out that the need to define classes and subclasses of objects, the objects themselves, and the attributes, messages, methods, and interrelationships of objects forces a better model of the system to be developed. Many objects developed in object-oriented programming (OOP) code will not be reused; however, the real benefit is that object-oriented programming (OOP) code is usually more lucid and well organized than traditional coding methods. The process that forces analysts to define the object hierarchies makes the analysts more familiar with the business in which the application will be used.

When these problems and objections are analyzed, many of them can be discounted; however, some remain. Viewed in isolation, object-oriented programming (OOP) is simply an attractive way to facilitate structured, self-documenting, highly maintainable, and reusable code. In the context of enterprisewide application building, Object-Oriented Programming does present unique challenges whose solutions require additional tools and management methods.

The Object-Oriented Programming Environment

As object-oriented techniques gradually find a place in corporate programming departments, there will be attempts to expand the use of this technology from single applications to broad suites of applications and from the sharing of objects among a limited group of applications developers to use by developers and users throughout the organization. To accomplish this expansion of use, Object-Oriented Programming will need to be used within a development framework that is composed of CASE tools implemented in a distributed, cooperative processing environment.

A likely scenario of the way in which organizations will want to use object-oriented programming (OOP) in the future is as follows:

- Objects will be used by decentralized development groups to create applications that are logically related to one another and for which common definitions (i.e., standards) are imposed by various levels of the organization.

- Users will employ objects to develop limited extensions of basic applications or to build local applications, in much the same way spreadsheets and query systems are currently used. Users may access corporate data bases in this environment through objects that encapsulate permitted user view of information.

- Object-oriented programming will become integrated with CASE platforms not only through the inclusion of object-capable languages, but through repositories of objects that contain both the objects themselves and the definitions of the objects and their permitted use. Improved CASE tools that can manage and control versions and releases of objects as well as programs will be needed.

This scenario envisions optimum use and benefit from object-oriented programming (OOP) through extensive reuse of proven code within a framework that allows authorized access to objects.

The current status of object-oriented programming (OOP) is far from this scenario. The effective use of object-oriented programming (OOP) depends on the ability to solve problems related to two major areas of concern: the management of the object inventory and the preservation of information security in an object-oriented development environment.

Managing the Object Inventory

Objects in the inventory must reside in a repository that uses an object-oriented data base management system. Objects are identified by classes and subclasses. (Object class definitions are themselves objects.) This identification provides a means of inventory management. For example, retrieving an object within a class called Accounts Payable would help to narrow the domain being searched for the object. A further narrowing can be done by finding a subclass called Vendor's Invoice, and so forth. Polymorphism allows the same object name to be used in different contexts, so the object Unit Price could be used within the context of the Vendor's Invoice subclass and the Purchase Order subclass. Some Relational Data Base Management System also allow polymorphism.

Several problems arise as a result of this organizational method. To take advantage of the reusability of objects, the user must be able to find the object with as little effort as possible. Within the classification scheme for a relatively straightforward application, this does not appear to present a substantial problem.

Most organizations undertake the development of applications on an incremental basis. That is, they do not attempt to develop all applications at once. Furthermore, retroactively analyzing and describing the data and process flows of the entire organization has failed

repeatedly. By the time all the analysis is completed, the uses have lost patience with the IS department.

It is feasible to limit objects to an application domain. However, limiting objects to use within the narrow domain of a single application may substantially reduce the opportunities for reuse. This means that developers will have to predict, to the extent possible, the potential use of an object to ensure its maximum utility.

Cross-Application Issues

It is possible to establish a class of objects that may be called cross-application objects. Such objects would be the same regardless of the context within which they were designed to be used. For example, the treatment of data related to a specific account in the corporate chart of accounts may always be the same. The word *account* appears in many contexts and uses throughout a business. Therefore, another approach to this problem is that some objects may be assigned an attribute of cross-application usability.

As more object-oriented applications are created, the typical data dictionary or respository will not be able to serve the needs of users for retrieving objects. Analysts and programmers who are required to move from one application to another to perform their work may find the proliferation of objects to be overwhelming. The IS department will need to develop taxonomies of names and definitions to permit effective retrieval.

Developing and maintaining a taxonomy is in itself a massive effort. For example, a large nuclear engineering company realized that the nuclear power plants it had designed would be decommissioned and dismantled in 50 years. The personnel responsible for dismantling a plant needed to know all about the plant's 50 years of maintenance in order to avoid potential contamination of the environment and injury to themselves.

The company discovered that various names were used for identical parts, materials, and processes (all of which are objects) in the average plant. Furthermore, because the plants were built throughout the world, these objects had names in many different languages. If personnel could not name an object, they could not find the engineering drawings or documents that described the object. If they searched for only the most likely names, they would overlook information that was stored under an unusual name.

A taxonomy project was initiated to adopt and use standard terminology for all components of the plant and all information relating to those components. Within two years, a massive volume was assembled. Still, several problems surfaced. It was impossible to know when the taxonomy would be complete. New terms had to be created to avoid duplication. The taxonomy manual was so large that engineers and other employees refused to use it.

This example can provide some obvious guidelines. A comprehensive, detailed data model will never be completed, because the organization constantly changes while the model is being created. Instead, a high-level process and information model of the organization should be designed to indicate potential or existing relationships between data. This model will also be used to identify data and objects that can be reused in future applications development projects. Limited domains or business processes should be chosen for the creation of objects within an application. Also, object-naming conventions and an object-inventory system should be established before any object-oriented application is developed. Most important, defining objects, as well as developing applications, is an incremental process and objects will not be reused if they cannot be easily found.

One dimension of the problem of naming and defining objects has been examined. In a world of increasingly distributed processing and decentralized use of computing, IS must also consider that:

- Analysts and programmers will not be under centralized control in all instances.

- Other personnel, such as engineers, clerical staff, and knowledge workers, will use objects to create their own programs.

Retrieval Methods to Facilitate Reuse

The ability of users to develop their own programs and applications is one of the greatest benefits that can be obtained from Object-Oriented Programming and shouldn't be ignored. Nor can the demands of an increasingly computer-literate clientele be refused. This means that the methods for retrieving objects must be available to all users for a relatively small amount of effort. If not, objects will not be reused.

With users as a recognized component of the management problem, another concern emerges. Objects must not only cross application domains, they must exist at various levels of the organization. For example, an object may be defined as applicable throughout the organization in a given context (i.e., a Standard object). Such an object may be called a Corporate object either through being in a class of corporate objects or by having a standard attribute as a corporate object. Another object may be applicable only within a specific strategic business unit and may be called, for example, an Engine Manufacturing Company object. At the next level, an object may be called a Casting Division object. Objects can be described in this manner down to the level of the desktop or the computer-controlled machine tool.

Two types of tools may come to partial rescue in resolving this problem. Text search and retrieval systems may provide the ability to allow users to search for objects within various contexts. The result, however, could be the retrieval of many possible objects from a repository, compelling the user to evaluate them before a selection is possible.

An approach is needed that allows the user to obtain a limited number of possible objects to solve a problem and yet does not force the organization to develop a taxonomy or limit the use of terms. Self-indexing files for nonhierarchical search may prove helpful, but this may mean using the object-oriented DBMS repository in a manner not compatible with its structure.

Regardless of the method used, there is a clear need to establish and conform to documentation standards for objects so that searches for objects will return meaningful results. One possible solution is to use an expert system in conjunction with a text search and retrieval system. Expert systems can accomplish classification and are capable of supporting natural language interfaces. Ideally, the user could describe to the system the nature of the object needed and the system could find the most appropriate object. The user could then describe the application at a high level and the system would find and assemble all appropriate objects that fit the system context.

Object Maintenance

When objects are used throughout a large organization it must be assumed that they will reside in repositories on a variety of machines in many locations. Each of these repositories must be maintained in synchronization with the master repository of approved objects for the organization and its divisions. Distributed environments imply additional problems that must be solved before object-oriented techniques can work successfully.

For example, if objects are automatically replaced with new versions, there must be a mechanism for scheduling the recompilation or relinking of programs that use the affected objects. If objects are used in an interpretive mode (rather than being compiled into machine code), replacements will automatically affect their use in existing procedures, perhaps to the detriment of the application. Some methods currently used to maintain distributed data base concurrency and to control the distribution of microcomputer programs throughout a network may be adapted to solve part of this problem. Another approach may adapt the

messaging capabilities of objects to send notification of a potential change to any subobject within the hierarchy of the object being replaced.

Another problem is that identical objects may need to be developed in different languages to meet the needs of users of different hardware systems. Even if objects are developed in the same language, the options are to use either a restricted subset of the language compatible with all potential environments or a language that allows compiler flags to be placed on code and alternative versions of the code embedded in the object. Neither of these choices is attractive, and the first may require other classes of objects to differentiate between identical objects used on different machines (though polymorphism can help in this respect). As a result, the testing process for new or replacement objects becomes more complex.

Organizations will also need to assign someone the job of deciding which objects should be distributed to which of the distributed repositories. Standard corporate objects may have wide distribution, whereas others may require more circumscribed distribution. Object and object-class management becomes a major administrative task.

Object Security

For users, analysts, and programmers to use objects in developing programs or applications, they need access to these objects. Such indiscriminate access provides a real threat to the security of objects.

Information security has been defined as consisting of three primary properties: availability, confidentiality, and integrity. As applied to the object inventory, these may be defined as:

- Making objects available to those who need to use them, when they need to use them.
- Ensuring the integrity of objects by preventing unauthorized changes.
- Ensuring the confidentiality of objects by preventing unauthorized access.

Current repositories and directories generally assume that all personnel authorized to access the directory are authorized to access any item in the directory. This line of thinking does not do for an object inventory.

Access Control

An object inventory requires an extended set of security controls to make its use safe for the organization. Such controls, required to preserve integrity, must be implemented at the object attribute level. For example, in a payroll file the individual salary rate (an attribute) may be restricted to certain users. The attribute must therefore have an attached attribute (sometimes called a facet) that specifies which programs are allowed to read the attribute Salary Rate. Alternatively, the salary rate attribute could have a facet that is a function that returns an empty field or no data to nonauthorized callers. In essence, each object defined in the inventory may need to be individually controlled as well as controlled within a set or class of objects.

A solution is to ensure that each object in the inventory can be separately locked to prevent change. When an object is accepted into inventory, the lock is activated. A system that truly intends to protect the integrity of the objects would not permit any change to a locked object. If an object needed to be changed, it would have to be deleted and replaced by an approved, tested replacement. Furthermore, a limited group of authorized inventory managers would be the only personnel able to delete an object. Finally, a safeguard system

would automatically file all deleted objects in a locked, back-up repository file so that they may be retrieved in the event of incorrect removal.

Locking logic itself is a problem. In current data base management systems, the problem referred to as a deadly embrace—that is, two parties concurrently attempting to update a record by different logical paths—has been solved. When the locking mechanism must deal with atomic objects rather than transactions or records, the solution may be more difficult.

Ownership

In current security practice, the levels of security assigned to information are designated by the application owner. Each application owner has the duty to specify who may access application information and under what conditions. When objects are in common use, new ways of designating ownership become necessary and certain questions must be addressed: Who owns an object that is used across many applications? Who owns a corporate object?

When the ownership decision is made, the next issue is how to assign access permission. Some access permissions may be assigned by sets or classes of workers. (In the new alliance model of business operations, it is not only employees who work with an organization's systems, but also its suppliers and customers.)Permissions may be granted by levels in the management hierarchy, by sets of people in specific functional areas, by organization unit, and by individual. Permissions need to include (as they do today) the authorization to perform certain functions with an object. Functions for which authorization may need to be defined include read only, delete, add, copy, use, and lock.

Integrity

Integrity may also be addressed by attaching rule-based logic to classes, subclasses, and objects to describe the conditions under which they may be used. The marriage of artificial intelligence techniques and object data base structure may be necessary to prevent misuse of objects.

Availability of objects partially depends on systems availability and network availability, for example. Another concern is that the object is appropriately distributed throughout the organization's processing resources so that it can be conveniently accessed by authorized personnel regardless of the time or location. In large organizations, objects may be distributed in repositories on a variety of machines in various locations, so the potential for erroneous use is multiplied.

Confidentiality

Confidentiality may require that two levels of information access are designated for objects. One level of access may be to permit a user to determine whether a desired object or reasonable facsimile exists in the inventory. This level may only permit authorized personnel to learn of the existence of objects and to obtain a brief description. A second level of access control may be needed to permit users to actually read the object content itself.

Confidentiality can be breached in another way. The aggregation of intelligence through repeated access to selected data bases of information is a threat to current systems. When the atomic level of applications is downsized to objects, a significant change occurs. The aggregation of objects into new relationships may permit combinations of information that would not usually be available to users, thereby enabling unauthorized users to assemble intelligence to which they are not entitled.

The property of inheritance—in which an object subclass contains information about the methods and structure of the superclass it is related to—presents special concerns. A

classification mechanism may be needed that defines permitted relationships among objects and establishes authorization for object relationships, perhaps as a facet or attribute. Alternatively, it is possible to maintain independence between data and code that permits access controls to be placed on the data at the user view or field levels within a data base.

Recommended Course of Action

Many potential problems faced by Object-Oriented Programming are similar to those that have plagued other systems development tools. However, to satisfy customer demands, these problems have been addressed by development tool vendors.

The potential benefits of object-oriented programming (OOP) appear to be substantial. However, until this technology enables users and managers to manage and protect their information assets, object-oriented programming (OOP) should be used under strictly controlled circumstances. As such, the following guidelines are recommended:

- The current lack of methods to manage inventories of objects poses a potential problem to effective widespread reuse. The inventory management capabilities of proposed Object-Oriented Programming development systems should be examined and only those tools with which management methods will work should be used.

- Without solving the problems related to object security, it may not be possible to protect information that is widely used throughout the organization.

- Corporations are real-world entities that change according to changes in business needs and strategies. A comprehensive, detailed data model will never be completed because the organization will always undergo changes. Building an enterprise data model should not be attempted. Instead, a high-level process and information model should be designed to indicate potential or existing relationships between data. Then, a limited domain or business process should be chosen for object-oriented development.

- Object-naming conventions and an object-inventory system should be established before any object-oriented application is developed. For subsequent development projects, the high-level process model should be used to identify potentially reusable data and objects.

- Vendors should be urged to develop appropriate inventory management and security control tools. As soon as such tools are available and proven, they should be acquired.

Author Biographies

Louis B. Fried

Louis B. Fried is vice-president of information technology for SRI International, Menlo Park CA.

APPLICATION SERVICE PROVIDERS

Andres Llana, Jr.

INSIDE

The ISP as an ASP; Moving into E-commerce; Budgetary Considerations; What Should be Outsourced; Integration with Existing Enterprise Systems; Application Hosting; Security is Still a Concern; How Good is Your ASP's Application? A Word on SLAs

THE ISP AS AN ASP

During the late 1960s, computer time-sharing utilities emerged that allowed remote users to dial up multi-million dollar computer centers to run their own Fortran programs. The cost for such computer service was inexpensive because the public network provided low-cost access for large numbers of remote users. Later, in the mid-1970s, these same computer utilities (GE and Boeing) provided online applications that were used to support transaction processing for field sales and maintenance workers, order entry, and other related business applications. Although these computer utilities were not known as such, these were the earliest application service providers for the 1970s and 1980s.

Today, with the growing use of the Internet, similar services are now being offered to the business community. They services are varied, supporting a number of vertical industry applications. These include just about any application software system that has been sold or licensed for operation on an in-house computer system or server.

The role of an ISP has changed. At one time, dial-up 28.8 Kbps or dedicated 56 Kbps was the key to the world. When DSL access arrived, the cost for Internet access and services changed forever. To compete, ISPs

PAYOFF IDEA

Some MIS managers may be concerned about letting a mission-critical application leave the premises. Now one can gain some valuable experience with a new application implemented on the Internet: arrange to rent access to the desired application on a per-use basis. Try to locate several vendors offering the same application. Next, select a test group to use the application for a three-month period. Keep detailed records on costs, user difficulty, customer service, salability, any other features that are important to the mission. At the end of the three-month period, analyze the results. One may find that the cost for running the application on an outsourced basis is far less costly than supporting it in-house. This may be especially true for those applications that support a small user population.

soon resorted to free Internet PCs, free e-mail, free Web hosting, and other incentives that changed the ISP model. To stay competitive and profitable, it became obvious to the survivors that they had to provide more value. Software applications embedded on the Internet provided an ideal solution for the ISP because the network for distribution was already in place. All that was required were the applications needed to support a specific business function.

While this was a new role for ISPs, it was one that they could easily embrace because they were positioned to install and run any time-shared application. This was a different process than Web hosting, because the process of supporting an online application requires a different pool of technical expertise.

These ISPs, turned application service providers (ASPs) made a lot of sense for a small but growing business because they could avoid the costs associated with establishing an expensive talent pool required to set up and run a wide area service network. The business in question could instead concentrate its core competencies in running the business enterprise.

MOVING INTO E-COMMERCE

Some firms have looked upon an ASP alternative as a way to enter into E-commerce. There are advantages to this strategy, not the least of which is the convenience of starting off with a ready-made application accessible through the public IP network. In this scenario, planners need not get involved with software development and implementation nor the agony of setting up a network. This type of a solution will work well when a company wants to set up shop on the Internet and needs the convenience of a ready-made order entry and customer fulfillment application. The readiness of the Internet and a proven application make starting an E-commerce enterprise a painless operation. It may seem surprising, but a large number of so called dot.com start-ups are using this approach.

KNOW WHAT THE COMPANY IS GETTING INTO

However, before plunging into the E-commerce free-for-all, companies need to take careful aim at their marketing objectives.

To begin with, one needs to understand one's business opportunity and whether or not it is truly an electronic opportunity. Might one be setting up another "grave site" or one that will really result in new business? For this solution, one needs a Web developer that knows how to develop a Web site. One also needs to work with someone who knows how to market products. One may have the prettiest site on the World Wide Web; but if one does not get the hits needed to generate the interest required to get business, one will be just another "grave site." Where possible, try to leverage existing legacy applications if they can contribute to

the E-business enterprise. Just because one has an ASP in sight, one may be better off managing in-house. Further, one must understand that any E-business solution needs to be tightly integrated with other business solutions that drive the overall business. Finally, one must be sure that customer, employees, and suppliers will want to use the system. The system should complement one's already successful business practices. This may mean working with the vendors that already service the legacy systems. These vendors know such systems best and may already know one's customers, one's infrastructure, and the solutions that work best for one's company.

BUDGETARY CONSIDERATIONS

Typically, an application service provider (ASP) will provide services from its own stand-alone facilities. Application services can be rented on a per-user basis, per-month basis, or any number of rental/lease arrangements.

Costs for renting software can run from $45 up to $1500 per user, depending on the service requested. However, some observers project the average cost for application services to be closer to $500 per month, depending on the degree of end-user services required. Avcom Technologies has launched an ASP portal designed to allow IT managers to implement rentable applications. There are three portals: MyIntranet, ASPNow, and MyApplications. The MyApplications portal allows a user to log on, be authenticated, and be billed for access to and use of any available application. This single-user, "by the drink" service will cost about $100 per user per month.

ASP service may include co-location and coordination of ongoing support and maintenance for a company's existing application on a shared server. For example, Sunburst Hospitality, owners of EconoLodges and Comfort Inns agreed in 1996 to pay its parent corporation approximately $1.3 million to develop a PeopleSoft financial system to support its operations. Functionally, the system did not work as planned and by late 1998, USinternetworking Inc. (USi), an emerging ASP, was contacted. USi agreed to purchase the PeopleSoft software and put the system up on USi servers. After a three-month conversion period, Sunburst went online in April 1999 with a reported savings of over 20 percent. Thereafter, all of the Sunburst units could access their usual application over the Internet.

It is not uncommon for a small to medium-sized company (one with five or fewer locations) to budget $275,000 to $300,000 per year for MIS personal; $245,000 per year for workstations and servers; and $325,000 for network costs. One such company, with $125 million in sales, decided to develop a special product ordering system to place on its Web site for use by its customers. After three years of mounting development costs (over $1 million), the project was abandoned because the company get the software to run as expected.

With costs like those above, it is entirely reasonable to segregate and identify those applications that can be rented on a per-use basis. If in doubt, try a single application on a per-use basis to determine costs for a six-month trial run. Compare these costs against in-house costs to run the same application on existing systems.

ASPs PROVIDE AN OPTION

For the emerging business wishing to come online with a specific set of remotely accessed business functions, ASPs can provide a viable option. Typically, companies will choose an outsourced software application that requires a high degree of online availability or technical expertise that the company does not have available. However, any move to an outsourced service should not be made until a detailed analysis has been made of the business' information processing requirements. In this regard, there are no short cuts that can be taken if a business is to compete in the marketplace.

Because there are no silver bullets in the information systems (IS) planning process, planners must examine those functional applications that will be required to run the business for the next five years. This is an important first step because planners must clearly understand their requirements before meeting with vendors to discuss outsourced services. Corporate planners familiar with the company's business are in a better position to determine the corporate IS profile and should not approach the vendor community in hopes of "learning" of developing their IS profile.

BUYER BEWARE

ASPs vary widely in terms of the levels of service that they are prepared to offer, because in today's market, detailed business experience is a commodity in short supply. Many of the vendor's personnel may be short on business experience and have little to offer beyond the application on which they are working.

Virtually any software application is available through an ASP, including comprehensive applications software like SAP or J.D. Edward's integrated information systems. However, not every application should be outsourced, and the corporate planner should resist the temptation to outsource all of the company's information processing stream. If there is an absolute need to outsource an application, it is incumbent upon planners to find an ASP with hands-on proven expertise in their specific mission critical applications.

There are a lot of good reasons for this. For example, starting out with an ASP with limited resources can prove disastrous. There were a few good examples of this in the recent case of the U.S. Chamber of Commerce or the United Way.

There are other safeguards that must be taken into consideration. For example, internal proprietary corporate information must be safeguarded against access beyond the corporate suite. This is particularly true where corporate financial data is at stake.

Other information vital to the corporation — like personnel, product design information, detailed sales and customer information — also needs to be protected from intrusion by any disgruntled former employees, competitors, or interests alien to the corporation.

WHAT SHOULD BE OUTSOURCED

In analyzing the corporate information profile, a clear line of demarcation should be made between what is critical to the internal interests of the company and that which is peripheral. Further, if budgets are tight, applications that are not critical to the proprietary information requirements of the company can be considered for outsourcing. For example, applications that are common from one company to another (like e-mail), may best be supported by an outside vendor that can do the job for less money.

Often, standardized day-to-day administrative applications can best be left to an outside vendor. Another common off-the-shelf application that is often outsourced is the payroll function. Firms like ADP have been supporting this important function since the early 1970s and their systems have proven to be absolutely solid.

In recent years, order entry and customer satisfaction or fulfillment systems have reached a high degree of refinement. Any company with such a requirement would be foolish to spend money to develop or maintain a similar system that could more economically be outsourced through an ASP. In years past, companies have made major commitments in such systems that require heavy investments in software and network expertise to operate a broadly accessible public system. While there may be good business reasons to maintain specific applications internally, a case can be made for an ASP-based business solution. The key is to establish a balance between internal resources and those which can more cost effectively augment one's corporate data processing profile.

INTEGRATION WITH EXISTING ENTERPRISE SYSTEMS

There are a large number of firms that still have their legacy systems running on a mainframe or networked AS/400 minicomputers. Some of these companies are rethinking their present legacy systems with an eye to reducing costs through outsourcing some of their MIS operations. It is shortsighted to think that a multimillion dollar mainframe system could be replaced overnight by a few downsized minicomputer servers. Often, these legacy systems have been operating successfully on software sys-

tems that have been programmed to meet very specific business requirements. These systems require a deliberate analysis to determine a migration path on an application-by-application basis. This often requires that a completely parallel application be set up on a separate dedicated server and tested as a beta system first, using a subdivision of the company. This process will ensure that any failure will not bring the company to its knees if the system goes down. Further, such testing will allow the establishment of fail-safe network access arrangements to ensure survivability.

Some large-scale mainframe users have been working with a process known as host publishing using 3270-to-HTML processes to convert 3270 datastreams to HTML. Earliest attempts at this process were fraught with problems because SNA-based function keys, specific printing or file transfers could not be handled effectively. However, specific applications like Novell's Intranet Web Host Publisher and downloadable applets have helped to alleviate some of these difficulties.

Middleware is also available that recognizes several versions of database managers. These middleware systems allow a developer to design, build, and manage standard reports over the Internet. This allows the placement of any number of different reports on the Web, making the generation of paper reports unnecessary. For example, Information Builders offer WebFocus Developer Tools that support the distribution of reports across the Web. Many large firms have started to deploy this technology on a phased basis to test out applications deployed on the Internet.

SELECTING A SITE THAT WILL SURVIVE

One of the principal advantages of an outsourced application service should be its survivability through several disaster scenarios. Because access is their business, network flexibility, salability, and security are some principal advantages of choosing an ASP. However, in planning for the deployment of an ASP-based application, it is important to examine the ASP's provisioning plans and server site (s) very carefully. Any ASP serving site that is not backed up by another remote site capable of taking over in a disaster should not be considered. In this regard, before considering any ASP vendor, planners should visit both their primary and secondary sites and insist on a dynamic recovery demonstration before going further with the vendor. During the site survey, careful attention should be paid to fire protection measures — both internal and external. For example, how fireproof is the site in which the server is located? Is the server facility protected by a halon or similar fire protection system? Is the building in which the ASP's server a concrete or frame building? Is there a fire alarm system within the server site or building in which it is located? How far away is a fire station? Where is the building located? Is

it likely to be flooded in a 100-year storm? If the facility is located in California or Oregon, has the building survived an earthquake?

Next, ask to see the telecommunications arrangements. Is there just one access point between the ASP's server and the Internet, or are alternate access arrangements in place (i.e., satellite, wireless, or alternate telecom path from the server to another serving central office)? Examine also the ASP's peering arrangements for network backup or support for network congestion. Every effort should be made to determine what, if any, spare capacity has been built into the ASP's systems to support expansion of one's application in anticipation of any expansion in one's business!

OTHER CONSIDERATIONS

Upon completion of the survivability evaluation, the planner's next evaluation should be of the ASP's ability to support the application through the several levels of service inauguration. Consider first the ASP's personnel complement. Is there sufficient depth to the ASP's professional staffing levels? Who will be the on-site professionals to support end-user training, resolve hardware interface issues, and the overall management of the application during the implementation process? Would one be comfortable in turning the business over to those people assigned to the project? Remember, while most vendors may stress accessibility via the Internet and a browser, there is no substitute for on-site assistance by a professional who has hands-on experience with one's application.

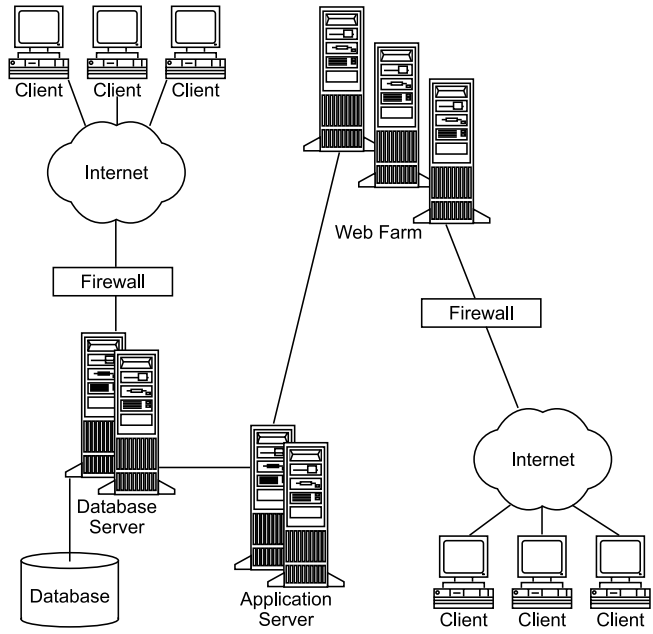
For the most part, it should be assumed that documentation for the application being installed will be inadequate for online, real-time resolution of system problems. For this reason, it is vital that an experienced professional be onsite during the migration to the new system.

APPLICATION HOSTING

Application hosting is another flavor of an ASP service offering. In this scenario, the owner or business has made a commitment to support an in-house professional team to support a specific functional application. The company uses the ASP as an external server site, together with the ASP's access to the Internet to host the application (see Exhibit 1).

In this arrangement, the ASP may supply a database server, application server, or Web farm (servers), all protected by a firewall. Typically, these can be UNIX or NT servers that support an SQL or Oracle database structure. This arrangement often serves to free the company of the burden of maintaining a private network with its attendant support requirements. Further, under a host agreement, the ASP may also be responsible for monitoring network performance levels and interfacing with the necessary carriers and Clecs for all local access arrangements. This may be

EXHIBIT 1 — Application Hosting Application Co-location



preferable where an ASP has a sufficient customer base from which to leverage favorable tariff arrangements with the carriers concerned.

SECURITY IS STILL A CONCERN

As with any computer facility open to the public, network security is still a concern and must be dealt with directly. However, just because one has deployed an ASP does not mean that one's security worries are over. Not in the least!

While the ASP may have made accommodation for security by maintaining firewall arrangements, one must be concerned with authentication, encryption, access levels, ASP sharing arrangements, and the ASP's level of operating security, backup, and recovery.

Now that one's data is naked on the Internet, should everyone have access? Passwords will not do it. One will have to set up a public key infrastructure (PKI) along with some sort of token generator and one-time password setup procedure. Encryption or cryptography must be implemented along with the PKI system. There are several vendors that just specialize in security issues that will need to be consulted long before one's application goes live on the Internet.

Access levels in legacy systems have been well-established as the industry has evolved. For this reason, one will have to test the ability of the new ASP's systems to enforce and directly manage who can read, write, or in anyway modify the data. There a number of products that one may wish to evaluate before relinquishing access control to an ASP. For example, Networld has tested Securant Technologies' ClearTrust and Netegrity's SiteMinder Web authorization tools and have recommended these as products for consideration.

One will also have to evaluate the number of "backdoors" that are open to one's ASP's system and one's data. This includes levels of access to Web pages for updates, remote administration, or other levels of access. In this regard, one may want to engage the services of a professional security analyst who is familiar with all of the present-day hacking methodologies. This is a very important consideration in that someone else may be getting to the data before one can do anything about it.

It is also very important to know how much or to what degree one's ASP is sharing so-called "dedicated" circuits. These arrangements will have a definite impact on one's security system.

The ASP's own internal security arrangements should be of concern, particularly those security arrangements for the operating system that control one's application. Also be aware of the security of the middleware that supports the application. How are updates and level changes controlled and enforced? One will need to know this information because it will affect the terms and conditions put into a service level agreement (SLA).

Now that one's application has been "off-loaded," planners may want to consider a separate and distinct firewall to protect the application. In this arrangement, planners will have to factor regular maintenance of an "owned" firewall, which includes maintaining levels of security.

Redundant backup, as in mirroring or RAID, is another issue that must be carefully defined and established now that one's application and data reside on an off-site server. In this regard, planners must be certain that there are both logical and physical redundancy procedures in place that work. This must be tested before the application and data go live. Clearly, security must be taken as a very guarded internal issue.

HOW GOOD IS YOUR ASP's APPLICATION?

Measuring the effectiveness of one's application as an E-business site may be easy if one is not getting any business. That is simple; one has a "grave site" instead of a Web site. In this situation, one needs to establish with the ASP how one plans to measure the performance of the application. In this situation, one must establish a plan with the ASP to capture very specific information on a regular basis that is key to Web marketing. For example, one will need to know things like which search engines re-

ferred the most customers to the Web site or where the FTP traffic is coming from. One also may want to know where traffic came from that referenced one's site and which pages end users referred to most often. Ideally, one should be able to get HTML reports on a regular or "as requested" basis.

One should also have some reporting of the throughput of the network. For example, is the ASP's network overloading such that packets are being dropped and hence one may be losing key traffic?

Finally, one should set up a plan to monitor the ASP's customer support center. If one's customers cannot use the E-business site, then one will surely lose business. Establish a list of frequently asked user questions, pretend you are a customer, and then try these on the ASP's customer service center. What is the level of response that customers are likely to receive? Would one be happy with what was found? Prompt and courteous customer service should be spelled out in the SLA and penalties assessed for lack of service. In any case, the ASP's application helpline should serve to assist in attaining one's business goals. Should there be any problems with the ASP's customer service center, the SLA should provide language supporting a reduction in service costs for poor customer service — in which case planners may want to set up their own in-house helpline if the application is critical to attaining sales quotas.

Where one may be using an off-site hosting center, and still maintaining much of the technical interface, one may want to have access to an internal technical support help desk. Here again, it is wise to test this service level to be certain that one is getting the service for which one has contracted. In the case of off-site hosting, a 7¥24 customer contact center should exist so when one's server goes down in the middle of the night, one can get it back online. The ASP should make one aware of one's system failures and provide whatever technical support is required to bring the system online.

Basically, through all of this performance reporting, one should be able to get and maintain some sort of feeling of well-being that one's application is performing as advertised.

A WORD ON SLAs

In today's operating environment, when one establishes a wide area network (WAN), one is using the facilities of many different carriers. However, as an end user, one may be dealing with only one vendor, who in turn would have a contractual relationship with all of the carriers supporting one's WAN. Working with an ASP is very much the same situation, in that one will be dealing with a single vendor representing both a wide area network and support for a specific online application. How then does one know what one is getting, and what the penalties should be when the ASP falls below an agreed-upon level of service?

Since the unbundling of network services, many new service providers — including ISPs turned ASPs — are out competing for business. Now more than ever before, the service level agreement (SLA) becomes the most powerful bargaining chip, as well as a legal recourse in any dispute in service levels. Presently, network performance tools have become so sophisticated that poor service levels can easily be monitored at any level. What would one do if there was a network outage? How would it affect one's business? What would be one's burden of proof?

In such a setting, an SLA becomes a vital contract between the user and the service provider. It defines the baseline for service, clearly outlining the penalties the service provider will be required to pay for service levels falling below a defined performance level. While network size or the geographic extension of the network affect levels of service, the common standard is for 99.9 percent uptime over a 30-day period.

Where a packet network supports an online application, there are parameters — such as the time to response, latency, and packet delivery levels — that must be contained in an SLA. These affect network delays as well as network throughput levels and become important to the successful delivery of services. Where Frame Relay, ISDN, and leased line SLA parameters are involved, standards of performance are established, with ATM performance parameters becoming better defined as this technology is more widely deployed. IP network SLAs are now in the development stage as new performance tools are developed to assist customers in evaluating IP network performance levels. These performance levels may be more difficult to build into an SLA. There are no silver bullets in negotiating an SLA to support one's business application. The key is to carefully define all of the hazards that may face one's application once it goes online. Once these hazards have been determined, planners must then work with their ASP to determine penalties for any failures on the part of an ASP that fall below agreed-upon performance levels. This being said, it is important to recognize that ASPs may be very short on experience in managing a wide area network spread across several service providers. For this reason, a corporate planner may want to seek the support of a very experienced consultant who has detailed experience in dealing with carriers and software vendors. There is no substitute for hands-on experience.

SUMMARY

ASP services provide an excellent opportunity for a small to medium-sized business to avoid the costs of setting up a large internal MIS department. However, it is also a time when detailed planning becomes all-important. It is common knowledge that many firms that rush into the E-commerce free-for-all without the proper plan in place, fail rather quickly. This can mean a serious loss in terms of capital and the ability to even

stay in business at all. This brief discussion has outlined some of the key issues that should be addressed with one's ASP, as well as those issues that should be addressed in setting up a service level agreement (SLA) that is equitable and fair. By all means, if one's company does not have the technical expertise to deal with the issues discussed, it would be foolhardy not to obtain outside technical expertise when dealing with an ASP.

Andres Llana, Jr., is a telecommunications consultant with Vermont Studies Group, Inc., in King of Prussia, Pennsylvania. He can be reached at llana@Bellatlantic.net.

94

Application Security

Walter S. Kobus, Jr., CISSP

Application security is broken down into three parts: (1) the application in development, (2) the application in production, and (3) the commercial off-the-shelf software (COTS) application that is introduced into production. Each one requires a different approach to secure the application. As with the Common Criteria ISO 15408, one must develop a security profile or baseline of security requirements and level of reasonability of risk.

The primary goal of application security is that it will operate with what senior management has decided is a reasonable risk to the organization's goals and its strategic business plan. Second, it will ensure that the application, once placed on the targeted platforms, is secure.

Application Security in the Development Life Cycle

In an ideal world, information security starts when senior management is approached to fund the development of a new application. A well-designed application would include at least one document devoted to the application's security posture and plan for managing risks. This is normally referred to as a security plan.[1] However, many application development departments have worried little about application security until the recent advent of Web applications addressing E-commerce. Rather than a firewall guarding the network against a threat, poor coding of Web applications has now caused a new threat to surface: the ability of hacking at the browser level using a Secure Socket Layer (SSL) encrypted path to get access to a Web application and, finally, into the internal databases that support the core business. This threat has required many development shops to start a certification and accreditation (C&A) program or at least address security requirements during the development life cycle.

Security Requirements and Controls

Requirements that need to be addressed in the development cycle are sometimes difficult to keep focused on during all phases. One must remember that the security requirements are, in fact, broken down into two components: (1) security requirements that need to be in place to protect the application during the development life cycle, and (2) the security requirements that will follow the application into the targeted platform in the production environment.

Security Controls in the Development Life Cycle

Security controls in the development life cycle are often confused with the security controls in the production environment. One must remember that they are two separate issues, each with its own security requirements and controls. The following discussion represents some of the more important security application requirements on controls in the development life cycle.

Separation of Duties

There must be a clear separation of duties to prevent important project management controls from being overlooked. For example, in the production environment, developers must not modify production code without going through a change management process. In the development environment, code changes must also follow a development change management process. This becomes especially important when code is written that is highly sensitive, such as a cryptographic module or a calculation routine in a financial application. Therefore, developers must not perform quality assurance (QA) on their own code and must have peer or independent code reviews.

Responsibilities and privileges should be allocated in such a way that prevents an individual or a small group of collaborating individuals from inappropriately controlling multiple key aspects of any process or causing unacceptable harm or loss. Segregation is used to preserve the integrity, availability, and confidentiality of information assets by minimizing opportunities for security incidents, outages, and personnel problems. The risk is when individuals are assigned duties in which they are expected to verify their own work or approve work that accomplishes their goals; hence, the potential to bias the outcome. Separation of duties should be a concern throughout all phases of the development life cycle to ensure no conflict of duties or interests. This security requirement should start at the beginning of the development life cycle in the planning phase. The standard security requirements should be that no individual is assigned a position or responsibility that might result in a conflict of interest to the development of the application. There are several integrated development tools available that help development teams improve their productivity, version control, maintain a separation of duties within and between development phases, create quality software, and provide overall software configuration management through the system's life cycle.

Reporting Security Incidents

During the design, development, and testing of a new application, security incidents may occur. These incidents may result from people granted improper access or successful intrusion into both the software and hardware of a test environment and stealing new code. All security incidents must be tracked and corrective action taken prior to the system being placed into production. The failure to document, assess, and take corrective action on security incidents that arise in the development cycle could lead to the deployment of an application containing serious security exposures. Included are potential damage to the system or information contained within it and a violation of privacy rights.

These types of incidents need to be evaluated for the possible loss of confidentiality, loss of integrity, denial of service, and the risk they present to the business goals in terms of customer trust.

Security incidents can occur at any time during the development life cycle. It is important to inform all development project team members of this potential in the planning phase.

Security Awareness

Security awareness training must be required for all team members working on the development project. If a particular team member does not understand the need for the security controls and the measures implemented, there is a risk that he or she will circumvent or bypass these controls and weaken the security of the application. In short, inadequate security awareness training may translate into inadequate protection mechanisms within the application. The initial security briefing should be conducted during the planning phase, with additional security awareness, as appropriate, throughout the development life cycle. A standard for compliance with the security requirement is to review the security awareness training program to ensure that all project team members are aware of the security policies that apply to the development of the project.

Access

For each application developed, an evaluation must be made to determine who should be granted access to the application or system. A properly completed access form needs to be filled out by the development manager for each member who needs access to the development system and development software package. User identification and an audit trail are essential for adequate accountability during the development life cycle. If this security requirement has not been satisfied, there is a possibility that unauthorized individuals may access the test system and data, thereby learning about the application design. This is of special concern in applications

that are sensitive and critical to the business operations of the organization. Access decisions for team personnel should be made at the assignment stage of the development project and no later than the planning stage of the development life cycle.

Determination of Sensitivity and Criticality

For every application that will be placed into the development and production environments, there must be a determination regarding the sensitivity of the information that will reside on that system and its criticality to the business. A formal letter of determination of sensitivity and criticality is required. This should be done prior to the approval stage of the application by senior management because it will impact resources and money. The letter of determination of sensitivity is based on an analysis of the information processed. This determination should be made prior to any development work on the project and coordinated with the privacy officer or general counsel. The letter of criticality is used to evaluate the criticality of the application and its priority to the business operation. This document should be coordinated with the disaster and contingency officer. Both documents should be distributed to the appropriate IT managers (operations, network, development, and security).

Applications that are sensitive and critical require more care and, consequently, have more security requirements than a nonsensitive or noncritical system. The improper classification of information or criticality in an "undetermined state" could result in users not properly safeguarding information, inadequate security controls implemented, and inadequate protection and recovery mechanisms designed into the application or the targeted platform system.

Labeling Sensitive Information

All sensitive documentation must be properly labeled to inform others of their sensitive nature. Each screen display, report, or document containing sensitive information must have an appropriate label, such as *Sensitive Information* or *Confidential Information*. If labeling is incorrect or has not been performed, there is a risk that sensitive information will be read by those without a need to know when the application moves into production. Labeling should begin at the time that reports, screens, etc., are coded and continue through the system life cycle.

Use of Production Data

If production data is used for developing or testing an application, a letter specifying how the data will be safeguarded is required; and permission is needed from the owner of the data, operations manager, and security. Sensitive production data should not be used to test an application. If, however, production data must be used, it should be modified to remove traceability and protect individual privacy. It may be necessary to use encryption or hash techniques to protect the data. When the development effort is complete, it is important to scrub the hardware and properly dispose of the production data to minimize security risk. The risk of using production data in a development and test environment is that there might be privacy violations that result in a loss of customer and employee trust or violation of law. Development personnel should not have access to sensitive information.

Code Reviews

The security purpose of the application code review is to deter threats under any circumstance; events with the potential to cause harm to the organization through the disclosure, modification, or destruction of information; or by the denial of critical services. Typical threats in an Internet environment include:

- *Component failure.* Failure due to design flaws or hardware/software faults can lead to denial of service or security compromises through the malfunction of a system component. Downtimes of a firewall or false rejections by authorization servers are examples of failures that affect security.
- *Information browsing.* Unauthorized viewing of sensitive information by intruders or legitimate users may occur through a variety of mechanisms.
- *Misuse.* The use of information assets for other than authorized purposes can result in denial of service, increased cost, or damage to reputations. Internal or external users can initiate misuse.

- *Unauthorized deletion, modification, or disclosure of information.* Intentional damage to information assets that result in the loss of integrity or confidentiality of business functions and information.
- *Penetration.* Attacks by unauthorized persons or systems that may result in denial of service or significant increases in incident handling costs.
- *Misrepresentation.* Attempts to masquerade as a legitimate user to steal services or information, or to initiate transactions that result in financial loss or embarrassment to the organization.

An independent review of the application code and application documentation is an attempt to find defects or errors and to assure that the application is coded in a language that has been approved for company development. The reviewer shall assure that the implementation of the application faithfully represents the design. The data owner, in consultation with information security, can then determine whether the risks identified are acceptable or require remediation. Application code reviews are further divided into peer code reviews and independent code reviews, as follows.

- Peer code reviews shall be conducted on all applications developed whether the application is nonsensitive, sensitive, or is defined as a major application. Peer reviews are defined as reviews by a second party and are sometimes referred to as *walk-throughs*. Peer code review shall be incorporated as part of the development life cycle process and shall be conducted at appropriate intervals during the development life cycle process.
- The primary purpose of an independent code review is to identify and correct potential software code problems that might affect the integrity, confidentiality, or availability once the application has been placed into production. The review is intended to provide the company a level of assurance that the application has been designed and constructed in such a way that it will operate as a secure computing environment and maintain employee and public trust. The independent third-party code review process is initiated upon the completion of the application source code and program documentation. This is to ensure that adequate documentation and source code shall be available for the independent code review. Independent code reviews shall be done under the following guidelines:
  — Independent third-party code reviews should be conducted for all Web applications, whether they are classified sensitive or nonsensitive, that are designed for external access (such as E-commerce customers, business partners, etc.). This independent third-party code review should be conducted in addition to the peer code review.
  — Security requirements for cryptographic modules are contained in FIPS 140-2 and can be downloaded at http://csrc.nist.gov/cryptval/ 140-2.htm. When programming a cryptographic module, you will be required to seek independent validation of FIPS 140-2. You can access those approved vendors at http://csrc.nist.gov/cryptval/140-1/ 1401val2001.htm.

Application Security in Production

When an application completes the development life cycle and is ready to move to the targeted production platform, a whole new set of security requirements must be considered. Many of the security requirements require the development manager to coordinate with other IT functions to ensure that the application will be placed into a secure production environment. Exhibit 94.1 shows an example representing an e-mail message addressed to the group maintaining processing hardware to confirm that the application's information, integrity, and availability are assured.

A similar e-mail message could also be sent to the network function requesting the items in Exhibit 94.2.

Commercial Off-The-Shelf Software Application Security

It would be great if all vendors practiced application security and provided their clients with a report of the security requirements and controls that were used and validated. Unfortunately, that is far from the case, except when dealing with cryptographic modules. Every time an organization buys an off-the-shelf software application, it takes risk — risk that the code contains major flaws that could cause a loss in revenue, customer and employee privacy information, etc. This is why it is so important to think of protecting applications using the defense-in-depth methodology. With a tiny hole in Web application code, a hacker can reach right through from the browser to an E-commerce Web site. This is referred to as *Web perversion,* and hackers with a little

EXHIBIT 94.1 Confirmation that the Application's Information, Integrity, and Availability Are Assured

As the development Project Manager of XYZ application, I will need the following number of (NT or UNIX) servers. These servers need to be configured to store and process confidential information and ensure the integrity and the availability of XYZ application. To satisfy the security of the application, I need assurance that these servers will have a minimum security configured as follows:
Password standards
Access standards
Backup and disaster plan
Approved banner log-on server
Surge and power protection for all servers
Latest patches installed
Appropriate shutdown and restart procedures are in place
Appropriate level of auditing is turned on
Appropriate virus protection
Appropriate vendor licenses/copyrights
Physical security of servers
Implementation of system timeout
Object reuse controls
Please indicate whether each security control is in compliance by indicating a "Yes" or "No." If any of the security controls above is not in compliance, please comment as to when the risk will be mitigated. Your prompt reply would be appreciated not later than [date].

EXHIBIT 94.2 Request for Security

As the development Project Manager of XYZ application, I will need the assurance that the production network environment is configured to process confidential information and ensure the integrity and the availability of XYZ application to satisfy the security of the application. The network should have the following minimum security:
Inbound/outbound ports
Access control language
Password standards
Latest patches
Firewall
Configuration
Inbound/outbound services
Architecture provides security protection and avoids single point of failure
Please indicate whether each security control is in compliance by indicating a "Yes" or "No." If any of the security controls above is not in compliance please comment as to when the risk will be mitigated. Your prompt reply would be appreciated not later than [date].

determination can steal digital property, sensitive client information, trade secrets, and goods and services. There are two COTS packages available on the market today to protect E-commerce sites from such attacks. One software program on the market stops application-level attacks by identifying legitimate requests, and another software program automates the manual tasks of auditing Web applications.

Outsourced Development Services

Outsourced development services should be treated no differently than in-house development. Both should adhere to a strict set of security application requirements. In the case of the outsourced development effort, it will be up to technical contract representatives to ensure that all security requirements are addressed and covered during an independent code review. This should be spelled out in the requirements section of the

Request for Proposal. Failure to pass an independent code review then requires a second review, which should be paid for by the contractor as a penalty.

Summary

The three basic areas of applications security — development, production, and commercial off-the-shelf software — are present in all organizations. Some organizations will address application security in all three areas, while others only in one or two areas. Whether an organization develops applications for internal use, for clients as a service company, or for commercial sale, the necessity of practice plays a major role in the area of trust and repeated business. In today's world, organizations are faced with new and old laws that demand assurance that the software was developed with appropriate security requirements and controls. Until now, the majority of developers, pressured by senior management or by marketing concerns, have pushed to get products into production without any guidance of or concern for security requirements or controls. Security now plays a major role in the bottom line of E-commerce and critical infrastructure organizations. In some cases, it can be the leading factor as to whether a company can recover from a cyber-security attack. Represented as a major component in the protection of our critical infrastructure from cyber-security attacks, application security can no longer be an afterthought. Many companies have perceived application security as an afterthought, pushing it aside in order to get a product to market. Security issues were then taken care of through patches and version upgrades. This method rarely worked well, and in the end it led to a lack of customer trust and reflected negatively on the integrity of the development company. The practice of application security as an up-front design consideration can be a marketing advantage to a company. This can be marketed as an added feature so that, when the application is installed on an appropriately secure platform, it will enhance the customer's enterprise security program — not help to compromise it.

Reference

1. NIST Special Publication 800-16, *Guide for Developing Security Plans for Information Technology Systems*, 1999.

95

Covert Channels

Anton Chuvakin, Ph.D., GCIA, GCIH

Although the words "covert channeling" bring up for some people images of spies and evil spirits, the meaning we discuss in this chapter is even more interesting and sometimes even more sinister.

Secret communications, where there is seemingly no communication happening within the same machine or even across the network, can be accomplished with covert channels. Specifically, communication that violates a site security policy despite the deployed technology safeguards is of particular interest.

We should note that we are not talking about steganography, which is mostly about hiding data and not about moving data from place to place. Hidden data can be moved together with the object it is hidden in, but if all such communication is also blocked, steganography just will not help. A covert channel, however, might still be established. To some extent, transmitting data embedded in images via steganography in case such image transfers are allowed would likely constitute a "covert channel" (see the formal definitions below).

First, we would like to introduce some background of the problem of covert channels. Indeed, covert channeling is a problem from the attacker's point of view (how to channel covertly and effectively) and from the defender's point of view (how to detect and prevent such channels).

The notion of covert channels was popularized by the "rainbow series" of the books by the National Computer Security Center (NCSC) affiliated with the National Security Agency (NSA). This series is officially known as the Department of Defense Trusted Computer System Evaluation Criteria (TCSEC). The "Light Pink Book," officially titled *A Guide to Understanding Covert Channel Analysis of Trusted Systems,* contained the definitions, classifications, identification, and handling of covert channels as well as methods to limit the possibilities for covert channeling during the system design phase. It was published in 1993, prior to the snowballing growth of the Internet. Before that time, covert channels were discussed in some computer science publications within academia and the military.[1]

The "Light Pink Book" provides many definitions of the covert channel. For example:

> A communication channel is covert if it is neither designed nor intended to transfer information at all

or a channel

> …using entities not normally viewed as data objects to transfer information from one subject to another.

Currently, covert channels can be viewed as "old" and "new." The classic descriptions from the "Light Pink Book" are not very relevant in today's highly distributed networking environment, where workstations and servers exchange data across WANs and LANs, and multilevel operating systems are all but absent from most computing environments. An ability to signal other users by accessing the swap file or changing an entry in /tmp directory on a UNIX system does not sound like a terrible risk to the E-commerce site. On the other hand, an ability to send information from the customer database in real-time through firewalls while being invisible to the intrusion detection systems might scare many an executive. Thus, old covert channels such as information leaks across the security levels on a multilevel mainframe are likely left in the 1980s, and the new covert channels such as risks of hidden network accesses and invisible tunneling for data theft are here to stay for the foreseeable future. The study of communication in a highly restricted network environment where most normal protocols are blocked and monitored also presents some interest at this time.

Additionally, the fusion of malicious software and autonomous attack agents with covert channels might bring the risk level from "blended threats" (as touted by some security vendors) to a new level and limit the effectiveness of many current security controls.

In spite of the relative obscurity and obsolete nature of classic host-based covert channels, we will review some of the theory behind them and some methods to eliminate such communication during the system design stage. A lot of effort was dedicated to such research in the 1970s, 1980s, and the early 1990s.

The "Light Pink Book," which defined the comprehensive covert channel analysis (CCA), listed the following four objectives of covert channel analysis:[2]

1. Identification of covert channels
2. Determination of covert channels' maximum attainable bandwidth
3. Handling covert channels using a well-defined policy consistent with the TCSEC objectives
4. Generation of assurance evidence to show that all channels are handled according to the policy in force

Just to clarify, the environment in which the described covert channels take place — a secure multilevel OS with mandatory access controls (MAC) — is described by a security policy similar to the following:

- The process at higher security levels can read the objects at lower security levels but cannot write to them (because that will constitute a data leak)
- The process at lower security levels can write to the objects at higher security levels but cannot read them (because that will constitute an access to forbidden information)

Two main types of covert channels identified in the "Light Pink Book" are storage and timing channels. As defined in the book, "a potential covert channel is a storage channel" if its scenario of use

> …involves the direct or indirect writing of a storage location by one process and the direct or indirect reading of the storage location by another process.

That means that the processes communicate by allocating some resource and checking for the evidences of such allocation.

Similarly, "a potential covert channel is a timing channel" if its scenario of use involves a process that

> …signals information to another by modulating its own use of system resources (e.g., CPU time) in such a way that this manipulation affects the real response time observed by the second process.

That means that one process attempts to influence the timing of whatever event is visible to the second process. Examples of both kinds are provided later.

As for countermeasures, early researchers agreed that it is impossible to eliminate covert channels from the system. Some methods (such as avoiding resource sharing completely, usually at some performance penalty) were developed. However, it was deemed more effective to try to reduce their bandwidth. Keeping in mind a particular covert channel, the system designers will introduce noise in the covert information flow, thus hindering the transmission by reducing the bandwidth. By making the channel noisy by adding random delays and other factors into various system processes while keeping the performance adequate, the designers usually managed to reduce the bandwidth of known covert channels. It was also required to carefully document all possible channels discovered during the system design and implementation phases and provide methods to reduce their bandwidth. In many cases, the bandwidth of several bits per second was deemed acceptable, and sometimes even high numbers (such as for systems processing images) were acceptable.

Following are some classic examples of such covert channels. Keep in mind that the described events occur in the multilevel OS platform where the communication between levels is prevented based on the special policy. Thus, the example might sound unimportant and even downright silly for the common commercially available systems, but apparently were viewed as critical in secure OS.

1. One program locks the file for access (such as for writing) from one security level and another one is checking the lock. One bit of information can be transmitted per time unit; file is locked corresponds to 1 and unlocked is 0.
2. One process allocates disk space and another is checking for available space. If the second process fails to allocate, it knows that the first is transmitting the 1, and allocation success indicates 0.
3. The program reads a page of data. When a second program tries to read the same page, it comes quickly (already loaded in memory, 1) or slowly (had to be received from disk, 0). Thus, 1 bit is transmitted between the security levels.

4. The program creates an object, thus exhausting a unique object identifier of some kind (such as a UNIX user ID). The second program also attempts to create such an object and notices the available unique identifier. Thus, it can deduce that the first program actually tried to create an object (1) or that it did not (0).
5. A process tries to unmount a file system, which might or might not be busy. The second process tries to send information by allocating or deallocating disk space on the same file system.

To conclude and to illustrate the relevance (or rather total irrelevance) of these covert channels for modern information systems, one should note that the NSCS' CCA guide applied only to systems rated B2, B3, and A1 by the TCSEC criteria. The TCSEC ratings go (or rather went, since TCSEC is now supplanted by Common Criteria) from the least secure D to C1, C2, B1, B2, B3, and the most secure A1 (see http://www.radium.ncsc.mil/tpep/epl/epl-by-class.html). Most commercial UNIX and NT systems would be rated at C1; some with high-security packs and add-ons get to C2. Few heavily modified UNIX systems rate as B1 and no general-purpose OS ever got to B2. Thus, CCA and covert channels, as defined and evaluated in the "Light Pink Book," have absolutely no relevance in the modern computing environment, perhaps outside the highly restricted government installations using special-purpose operating systems. Additionally, the book directly states that "the notion of covert channels is irrelevant to discretionary security models" such as those used in most commercial OS.

We will now turn to more modern times and look at covert channeling across the protected network. We will first look at covert channels within the basic TCP/IP protocols and then briefly describe the application protocol covert channeling (and tunneling, as its trivial case).

Before we delve into the exciting world of covert network communications, we will briefly review TCP/IP networking, which powers most of today's networks.

Applications communicating over TCP/IP networks use a subset of OSI (Open Systems Interconnection) network protocol layers. Briefly, the application typically communicates using an application layer protocol (such as SMTP, HTTP, POP3, IMAP, SNMP, and many others, both open and proprietary). Such communications (e.g., client requests and server responses) are formed using the rules defined by these application protocols. The application protocol messages (such as a GET request to download a Web page in HTTP) are then encapsulated in the appropriate network layer protocol (such as TCP or UDP). The encapsulation process involves adding headers and footers (in some cases); also, sometimes an intermediate layer (e.g., session or transport, such as SSL or TLS) is also used before the network layer. Further, the TCP or UDP message is encapsulated in the IP message, again adding appropriate protocol headers. Then, depending on the physical transmission media, the IP message, also called a "packet," is encapsulated in the data-link layer (such as the Ethernet, ATM, or Frame Relay) messages, called "frames." Next, it reaches the bottom of the protocol stack at the physical layer, which handles the electrical or optical signals carrying the data through the wire.

Exhibit 95.1 shows an example using the Ethereal protocol analyzer. The picture shows all the protocol layers from telnet (application layer) to the Ethernet frame (physical layer).

We will also look at the headers that are added in the encapsulation process. Exhibit 95.2 shows the structure of the TCP header. Some of the fields in the header are source and destination ports, urgent flag, sequence (SN) and acknowledgment numbers (ACK), offset, options, and others. The field sizes (important for our further analysis) are also shown. For example, the destination or source port is a 16-bit value (ports go from 0 to 65535, which is 2^{16-1}) and the sequence number is a full 32-bit field.

Exhibit 95.3 shows the IP header. Some of the fields in the header are source and destination addresses, version, type of services (TOS, recently also assigned to ECN, explicit congestion notification), padding, length, time-to-live (TTL), identification (IP ID), protocol, options, and others. The field sizes (important for our further analysis) are also shown. For example, the IP ID is a 16-bit field and version is a small, 4-bit field.

Here is how it is relevant to network covert channels. Many of the fields in the TCP (also UDP) and IP headers are somewhat undefined (TOS/ECN), unset (padding), set to random values (the initial sequence number), set to varied values (IP ID), or are optional (such as options). This very important fact creates possibilities for mixing in the information without:

- Breaking the TCP/IP standard (and thus preventing the transmission of the packet)
- Making the packet appear anomalous (and thus triggering the network intrusion detection systems)

For example, whenever a TCP connection is established, a random initial sequence number is generated by the sender for the first packet in the connection (carrying the SYN flag). The following is how such a packet is shown in the tcpdump tool (flags: -vvv):

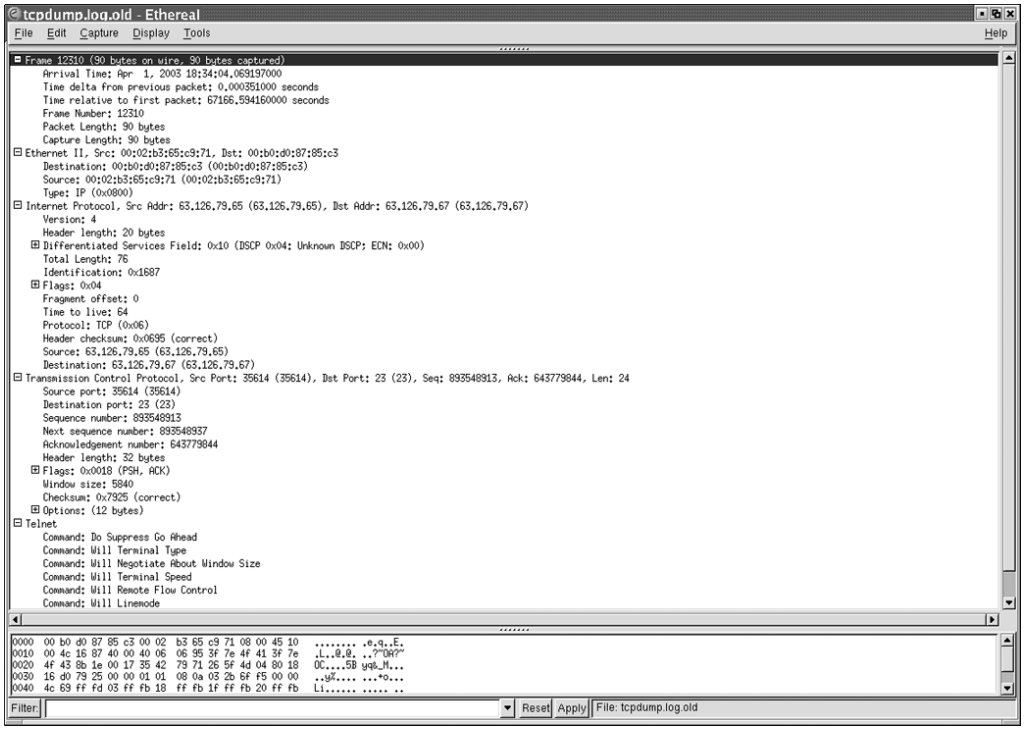


EXHIBIT 95.1 Network protocol encapsulation.

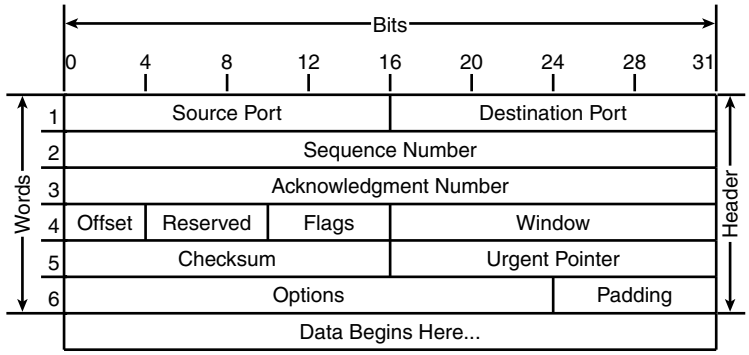


EXHIBIT 95.2 The TCP header structure.

```
11:45:43.965497 src.thisdomain.com.34620 > dst.thatdomain.com.telnet:
S [tcp sum ok]

738144346:738144346(0) win 5840 <mss 1460,sac kOK,timestamp 8566305
0,nop,wscale 0> (DF) [tos 0x10] (ttl 64, id 34427, len 60)
```

The initial sequence number (ISN) is 738144346. It is worth noting that different operating systems use different algorithms for this number generation, from almost-random to deterministic. The covert channel is apparent here: if one is to encode a message (or part of the message) in the ISN, one can carry almost the full 32 bits of information (or less if some random bits are added for higher security) per established TCP session

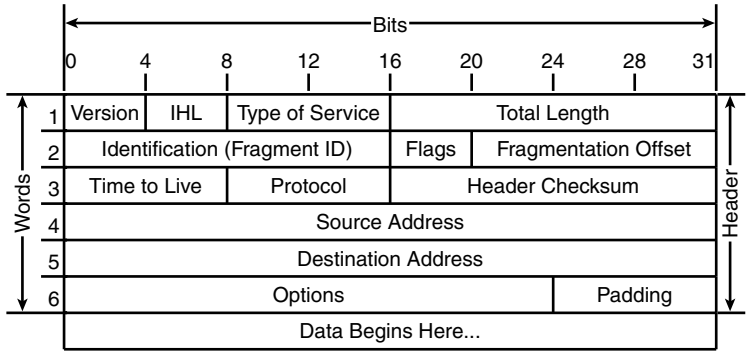


EXHIBIT 95.3 The IP header structure.

(all subsequent sequence numbers are derived from the first one). A similar channel can be established using the acknowledgment sequence number.

This channel is likely impossible to detect and stop, unless a connection goes through an application-level proxy (such as a good proxy firewall) or other device that breaks the original TCP session. Additionally, some NAT (network address translation) implementations might break some of the header fields, such as IP ID.

Sending a lot of information is unlikely with the above channel because one has to establish a lot of TCP sessions, which might appear suspicious. We would like the opportunity to carry data in every packet of the connection and not only in the initial one.

Using the IP ID field was suggested by Rowland.[3] The field may have a nonzero value on any packet, which allows the information transfer of up to 16 bits per packet without raising suspicion, because the IP ID field can have any legitimate value. Such a covert channel is implemented in the covert_tcp program.[3] Application proxy will always break such a covert channel as referenced above.

Covert channels can be significantly improved by adding spoofing and bouncing. Spoofing can help conceal the source of the communication, but can complicate things because response to such communication needs to be picked up off the wire by the sniffer. Spoofing also can help to create diversions by initiating spurious connections to third-party machines not related to the communicating parties. Bouncing (possible with, for example, ACK sequence number channel) works by initiating a spoofed communication with an innocent third party, which would then unwittingly respond to the intended destination of communication. More details on implementing this are also provided by Rowland.[3]

Similarly, encrypting the message before transmitting it over the covert channel is also helpful to add another layer of protection in case the channel is required. It can also help to prevent various man-in-the-middle and message injection attacks, possible in case the channel is discovered.

A detailed look at all the IP, TCP, UDP, ICMP, and other network protocol header options for the purpose of evaluating the potential of covert channels (with suggestions on blocking them) will provide a fascinating area of study, but unfortunately lies outside the scope of the current chapter. One of the efforts that covers many other header fields is found in Hintz.[4]

We should also note that covert communication (while not strictly a covert channel) is possible using the "uncommon" protocols (e.g., NVP, IGMP, EGP, GGP, etc.), which are not expected to carry interactive sessions. A casual look at "/etc/protocols" file on any UNIX machine reveals a long list.

Fortunately, or unfortunately, it depends on the side of the "security equation"; any device that interrupts the flow of the TCP/IP connection at higher layers, such as application proxy (Web proxy, SOCKS, etc.) or a good proxy firewall, will recreate the TCP/IP header and wipe out all the information hidden therein, with the exception of the destination port, which cannot be used for covert channeling due to its fixed value. Additionally, such a device will block the "uncommon" protocols, only allowing the specified list. How can one bypass this limitation? A higher-layer covert channel is the answer.

The trend to tunnel various network protocols over HTTP disturbs many security professionals because "everything over HTTP" means that many new attack vectors become possible through the firewall. This scenario also gives rise to new possibilities for covert channels. A classic example is a flurry of normal HTTP

GET requests (used to fetch the content off the Web server) to specific "scripts" or "Web applications." Many URLs used by today's Web applications are complicated and can be made to carry information. Requesting "http://www.example.com/detail/-/0130259608/102-5403649-1054521?akg" might mean something different from requesting "http://www.example.com/detail/-/0130259608/102-5403649-1054521?bkg," and such long URLs can carry hundreds of bytes of information from the client machine to the malicious server. The response is possible via the pages themselves or via HTTP response codes (200, 302, 403, 404, etc.). Many programs utilizing telnet-like connectivity over the HTTP protocol are known (e.g., see "wwwshell"[5]).

Other application protocols (such as DNS) also open tunneling and covert channeling possibilities. In fact, "telnet over DNS" implementations are known, as are some others (such as "ICMP telnet" or Loki, detected by most current intrusion detection systems). Even "shell over SMTP," i.e., over e-mail, was implemented. Application protocols are well suited for tunneling because such communication can be made to pass through high-security proxy firewalls provided that the rules enforced by the firewalls are not violated. For example, the above HTTP GET methods should be completely transparent to the firewalls. To summarize, we will refer the reader to the humorous example in Waitzman[6], which illustrates that tunneling is possible even in such extreme cases.

Recent advances in application-level tunneling include the "setiri" backdoor, described in Temmingh and Meer.[7] The backdoor utilizes the legitimate network applications to perform HTTP tunneling, thus avoiding not only network, but also host-based security controls.

Another real-life example of covert communication in action includes spoofing an NVP backdoor, discovered and analyzed by the Honeynet Project.

Now let us discuss covert channel risk analysis and countermeasures. As mentioned earlier, the classic host-based covert channels present almost no risk to the modern IT environment. Secure multilevel operating systems, where such channels manifest themselves, are not in widespread use.

The risk of network-based covert channeling is harder to evaluate. Due to the extreme advantage that the attacking party possesses in this case, it is suspected that most cases of covert channel use are never discovered and prevented. Automated attack agents such as worms and Trojans utilizing covert communication would present a high level of risk, provided they are actually discovered and described by anybody. We can only suspect that such methods are indeed used by advanced attackers.

As for preventive measures, keeping in mind that even the "Light Pink Book" authors stated that complete elimination is impossible on the host level, the network environment presents a more formidable challenge. Although system design analysis aimed at preventing some covert channels was conceivable in the tightly-controlled environment of the secure OS, no such analysis is likely to happen on the network. There is simply too much variety in methods of communication occurring on the modern networks.

To some extent, the proxy firewall and a combination of signature-based and anomaly-based intrusion detection systems can help, but infinite possibilities exist for evading such systems by various covert channels. Additionally, inline traffic normalizers (similar to the one proposed in Handley, Paxson, and Kreibich[8]) may serve as an additional layer of protection.

References

1. Lampson, B.W., A Note on the Confinement Problem, *Communications of the ACM,* 16, 10, 613–615, October 1973.
2. A Guide to Understanding Covert Channel Analysis of Trusted Systems, NCSC-TG-030 Version-1.0 ("Light Pink Book"), available at http://www.fas.org/irp/nsa/rainbow/tg030.htm, National Computer Security Center, November 1993.
3. Rowland, C.H., Covert Channels in the TCP/IP Protocol Suite, available at http://www.firstmonday.dk/issues/issue2_5/rowland/, also published in *First Monday,* 2, 5, May 5, 1997.
4. Hintz, D., Covert Channels in TCP and IP Headers, presented at DefCon X conference http://www.defcon.org/images/defcon-10/dc-10-presentations/dc10-hintz-covert.ppt.
5. Reverse WWW Tunnel Backdoor, available at http://www.securiteam.com/tools/5WP08206KU.html.
6. Waitzman, D., A Standard for the Transmission of IP Datagrams on Avian Carriers, available at http://www.ietf.org/rfc/rfc1149.txt, April 1, 1990.

7. Temmingh, R. and Meer, H., Setiri: Advances in Trojan Technology, presented at DefCon X conference, available at http://www.defcon.org/images/defcon-10/dc-10-presentations/dc10-sensepost-setiri.ppt.
8. Handley, M., Paxson, V., and Kreibich, C., Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics, presented at USENIX, available at http://www.icir.org/vern/papers/norm-usenix-sec-01-html/.

96

Security as a Value Enhancer in Application Systems Development

Lowell Bruce McCulley, CISSP

> If carpenters built houses the way programmers build programs, the first woodpecker that came along would destroy civilization.
>
> **— Weinberg's Second Law of Computer Programming**
>
> Woodpeckers are just attempting to remove bugs.
>
> **— Further commentary by Weinberg**

Jerry Weinberg was actually commenting on the state of the art in software engineering in the 1960s, not present-day security engineering, when he authored his second law. The fact that his comment is as pertinent to today's malicious hackers as it was to innocent practitioners of by-gone days illustrates the fundamental truth that security is an inherent attribute of well-designed information systems. His additional commentary points out that systems-engineering activities (e.g., debugging) destabilize systems, clashing with the security imperative for stable systems. This chapter suggests that enlisting woodpeckers (or systems developers) in the security effort benefits both security and development. We posit that it is best to justify information security programs on economic issues in the management hierarchy by showing value from cooperating on technical issues in the project arena. The best way to benefit the development team and the entire organization is by working in harmony with development priorities, so we present several ways to do so.

We begin by surveying the current state of the art in information security programs, in which we identify some things that do not work as well as they might. Economic factors are discussed as the fundamental drivers of management decisions about technology, applications systems, and security. We proceed to an examination of the nature of application systems and associated technologies, to better define our focus and the scope and bounds of our concerns. This leads into a review of the systems development life cycle that applications follow, to understand how the development activities and security concerns change at different stages in the existence of applications systems. Finally, we introduce an innovative approach to using a new security engineering tool in a way that generates value for the systems development process. We close by discussing the integration of that approach into the systems development life cycle, and identifying some potential directions for future research and development.

State of the Art in Business Applications Systems Security

A paradigm shift seems needed in our approach to securing business information systems.

The fundamental shift is to position security as a value enhancer throughout the application systems life cycle, especially the development engineering process. Application systems security would benefit from several effects of this shift, based on decades of experience developing critical systems. The reason is that business

organizations often resist rather than promote security programs, on economic grounds. Application systems are the most important point of focus, because they are the *raison d'etre* for information systems (and thus for information security) in the business world. To successfully accomplish this, we must first understand several things, including economic factors, the nature of application systems and their life cycle, security drivers, and even historical context. This chapter presents a framework and some tools to help integrate security into the application systems development process as a value enhancer.

Dr. Peter Tippett, CTO of TruSecure, recently wrote:

> For years, the focus of most security efforts has been centered on identifying and then fixing vulnerabilities in technology. The prevailing belief is that if a hole is found in the IT armor of an organization, it should be fixed immediately before it can be exploited by some cyber-deviant. While this approach sounds logical and effective, it is actually the beginning of a vicious cycle that occupies vast amounts of time and wastes several millions of corporate, government, and consumer dollars every year.[1]

Dr. Tippett goes on to draw an analogy with healthcare, saying:

> The current approach to security would also have us inoculated for the most minor of illnesses, and protected against every possible cut, bruise, or blister….

which is both ineffective and impractical. Medicine has progressed beyond this piecemeal approach by taking a holistic view of the organism and by emphasizing prevention as the best cure. Unfortunately information security has not followed that model, at least not yet, but it suggests a framework to use as a model to improve our struggling InfoSec efforts. We need to extend our focus to view information systems as functional entities rather than collections of technical components, and to define and address security concerns in that holistic context. By doing so, we also have the opportunity to transform our security efforts from a costly burden into a valuable benefit.

Securing Web-based business-to-business (B2B) E-commerce application systems poses new problems requiring a new approach to engineering security into the application systems development life cycle. A typical Web-based application utilizes external (e.g., Internet) connections from existing segmented network infrastructures that provide a layered defense-in-depth. The external connections are firewalled to protect an exposed demilitarized zone (DMZ) with hardened bastion hosts providing authorized services, monitored by intrusion detection systems (IDS), and isolated from the internal network by additional firewalls. No unnecessary ports are left open, and external network scans will find no vulnerabilities. This effectively isolates the internal systems from the uncontrolled external environment at the network infrastructure level, but at the application level things are different. By design, the Web server provides external connectivity to internal functions because that is the powerful advantage of E-commerce. However, this means that the external users are interacting with database and application servers that are not directly exposed through the infrastructure, but which may now be left exposed to attacks through the application design. The traditional approach of patching components when security vulnerabilities are found is no longer acceptable when those vulnerabilities may be discovered by attacks that disrupt databases critical to production scheduling or supply-chain ordering.

The reason for this situation is that today's integrated business information systems are highly evolved and complex systems of interdependent components structured in a logical organization, not a piecemeal collection of independent components to be patched and secured independently. As the complexity of our systems increases, the difficulty of finding and patching all the chinks in their armor becomes unmanageable. Worse, hidden dependencies arise that prevent recognition of vulnerabilities or prevent the application of patches, as well as obscuring responsibility for maintaining security. These factors all raise the cost of maintaining application systems security, which could be mitigated by more effective consideration of security when developing application systems.

For example, many systems affected by the SQL Slammer worm were reportedly running applications that embedded the affected Microsoft server code. Some of the system owners may not have even known their system was running the Microsoft code as a dependency within another package, which raises the question of whether they or the third-party software vendor (TPSV) bore responsibility for applying the requisite security patches. Many customers turn to TPSVs because the customer technical resources are limited, so they are dependent on the TPSV for support, including security issues associated with TPSV packages. TPSVs cannot blindly pick up patches from platform vendors and apply them to production systems at customer sites, because of risk that the patch may cause unforeseen and undesired side effects. The cost of qualifying vendor patches and applying them at customer sites is economically unpalatable for TPSVs, so it is unlikely that they will assume this role without some prodding. Potential liability exposure might be the necessary incentive, but

reducing the required expense also would reduce the disincentive. Better engineering of security as a part of application systems development could provide this reduction.

The key to engineering security as a part of the application systems development process is to see security as an inherent attribute or characteristic of systems, not a separate feature. Basically, security is a way of expressing the robustness or fragility of systems. Information security concerns are described as confidentiality, availability, and integrity. When any of those is violated and expectations or requirements are not met, it is irrelevant whether they are broken by a malicious actor or the perversity of nature. Downtime, data corruption, and inappropriate disclosure are undesirable because they cause bad effects, not because they are caused by hostile adversaries. This definition makes security a feature that should be addressed within the established application systems development community, not parceled out for assignment to a separate organizational function. Information security practitioners can best promote improved practices by forming cooperative partnerships with application systems development organizations.

As a starting point, consider application security as a systems problem in which the overall security requirements and results are determined by the system environment. This is really another way of saying that appropriate security is accomplished by defense-in-depth, with the defense designed into overall system structure. The appropriate security is determined by application system requirements and implemented by making design tradeoffs and utilizing underlying host and network facilities. For example, consider a sensitive application that sends user IDs and passwords unencrypted over a highly secure network using private protocols. Conventional information security practices might argue that an environment using unencrypted passwords should not be described as highly secure, but, in light of other design features, the cost of encryption is not justified by the value. Overall, the system is sufficiently secure, although one component may be less secure than it might possibly be. The successful security practitioner must understand how much security is enough, and how to accomplish that level of security cost effectively. Exploiting existing processes in the application systems development organization is a good way to accomplish this, and this chapter offers ways to do so.

Economic Factors

In the real world of business organizations, applications are the reason systems get built and deployed, to create and promote real economic value. Management decisions are driven most clearly by economic factors in the business world, but cost–benefit analyses are the underlying decisive factors in most sectors. There are complex psychological factors involved in accepting a certain cost in order to prevent risking an uncertain cost, so justifying the costs of information security programs on the basis of risk and cost avoidance can be difficult. It seems better to understand the forces that drive business initiatives and align security program justifications in harmony with them.

The fundamental issues that motivate the need for continued improvement in applications systems in business are nontechnical in nature. Economics is always the overriding priority, because even long-term strategic initiatives are undertaken in expectation of profitable returns on the investment. This gives systems associated with direct revenue producing activities a high stature, with those involved with handling money equally important (in many but not all companies, sales is more important than finance or operations). Systems dealing with cost containment and organizational overhead are not as high a priority, which may be significant to security program investments. Competitive advantage is a significant priority, because it generates economic benefits. Managers are always under pressure to reduce costs, and schedule is a cost, so managers are also pressed to shorten delivery dates as much as possible. All of these factors work against an isolated information security program that presents a clearly measurable cost against benefits of uncertain economic value, and make it desirable to find ways to use security programs to add measurable value.

Costs of developing information systems are a particularly difficult issue for most organizations, because of a number of inherent factors. Systems development is a highly specialized technical discipline that requires creative problem solving. The combination of discipline and creativity is not easily managed, leading to frequent schedule problems and associated budget overruns. Until a system is completed, the development results are not apparent, which forces management to expect success in large part based only on faith in the developers. These factors make development managers especially sensitive to issues that might affect schedule and costs. Security requirements introduce additional complexity and requirements into an already-difficult development environment, so information security programs are often not embraced enthusiastically by systems developers. Using security initiatives to help facilitate meeting development schedules and budget requirements is a desirable alternative that improves teamwork.

Experience has consistently shown that the cost of fixing problems scales dramatically upward later in the application systems life cycle. Obviously, the cost of fixing a problem in design is much less than the cost of finding and fixing it once the system is built and in QA testing, and the cost of finding and fixing it once the system is in production use is even more. As a rough rule of thumb, the cost of fixing problems increases by an order of magnitude, or is about ten times as much, for each stage later in the life cycle that the problem is found and fixed. Doing it right the first time is easiest and cheapest! This is really the fundamental drawback in the common approach to fixing security flaws as they are found in the field.

This phenomenon provides a great opportunity to turn the situation around and use security engineering to contribute positive value during the development process. By providing tools and techniques to identify and fix problems earlier in the system life cycle, security engineering can help to reduce the costs of those problems. For a simple example, buffer overruns frequently are the cause of vulnerabilities exploited by malicious adversaries, but they are also a cause of failures due to inadvertent errors, so they are undesirable because they cause a variety of problems. Thus, QA should and often does test for such scenarios. If QA is testing for buffer overruns, it will be much less expensive for developers to diligently avoid creating any that reach QA. That means using design and implementation techniques that prevent them and development tools that automatically recognize and test for them. This simple example shows good development engineering practice as well as purely information security considerations, but it illustrates the potential value that security engineering can provide by helping to reduce the cost of developing robust systems.

One major contributor to the cost escalation as problems are found and fixed later in the life cycle is the investment in schedule resources. Personnel and equipment have associated costs that must be accrued over time, so any extension of the schedule causes an increase in costs. This is a very important point for security practitioners to consider in their interaction with development organizations, because schedule is a very important and sensitive issue for developers. Any perception by the development team that security measures might cause delays or impede schedule progress is likely to lead to an adversarial relationship between the developers and the security practitioners. On the other hand, sensitivity to schedule issues and helpful cooperation in seeking to improve schedule performance will engender a much more positive relationship. Because many of the security concerns, especially those associated with availability and integrity, are also aspects of robust, reliable application systems, promoting good information security practices will contribute to improving quality without impacting schedule.

One particular issue around schedule may be a particular concern and an especially sensitive issue for the security practitioner to consider in certain development organizations. Software developers make a distinction between software prototypes, which are "quick and dirty" implementations used to explore design alternatives and evaluate their characteristics, and production-quality code that refines the chosen design alternative into a solid, robust implementation. A frequent issue is the pressure to take software prototypes to release prematurely, before refinements such as error checking or buffer bounds checking are added. A software development methodology referred to by terms such as "rapid deployment" or "extreme development" has gained some vogue, based on alleged cost reductions realized from dramatic schedule reductions. This methodology purports to reduce time and cost spent in development by using a quick turnaround to reduce the cost of fixing only those problems that are found to occur in production operations (the argument is "why waste time designing out problems that may never occur?"). This may simply hide costs by shifting them from development to operations or applications users, which is where the effects of production problems will be borne. The security risk is that such extreme development methodologies may be encouraging bad behavior (in slighting design and QA) for schedule rewards at the expense of introducing vulnerabilities that will only be recognized when they are exposed by operational incidents. These methodologies may have value to the organization, but need to be scrutinized carefully for total life-cycle cost justifications. Security practitioners should be aware that such "bleeding edge" approaches are often extremely attractive to the creative technical personnel on development teams so that related issues (such as security compromises) may turn into political hot potatoes.

To summarize, the main factors that are the drivers for business applications of information systems are nontechnical and primarily economic in nature. Direct financial impacts such as revenues and cost are extremely important, and strategic issues such as agility and competitive position are also very significant. These needs motivate the need for applications systems and also shape the organizational environment and life cycle of such systems. Businesses will always want better systems sooner and cheaper, so anything contrary to those imperatives will be swimming against the tide. Information security practitioners need to align their efforts to promote these business priorities and position themselves in the mainstream of organizational efforts supporting those priorities in order to effectively accomplish the mission of protecting the information assets

of the organization. One way to accomplish this is to take the role of collaborator and promoter or evangelist preaching value of security and cost of insecurity within the application systems development community.

Application Systems Technology Base

It is important to remember that applications are the reason systems get built and deployed, to create and promote real business value. All the technology involved is simply a means to the end of delivering application functions to the users that benefit from their value. The systems environment, including the operating system kernel, utilities and administrative tools, user interfaces, software environments, network infrastructure, etc., is just the overhead required to deliver applications and realize the value that justifies their existence. Information systems security seeks to protect the components comprising the application systems environment for two basic reasons: (1) to keep them from being used to mount attacks and (2) because they are needed by applications. Protecting those components is a means to the end of safeguarding business information assets, not an end in itself.

Business information assets exist within the context of information systems. Safeguarding those assets is accomplished by protecting the information systems that contain them. In seeking to do so, it is helpful to understand the nature of the information systems as well as the information assets we seek to protect. This section presents a discussion of information systems theory and practice, focused on some features of great practical importance to applications and to security.

In the most general meaning, systems are a collection of functional elements organized in structure so that they interact to perform a particular function or task. Elements are often modular subsystems that can be viewed as independent systems themselves. Thus, a distributed application system may be comprised of network elements such as hosts and servers that are also individual systems operating in a network environment. The view of systems as a collection of subsystems that may be considered as independent systems themselves has some very important consequences that must be understood by the security practitioner concerned with systems security.

For one, a complex networked system may be a fragile assembly of robust components, because the structure and interactions of components are essential for the proper function of the system. The common approach of fixing security vulnerabilities as they are discovered has the effect of hardening the local components at the level of the patch, but not necessarily improving the security of the systems that incorporate those components. For example, a buffer overflow attack is a way of circumventing access controls on a hardened network. Using permitted traffic to carry malicious content through the controls on secured channels, in order to ultimately exploit an implementation flaw, allows the perpetrator to break containment and obtain unsecured access on a bastion host within a secured perimeter. Arguably, the implementation flaw could be said to make the network vulnerable instead of secure, but the vulnerability could be masked by filtering malformed traffic within the network instead of exposing the flawed implementation to potentially hostile input. The point is that the network system as a whole may be more or less vulnerable, independent of any one component.

Another consequence of viewing systems as a collection of subsystems is that it creates a hierarchical relationship in which it is essential to define the appropriate level of discussion in order to establish the scope and bounds of the system entities. This is extremely important for the development process, because the most common approach to developing information systems is to define modular functions that are subsequently refined and arranged in structures of increasing complexity. Managing this process and the resulting complexity is one of the major challenges in the field of business information systems, and especially in systems development. Failure to adequately meet this challenge may be the underlying cause of most security vulnerabilities.

One approach to managing this complexity is to view the hierarchical structure of information systems in an orderly sequence from a particular perspective. Two perspectives commonly encountered are top down and bottom up. Top-down design generates abstract systems design, broken down into software subsystems of programs and data structures. Bottom-up construction assembles physical resources into networks that run programs and communicate data. The software engineering process designs application systems from the top down and builds them from the bottom up.

Another way to express this is to consider that automated information systems exist at the intersection of a top-down perspective that describes abstract logical design and a bottom-up view of concrete physical implementation. The top-down approach deals with functional business information systems (e.g., payroll, order entry, etc.) and the bottom-up approach deals with programs and data on networked hardware and software systems.

This creates an ambiguity that commonly leads to confusion over which view is meant when referring to systems, e.g., identifying systems for a security assessment. Do we mean the logical business function or the software and hardware that implement it? Evaluating access controls on a distributed ERP application is not the same as evaluating access controls on the networked servers hosting it. The security practitioner must clearly understand and communicate which perspective is intended when the context does not sufficiently identify the reference to make it unambiguous.

Information security practitioners need to take both views into account. Effective security programs must consider the value at risk, which can really only be determined based on the business functions expressed in the top-down perspective, and the cost of protecting the information assets, which depends on the implementation details embodied in the bottom-up view. The challenge is to secure applications by incorporating security as an integral part of the engineering process that develops and integrates both the top-down design and the bottom-up implementation of application systems.

There are also two phases of an application system's life during which different security concerns should be considered. Most commonly, application systems security is focused on the application during production operations, as this is when the application is performing its function of generating value (and thus, where it spends most of its lifetime). The development of application systems is generally considered separately, more as a production application of development tools and systems than in the context of the application being developed. This may minimize several important concerns. For one thing, security breaches during development may disclose or introduce vulnerabilities in the application itself ("dumpster diving" is an exploit that may target development documentation to identify vulnerabilities to be attacked in the application system product). For another thing, the development process may interact with production operations during design, testing, and deployment in ways that create or expose vulnerabilities in the production environment. For those reasons, application development should be considered in conjunction with the operational application systems by security practitioners concerned with the security of such systems. This is particularly challenging because the nature of development organizations and activities is distinctly different from production operations. It may be best to avoid tackling security issues in the development environment head-on and instead cooperatively team with developers to focus on improving security of the resulting application systems, while also seeking to indirectly improve development environment security (awareness and influence will be more effective with the developer personalities than direct authority).

Application Systems Components

Application systems may be comprised of a tremendous variety of components or subsystems, each of which introduces its own particular issues and concerns regarding security. In addition, the relationships and interactions among components also introduce further security complications. Developers who might be ignorant of security considerations may overlook or underestimate the importance of these issues. The security practitioner should be aware of the nature of major components that frequently comprise application systems, and have some acquaintance with the security issues that might be associated with them.

A superficial survey of the various components associated with applications systems is provided in this section, as an introduction to the many aspects that need to be considered both by application developers and security practitioners. The full range of components potentially comprising application systems includes hardware and firmware, operating system components (kernel, drivers, memory management), process management software (loader, scheduler, termination handler, core dumper), file system, command interpreter (shell), utilities, system runtime environment (environment variables, ports, configuration parameters), network protocol stacks, database software (e.g., SQL [Structured Query Language]); user interfaces (GUIs [graphical user interfaces], command shells), help systems, runtime systems (language support libraries, object management systems), development tools (compilers, source management tools, profilers, debuggers, linkers, diagnostics), console management tools (backup utilities, remote administration packages, configuration management and remote deployment facilities, load managers, event loggers, tools, user account managers), and the organizational environment (management, operations personnel, users, developers, vendor support staff, etc.).

The foundation for any system is the hardware used to implement it. Unfortunately, there are often features designed into the hardware to support security that are not utilized within the systems and application software. Sometimes the features are ignored by the software environment; others are more or less fully supported by the basic system software, but hidden or unutilized in other software components. Some hardware provides

extremely flexible features that are normally utilized in a standard fashion, but can be used in other ways. This may camouflage security risks, because many users and technical staff may be unaware of the potential for alternative usages. An example is network interface cards (NICs) for Ethernet, which implement a media access control (MAC) address that is hard-coded by the manufacturer and encodes the manufacturer ID. However, the Ethernet chips used in some NIC cards allow the MAC address to be set to other arbitrary values by running software, which could introduce unrecognized security vulnerabilities in some systems.

Most intelligent hardware devices employ embedded firmware implementing the necessary system processing and control features. In the case of stand-alone network hardware, this firmware may embody the entire special purpose operating system required to install, configure, operate, maintain, and manage the device. General purpose computers incorporate firmware to extend basic hardware functions; for example, the NIC card MAC address functionality previously described is implemented by a combination of hardware and firmware. Differing firmware revision levels may introduce inconsistent security features, either fixing previously discovered vulnerabilities or introducing new ones. (A pseudo-scientific law of computer programming states that fixing any bug simply replaces it with two smaller bugs!) Firmware configuration management introduces potential security vulnerabilities. An example of the security vulnerabilities associated with firmware features would be the viruses that rewrite the firmware in the boot ROM to substitute virus code.

Operating system software provides functions to extend the basic hardware environment to provide more conveniently usable features for general purpose uses. The major operating system software consists of a kernel implementing I/O facilities, memory management, CPU scheduling, device drivers, file system code, and process management (loader, scheduler, termination handler, and perhaps a core dumper). The basic facilities to support user authentication, authorization, and access control, or privileges and protections, are provided by operating systems functions. In addition, the associated command interpreters (or shell) and utilities may be considered part of the operating system, although the distinction between bundled and unbundled system components becomes very indistinct in this area. This feature is often exploited by intruders who replace bundled system components with modified versions to cover their tracks or introduce additional vulnerabilities. The operating system environment is often considered as separate and distinct from applications systems components, although it really is an essential element determining the fundamental security characteristics presented to the application system. Many security problems result from attacks that exploit vulnerabilities in applications or utilities to break out of the software function, to gain access to unintended and unrestricted operating systems capabilities. The capabilities exposed to such exploits are determined by how the application systems developers have utilized the underlying operating system features, but generally they are very significant concerns for the security practitioner.

Network protocols are an essential element of distributed systems, generally following the layered architecture made famous by the ISO Open System Interconnection (OSI) protocol stack model. Internet protocols based on TCP/IP have become ubiquitous, but other protocol models still are used, although less widely. Many older protocols that once used an entirely proprietary stack have substituted TCP/IP for lower layers while retaining their distinct higher-level functional interfaces. There are many security concerns associated with network protocols. The criticality of their functions and their nature as communications media make them especially attractive targets for attacks, both as an end objective (e.g., denial of service, data theft) and as a stepping stone (e.g., worm vectors, relay systems). Because of this, network security is a separate specialized field, but the dependency on network protocols by distributed applications systems forces consideration of protocols as an important factor relevant to application security. The tight integration of network protocols with local I/O in some modern operating systems makes it easy to inject malicious input from remote sources. This is exploited by attacks such as relatively low-level buffer overflows and higher-level cross-site scripting attacks. Network protocols are extremely flexible and must be carefully considered for potentially dangerous interactions with applications systems. This is one reason that it is imperative to ensure that any protocols received by a system must be properly handled (i.e., no unnecessary open ports listening for TCP/IP input, and all services on required ports properly configured for security).

GUIs are commonly used for interactive applications, utilities, and commands in modern systems. It is important to keep in mind that many systems incorporate software that uses command line interfaces, either because they were developed before GUIs were so common (legacy code), or because command lines are more convenient for expert users and automated scripting. Such hidden non-GUI interfaces may provide targets for attackers, especially using network protocols to inject malicious input. Developers of new programs providing such interfaces for scripting convenience may assume that all input will come from local (and thus trusted) sources, and therefore not provide careful input validation and buffer checking, thus creating potential

vulnerabilities to remote attackers or malicious local users. Because system designers frequently separate user interfaces as front-end GUIs from back-end processing of application business logic, this should be an area of particular concern for application systems security.

Database software, such as SQL processors, is an essential component of many application systems, and, as such, must be a major security concern. SQL packages may themselves be subsystems including multiple components, and the interaction between these components may have important security implications. For example, the SQL Slammer worm exploited a vulnerability in an SQL component interface in order to cause malicious commands to be executed by other system components. This vulnerability was present not only in stand-alone SQL servers, but also in embedded database components hidden within packaged application systems.

There is a help system provided with most modern application systems and GUIs, to provide context-specific assistance to the application users. This is not normally considered a security concern, and has not been an attractive target for exploits. There is a slight possibility that the components used to provide application help could have vulnerabilities that might be subject to some attacks, but this seems fairly insignificant. A more significant concern might be the potential for inappropriate disclosure of information through context-specific help facilities, especially if the help facilities also provide an interface to remote diagnostic and support tools. In general, this area is probably not a major application systems security concern, but at the same time it should not be completely forgotten.

The runtime execution environment within a system consists of the various parameters that are used to set variable values controlling system functions; for example, the IP address of a networked host. Many of these configuration parameters are stored in some nonvolatile format (e.g., parameter files) and then used to initialize values for dynamic elements of the system. The configuration files may be read and interpreted by a script processor (e.g., through the command shell) or directly by the associated program itself. Sometimes the values are stored in environment variables to make them accessible over a longer period of time within the executing system environment. The contents of environment variables and configuration files are subject to attack and may provide avenues for exploits. These features are provided by the operating system and are subject to whatever access controls are implemented in that system and used by the developers of the particular features. An important issue regarding system privilege and protection mechanisms is that developers often find finely granular mechanisms cumbersome and inconvenient and thus may use shortcuts such as elevated privilege or less protection to reduce implementation efforts at the expense of security. Such features are usually considered internal details that are not exposed to external threats and thus may not be protected beyond "security through obscurity," which may leave vulnerabilities such as the potential for scripts to inject malicious commands (frequently executed with elevated privilege or undesirable account context). Also, inappropriate modification of these component values could well result in denial of service. The application systems security concerns associated with these features are certainly significant, but the relative obscurity of any vulnerabilities helps to moderate the priority of those concerns.

Modern software engineering seeks to abstract logical representations of function from the concrete (albeit virtual) resources used to implement those functions. As a consequence, application development tools such as object-oriented environments include extensive runtime support, which is often hidden even from the application developers. From a software engineering perspective, this is desirable as a means of hiding complexity, but from a security perspective this has the undesirable consequence of hiding dependencies and possible vulnerabilities. Object reuse is a major priority for reducing development costs, and this requires the most general and least constrained implementations. As a result, bounds and value checking may be compromised or complicated because the specific validation requirements often depend on the particular usage. It is not possible to effectively perform some validation (such as buffer size) external to the module or object using the values, but it may be more complicated to implement an effective check at the site of usage for arguments supplied externally by an invoking object or module. The security concerns in this area seem to be primarily focused on denial-of-service possibilities, although there should also be some awareness of dependencies on external vendors to provide secure components and eliminate vulnerabilities in their object management and compiler runtime systems. A related area of concern is the use of dynamic linked libraries (DLLs) in some systems, which provides a potential vulnerability for substitution of components incorporating malicious code in place of the original trusted components. This could be utilized by "root kits" installed to further exploit a compromised system. Application systems would be vulnerable to this exploit, although it may be more likely to target bundled host system components that are more widely known to attackers.

Management and operational support tools are essential components associated with any significant application systems, especially in a distributed network environment that may use "lights out" data center

practices. The phrase "lights out" refers to data centers running 24/7 without being staffed 24/7, depending on automated management tools to allow remote administration by remote operations centers with online monitoring, or on-call operations personnel alerted using pagers. Event loggers, reporting and filtering tools, centralized monitors, and remote access to management consoles are all elements of the management systems used to support online operations for network systems delivering critical applications. These components are especially critical because they are vital to maintaining security of applications systems, and they are complex and subject to vulnerabilities themselves. The good news is that management systems are frequently supplied by major vendors who recognize the critical role of such systems and are committed to their security. The bad news is that such powerful management systems may introduce vulnerabilities especially to application dependencies (the most common denial-of-service attacks are those inadvertently perpetrated by system and network administrators making mistakes during routine operations). Other management and operational support tools include backup utilities, load managers, deployment and configuration management tools, and user account managers. Such tools are obviously significant security concerns, but those concerns may not have received the same scrutiny for isolated functional utilities as they do for centralized console managers. For example, in small organizations or for less-visible applications, backups may be routinely performed but never tested. Failure modes need to be considered as potential security issues, so that a network glitch during a remote upgrade does not result in a complete denial of service (such considerations highlight the indistinct boundary between security and application design and implementation). The security practitioner concerned with application systems security needs to be very aware of and concerned about these tools, and may want to enlist operations and development staff to cooperatively review and address security implications in these areas.

As previously mentioned, applications systems development presents a unique environment with its own set of security considerations. Development tools include source management packages, compilers, linkers, profilers, debuggers, diagnostics, and many other utilities. In addition, developers and QA testers may need the ability to manipulate the running system environment in ways that production operations and ordinary users do not require (e.g., to set up or recover from specific test scenarios), and thus may be routinely granted access to use privileges that present security concerns. Because of this, development systems and accounts may be particularly attractive and valuable targets for attackers. There may also be vulnerabilities exposed in the development environment and process that are not present in production operations; for example, if samples of production data are used for testing without ensuring that appropriate protection is provided for sensitive content. This problem may be exacerbated once applications systems move to production, because problems during production may require access to sensitive data or even to production systems. Normally, a well-managed development organization will be effectively isolated from production to minimize security exposures, but this discipline comes at a cost and is especially subject to compromise when problems occur. Such situations require heightened awareness of security issues by all personnel involved (and, of course, entail a heightened stress level that makes everyone less receptive to reminders, highlighting the importance of cultivating routine awareness of good practice).

Finally, no application system functions in a vacuum. Applications systems exist to serve human purposes in some form or fashion. The interactions with humans occur within an organizational environment and culture that defines the fundamental security context that must be considered by any effective practitioner. The organization includes management, users, operations personnel, developers, and external personnel such as vendor support staff. Each has their own function and may place their job as a higher priority than security, so it is human nature that they may take shortcuts for convenience or intentionally or unintentionally compromise security in other ways. The security practitioner must remember that the goal of security is to protect the utility of systems to the organization, which requires promoting awareness of security considerations by all personnel. Most importantly, the practitioner must remember that the greatest utility is likely not the most secure system, but one with carefully considered security policies and practices that are appropriate to the system and organization. The reason for cooperatively integrating application systems security concerns into the development process is to properly establish the most appropriate security posture and to effectively implement it.

Technical Concerns for Application Systems

Some specific technical areas frequently cause security issues within application systems. This may be caused by the characteristics of the technical features involved (difficulty of use or complexity of feature), the nature

of the use, or the limitations of application developers. Some particular concerns are input validation (filter for illegal values as well as protecting for buffer overflows), memory management (especially buffer overflow protection, but also stale data violating confidentiality, etc.), authentication/authorization/access AAA control (application implementations often trade strength for user convenience), session management (HTTP is stateless, so cookies are used to provide persistent context with extremely weak AAA), and configuration management (change control and QA to prevent insecure software in production). Security practitioners need to focus attention on these issues during design, development, and testing, to avoid the costly problems surfacing later in the life cycle. Designing sound solutions in these areas will help make implementation and testing easier, benefiting the entire team.

Application packages provided to third parties (including separate organizational entities within the same corporate umbrella) should specifically identify dependencies on platform and external package features in sufficient detail to understand security issues associated with those dependencies (including but not limited to potential denial-of-service attacks). Application providers should disclose such details and their clients or customers should insist on disclosure. Internally within development organizations, engineers should document, test, and monitor security of all dependency interfaces.

Application Systems Development Life Cycle (SDLC)

The existence of such application systems follows a very well-understood life cycle, initially determining and specifying functional requirements for the system to be implemented. This initial functional design phase moves into an implementation design phase, which determines the technical details that will be used to implement the system. The implementation design proceeds into a development process that further refines and arranges details of technical components to create the requisite functionality required by the initial functional specifications to answer business requirements. There is an iterative process of development and testing for both individual components and the entire system as implementation progresses, to assure satisfactory quality before release for production operations.

When the QA function determines that testing has found that requirements have been successfully met for satisfactory production operations, the application system is released for deployment to production. This stage of the SDLC is sometimes called release engineering, for obvious reasons. Production deployment may be a simple transition of starting to use a new system, or it may require a very extensive process of parallel testing and progressive migration of critical functions onto the new implementation with provisions for falling back to previously used systems in the event of problems. The deployment into production requires updating configuration management systems used to control production systems, and often uses automated tools to install the appropriate configuration on production systems automatically. There may be provisions for backing out of releases especially in extremely critical production operations, to ensure that any new release does not cause unforeseen problems (e.g., the scale of production traffic may be difficult to reproduce in QA, leaving the potential for unrecognized problems caused by volume over time).

Upon the ultimate completion of production deployment, the application system enters routine production operations and maintenance. During this phase, requirements may evolve (e.g., rules for regulatory compliance may change slightly) and new or unusual situations may reveal flaws in the design or implementation that were not caught before release. These occurrences will require some maintenance upgrades to the production application system, so production operations are often referred to as the maintenance phase of the system development life cycle. Any changes will normally require appropriate testing before release, and should follow release engineering procedures similar to major new systems.

Security practitioners concerned with disaster recovery and business continuity planning need to be especially interested in the interaction of release engineering and deployment with configuration management and console operations tools. One powerful motivator for automating configuration changes and management is the impossibility of recovering to an unknown configuration following any disaster! On a less dramatic scale, problems affecting routine system updates can have a costly ripple effect if the recovery from problems interferes with continuity of routine business operations. For example, if a network glitch interrupts the routine deployment of an automated update to a production server, the server may be left in an insecure state or simply unavailable until manual intervention restores a serviceable configuration. Preventing such situations (or recognizing and remedying them) is an opportunity to add value beneficial to the entire organization.

Ultimately, the cycle ends when changing business requirements or technology motivate replacement or major enhancement of the production application system, and a new development cycle will be initiated, with deployment of the new system leading to replacement of its predecessor. Sometimes the functions provided by the application system will no longer be needed and the retirement of the system will not include any replacement. This situation can lead to legacy systems becoming unused and forgotten but not removed, with an increased risk that inattention will lead to insecurity.

Integrating Security into the Systems Life Cycle

The introduction to this chapter discussed the historical approach of information security programs, focusing efforts and resources bottom up, on technical components rather than taking a holistic systems-oriented view of the problem. This approach is appropriate during the operational phase of the systems life cycle, but as the discussion about economic factors showed, retrofitting security with patches after system deployment is woefully expensive as well as fundamentally ineffective because of the nature of systems themselves. The paradigm shift suggested at the beginning of this chapter focuses on integrating security into all phases of the systems development life cycle as a way to provide more cost-effective improvements in application system security.

Treating security as a separate issue assigned to an isolated organizational unit creates a situation in which the security function too often ends up the antagonist of developers in the application systems development process. Because the development team goal is to ship the product as soon as possible, imposing security requirements on the implementation design seems a costly impediment to achieving that goal. However, as we have seen, the development team and the information security practitioner share a common interest in deploying robust systems, because availability and integrity are fundamental requirements for a functional system. Confidentiality is also a common interest, but based on separate business issues of competition, compliance, customer care (or privacy), which might be called the "four Cs" of confidentiality.

Benefits from including security in the entire system development life cycle start with the early top-down engineering design process, by helping to design robust systems more cost effectively. As previously discussed, system development economics benefit greatly by meeting requirements earlier in the development process instead of reworking designs to fix shortcomings later. Presenting security requirements as metrics of robust quality early in the process motivates good practice in a cooperative rather than an antagonistic fashion. Throughout the development process, security considerations can be used to focus attention on critical aspects of the application system to improve product quality while avoiding costs for later patchwork. Overall, security can be an enabler of better performance by development teams, improving quality without impacting schedule, by better identifying and addressing critical concerns affecting robust quality.

Different stages in the application systems development life cycle have different security requirements and present different security challenges. Requirements documents and functional specifications are frequently housed on centralized document management or groupware systems, so security administration is not particularly challenging. Development hosts often present a particularly challenging technical environment, because creative systems developers are often inclined to push the limits both organizationally as well as technically. There is often friction between system administrators responsible for development systems and the developers using those systems, especially when powerful desktop workstations are used to facilitate development in a centrally managed network environment. Systems used for testing and quality assurance are usually much more cut-and-dried in their security requirements, because they normally should use environments identical to production as much as possible (exceptions should be clearly justified, perhaps by test management toolset requirements).

Deployment, or release engineering, is the interface and transition between development and production. Because they are responsible for moving system packages that have completed testing into production, security is a routine concern to which the users of these systems are well attuned. The security practitioner should keep in mind that these systems may not be monitored in the same way that production operations are monitored, although they would be high-value targets for an adversary seeking to inject malicious code into the production environment, or to simply disrupt production by causing unserviceable components to be released. Also, careful management of deployed configurations is an essential requirement for successful disaster recovery efforts, because it is impossible to recover to an unknown configuration.

The operations phase of the systems development life cycle is the usual focus of information security programs, so it is regarded as outside the scope of this chapter except for one aspect. Failures occurring during

production operations may require unusual diagnostic or emergency maintenance activities that force exceptions to normal operational security practices, or involve development or vendor personnel. These situations may cause unforeseen security implications, such as the potential exposure of confidential information contained in diagnostic files (e.g., core dumps) transmitted outside the normal security perimeter. Pressure to get corrections into production may lead to compromises in security, and such issues need to be carefully managed to ensure that such compromises are appropriate and not just convenient.

Security practitioners may find that system administrators and development managers share concerns over systems security issues, especially for development systems, and the most effective way to address those security concerns might be in the guise of organizational issues within the development team. For example, developers that use elevated privileges to bypass access control mechanisms during implementation may inadvertently introduce dependencies that are inappropriate to the production environment. These are subtle and costly problems, because they may not be discovered until much later in the QA process, or even after production release, necessitating costly correction efforts. Aligning security concerns with project management issues in this way allows the practitioner to develop a recognition of the security function as supporting important values for the entire application systems development organization.

One way to classify security vulnerabilities is to identify the stage in the systems development life cycle in which the vulnerability is created, as a way to help to focus appropriate attention on correcting vulnerabilities. This also allows defect tracking to assign responsibility if a flaw is discovered in the implementation. For example, input validation should be considered a design requirement, and thus included as a part of the functional specifications implemented in development. QA testing is commonly driven from functional specifications, so the discovery of a vulnerability because input validation is lacking might be a specification failure or a combination of implementation and testing failures. This feedback can be used for process improvement within the development organization, and may often be provided by defect tracking tools. Integrating security concerns into this feedback process is a way to align security efforts with the organizational efforts to continuously improve the development process and results.

Information Criticality Matrix Tool for Security Evaluation

> *Disclaimer:* The National Security Agency has neither reviewed nor approved the following material. It is purely the author's understanding of material obtained from a variety of sources, and his logical extensions of that material.

The InfoSec Assessment Methodology (IAM) developed by the National Security Agency (NSA) provides many useful features. One element of the IAM is particularly promising as a tool for improving application systems security and providing benefits of value to development schedules and results. This section will summarize the IAM, introduce the Information Criticality Matrix used in the IAM, and suggest extensions of that matrix for use in application systems development.

One of the roles for the National Security Agency (NSA) is responsibility for information assurance for information infrastructures critical to U.S. national security interests, through the Information Assurance Directorate (IAD). One NSA/IAD program is the InfoSec Assessment Training and Rating Program (IATRP). According to the NSA Web site (http://www.nsa.gov/isso/iam/index.htm), NSA developed the IATRP, a two-part (training and rating) program, for the benefit of government organizations trying to raise their InfoSec posture in general or specifically trying to comply with the PDD-63 (Presidential Decision Directive) requirement for vulnerability assessments. The IAM is a detailed and systematic way of examining information security programs.

The IAM framework specifically provides for customized extensions to accommodate particular situations having needs that do not fit or that go beyond the standard IAM requirements, with the provision that any modifications not reduce the level of assurance required to be IAM compliant. Much of the IAM codifies accepted practices, describing project organization, standard activities, required elements, and minimum performance expectations for acceptable results. A key feature is the use of a matrix to identify information and systems and structure measurement of the criticality of security for those components. Consistent with common information security practice, the IAM is primarily focused on the needs of operational organizations and their processes rather than their downstream products. This chapter proposes extending the framework and techniques used in the IAM by applying them in coordination with the application systems life cycle.

To summarize the IAM, it provides a framework for projects evaluating information systems security programs. The purpose is to review the information system security posture of a specified operational system to assure that the security program is appropriate for the system requirements. It does not encompass technical vulnerability assessments such as penetration testing or network mapping. There are three phases to the IAM: (1) the preassessment phase, (2) an on-site activities phase, and (3) a postassessment phase. The preassessment phase entails project planning and preparation, including organizational agreements, establishing the scope and bounds of the project, reviewing information about the systems being assessed, reviewing existing security program documentation, and planning and preparing for the on-site activities. The on-site activities gather data to explore and validate information from the preassessment phase and provide initial analysis and feedback to the organization responsible for the systems being assessed. The postassessment phase finalizes the analysis by incorporating results of the on-site activities with information provided during the preassessment phase, and produces a final report.

The IAM specifies a set of baseline categories that are normally reviewed by a compliant evaluation project, unless particular items are specifically excluded by agreement with the assessment client. Any categories that are omitted must be identified and justified, with the requirement that the omission not reduce the level of assurance provided by the assessment. The standard IAM baseline information categories are InfoSec documentation, InfoSec roles and responsibilities, identification and authorization, account management, session controls, external connectivity, telecommunications, auditing, virus protection, contingency planning, maintenance, configuration management, backups, labeling, media sanitization/disposal, physical environment, personnel security, training, and awareness. Additional categories may optionally be added to accommodate specific requirements of the particular systems being evaluated (e.g., encryption), or to provide finer granularity. For example, incident response might be considered part of InfoSec roles and responsibilities and intrusion detection might be included under auditing, or they might be broken out as separate categories.

The purpose of the IAM is to ensure compliance with federal law mandating appropriate security for automated information systems at "SBU" (sensitive but unclassified) level or above. One purpose of the preassessment phase is to "identify subject systems, including system boundaries." This requires addressing both logical and physical systems, along the lines discussed in the section of this chapter discussing application systems technology. Because a logical application system may encompass many physical systems, each of which processes a subset of the system information, it is very useful to have a means of establishing the security requirements for each individual component of the system. The subset may be a particular piece of information or a particular piece of physical equipment. In practice, the security requirements are determined by the nature of the information involved, so the equipment security requirements are derived from the security requirements of the information processed by the particular equipment. The "information criticality matrix" is a tool invented by Mr. Wilbur J. Hildebrand, Jr., NSA's Chief of InfoSec Assessment Services, for use in the IAM to determine the security requirements for particular items of information.

The "information criticality matrix" structures the determination of information security requirements by listing the information elements within the logical system and associated impact values for security attributes. The IAM uses confidentiality, integrity, and availability as the three required standard attributes, and requires that any change to this list be clearly documented. For example, one potential addition might be nonrepudiation, and it would be appropriate to justify the requirement for including it as a separate critical attribute. The result of this matrix provides an initial determination of information security requirements for the overall system, and also values to be used in further refinement of security requirements. The first refinement is the analysis of logical subsystems by selecting the entries for the specific information handled by those subsystems and using them to determine information security requirements for the subsystem. Another refinement is to determine the information security requirement for physical components, based on the information security requirements of all the information (or subsystems) processed by the component. These refinements provide the basis for evaluating whether the information security programs for the affected systems are appropriate for the security requirements of the information contained therein.

Criticality Matrix Use in Application Systems Development

The IAM criticality matrix provides a tool for initially determining information security requirements from a top-down logical systems perspective and then deriving security requirements for the bottom-up systems implementation. This can be productively applied to the development of application systems in several ways. One powerful extension would be to generalize the information resources evaluated using the criticality matrix

to include functional processing components within the logical system design, so that the importance of particular software modules can be determined. This not only serves to focus security requirements, it provides value of great benefit to the systems development project in general, because availability and integrity measure, not just security requirements, but overall importance for the particular functions evaluated. The ability to better measure the importance of functional modules is very beneficial for the systems development project in general because it helps to guide project planning and management in areas such as resource allocation, design attention, testing requirements, defect tracking, etc.

Another use of the criticality matrix to integrate security engineering into the application systems development process would be to focus more attention on addressing technical vulnerabilities (such as buffer overflows) in areas where they would affect critical components vs. areas that are relatively less critical. In some environments, this might help guide management decisions about whether rapid prototyping is an appropriate tool or whether critical components might require additional development attention to ensure appropriate production-quality systems are released for deployment. This provides another opportunity for security practitioners to develop a cooperative relationship as productive contributors generating value important to the application systems development team.

The criticality matrix could even be used to analyze the information security requirements of an application development project over the course of the system development life cycle, and thus to better focus efforts to provide appropriate security for systems used by development projects. Security requirements for systems housing functional specifications and design documents will be different from those of systems used for implementation development, testing, or deployment; and some of those security profiles may be different, depending on the security requirements of the application systems involved. The criticality matrix provides a tool to facilitate consistent evaluation of those security requirements, so that the development projects are neither burdened nor exposed inappropriately.

The criticality matrix can be used in different ways during different stages of the systems development life cycle. During application systems design, it can be used to set security and quality requirements for project features and for project planning and management. During development, it can be used to set appropriate standards for production implementation quality, source management, and feature completion. During QA, it can be used to focus test efforts most effectively, design test strategies, determine the scope and coverage of testing, and track defects according to importance and priority. In operations, it can guide configuration management and deployment planning, and rollout; prioritize bug tracking; and map defects into the systems development life cycle quality and security matrix to provide feedback for process improvement.

Future Directions

This chapter has surveyed some information security considerations pertinent to application systems development, reviewed a number of areas related to application systems and the technical and organizational development environments, and described a novel tool for incorporating security engineering into the application development process. In the course of these topics, several suggestions for future research and development were mentioned. This section reviews some possible directions for future efforts.

There are a number of automated tools in use for managing systems development projects, automating testing, tracking defects, and configuration management and deployment. Incorporation of support for security engineering facilities such as the criticality matrix could be a useful enhancement to such tools. Similarly, intrusion detection systems and management console tools used for systems and network administration of production operations could be enhanced to use the IAM criticality matrix as a factor in prioritizing alerts for all events based on system criticality. It seems especially useful to have configuration management systems provide alerts for discrepancies, and management consoles to report those alerts, with severity settings keyed to the criticality of the subject system, as an adjunct to other IDS monitoring facilities. Undoubtedly, experience will suggest even more and better possibilities in the future.

Resources

1. Available at http://turing.acm.org/technews/articles/2003-5/0312w.html#item8.
2. InfoSec Assessment Methodology, see http://cisse.info/CISSE%20J/2001/RKSm.pdf.

3. Defect costs, see http://www.cebase.org/www/AboutCebase/News/top-10-defects.html and http://www.jrothman.com/Papers/Costtofixdefect.html
4. Systems Development Life Cycle, see http://www.usdoj.gov/jmd/irm/life cycle/table.htm

Acknowledgments

The author would like to express grateful appreciation and thanks to Wilbur J. Hildebrand, Dr. Peter S. Tippett, and Jerry Weinberg.

97

Open Source versus Closed Source

Ed Skoudis, CISSP

> Whoever controls the source, controls the world.
>
> **— Anonymous**

Open source software is remarkably popular right now, and is turning many economic assumptions of the computer software business on their head. It just might have profound security implications, too. We have seen an explosion in open source software being used to run the infrastructure of many corporations and the Internet itself. From the esoteric refuge of high-tech geeks several years ago, open source is becoming mainstream. Chances are, if you use a computer connected to the Internet, you are very reliant on many open source software products, perhaps without realizing it.

In the traditional commercial model of the software industry, a single vendor tightly guards the source code for its products. The customer purchasing a product receives only the executable program, which has been converted from the human-understandable programming language (the source code, which at least some humans can understand) into a form that will directly run on a computer (the executable program itself, which is designed for computers to understand). With only the executable in their hands, customers are totally reliant on the software vendor for fixing bugs and adding new features. Changing the program's operation without access to the source code is distressingly complex, costs large amounts of money, and usually violates the software license agreement imposed by the vendor. Therefore, whoever has the source code for a software tool controls the product and its destiny. For this reason, most mainstream software companies wholeheartedly endorse this so-called "closed source" model — it gives them control.

Rather than have a single company hold the source code, the open source software model distributes the source code far and wide so many people can take advantage of it. Anyone with a legitimate (and often free) license for the product gets both the source code and the executable program. If you want to change the program, you can feel free to alter the source code and generate new executable programs with bug fixes, new features, and modified functionality.

Free versus Open Software Source

It is worth noting that the open source movement itself is not a monolith. It is split into several camps. The two biggest camps are people who support "free" software and those who support commercial software that includes the source code. The free software movement, spearheaded by Richard Stallman, is founded on the idea that users of a software product should have freedom in the use, modification, and redistribution of both the executable and source code. The code is free in the sense that you can do nearly anything you want with it; the user has freedom. This nifty concept of free software is embodied in the Gnu General Public License.*

Open source software, as opposed to free software, may or may not impose additional limitations on the rights of the user. Like free software, the user gets the source code and can customize it to meet various needs.

* Gnu General Public License, http://www.gnu.org/copyleft/gpl.html.

Potentially unlike free software, the user may or may not have limitations in redistributing or selling the source code. Some open source vendors limit users' ability to distribute code, while others do not. Additionally, not all closed source software comes with a price tag. Indeed, there is a bunch of closed source software that vendors and hobbyists write and distribute free of charge. So, there are many categories of free, commercial, open source, and closed source products.

Because this chapter focuses purely on security topics, we are not going to wade into the complex and often baffling waters of the debate between free and open source software. We also will not deal with free closed source software. Instead, we will focus on where the action is — the security of closed source software versus open source software.

Growing in Leaps and Bounds

Open source software is popping up everywhere. Although the software on your home computer might not be open source, whenever you surf the Net you are likely relying on several open source products on the Internet itself. Open source software products are not just toys for the techno-elite. For decades, they have powered major portions of the computer industry. If you doubt the relevance of open-source software, consider the enormous impact of the following open source products:

- *Apache.* This amazing product is the most widely deployed Web server today with over two thirds of Internet-accessible Web sites running on it, easily outpacing its nearest competitor, Microsoft's closed source Internet Information Server (IIS).
- *BIND.* The Berkeley Internet Name Domain server, distributed by the Internet Software Consortium, is the most popular domain name server (DNS) in use today. DNS servers stitch together the infrastructure of the Internet, making it usable by both humans and computers by turning domain names (such as www.counterhack.net) into IP addresses (10.1.1.1), looking up mail server addresses, and performing numerous other critical functions.
- *Sendmail.* This e-mail server and mail transfer agent, maintained by the aptly named Sendmail Consortium, has millions of users. If you receive e-mail on the Internet (and who doesn't?), it more than likely propagated through a Sendmail server at some point.
- *Linux.* This open source operating system has Linus Torvalds as its kernel development leader (and part-time messiah, it sometimes seems). Linux continues to grow in popularity as a server and even a workstation system. If you have not yet used Linux, you should give it a spin. You just might fall in love. Or, Linux could make you long for the comfort of Windows or MacOS. Either way, experience with the ever-more-popular Linux is not a bad move for your career.
- *OpenBSD.* This open source operating system, whose lead designer and developer is Theo DeRaadt, is focused on being highly secure, with a goal of "trying to be the number-one most secure operating system." Until the summer of 2002, their motto was "no remote holes in the default install in nearly six years!" Due to some recent, novel attacks, their new motto is "one remote hole in the default install in nearly six years!" Still, despite the change, that is a breathtaking security record for a complex product like an operating system.
- *GCC and the rest of the Gnu family of tools.* The Gnu C Compiler is one of the most widely used software development tools in the computer industry. Other components of the Gnu Project, sponsored by the Free Software Foundation, make up enormous components of most Linux and OpenBSD distributions. In fact, counting sheer lines of code, the amount of Gnu Project software in standard Linux distributions outweighs the amount of pure Linux code.
- *Snort.* This free, open source intrusion detection system is taking the industry by storm. In addition to this base product, a diverse development community has released accompanying open source products, such as various GUI packages, firewall filtering capabilities, analysis tools, and back-end databases.

And this is only the start of open source software tools that pervade our digital universe. Not only are new open source software projects being added to the ranks of critical software, but the existing open source tools are getting more powerful and more widely used.

Many organizations are beginning to realize the benefits of having direct access to the source code for their operating systems, servers, and applications. If your company wants a custom feature, you can more easily add it to an open source product yourself or contract the work out to a software development firm. If you discover

a bug in an open source solution, you can have your developers rapidly create a fix or work-around for it, instead of having to wait on some pesky vendor to provide a patch. Also, you do not have to compete with other clients of the closed source vendor to get the features and patches you need to run your business.

Not all is completely rosy with open source software, however. I frequently deal with large financial institutions, which have been slow in warming to the charms of open source solutions. Other industries have moved very hesitantly as well, worried that open source just cannot meet their needs as well as traditional (read "closed source commercial") solutions. In my discussions with companies that shun open source tools, they often indicate that their wavering is caused by a variety of factors, including:

- *The view that there is little support available for open source products.* With a closed source commercial solution, you can always beat up on a vendor to fix problems. Although you can purchase support contracts for open source software, some people worry that they will not get the level of support they are accustomed to in the closed source world.
- *Concerns about liability issues and who is responsible for open source software.* Many companies fear that there is no one to sue if open source software goes haywire. Some feel that with a commercial vendor behind a product, there is more liability for their software. However, the onerous licensing agreements from major software manufacturers usually absolve them of all responsibility anyway.
- *Just plain fear of the unknown.* I believe many companies avoid using open source products because they just have not used such tools in the past and the economic model baffles them. I can just picture professional IT people in large companies having nightmares about open source. In their frightening dreams, the big scary boss rolls into the room, waving a stack of papers and yelling: "You chose open source software for what!?!? Don't we have a budget for this sort of thing? Your moronic idea brought down our whole infrastructure. You're FIRED!" As a common refrain in the IT industry admonishes: nobody ever got fired for buying Microsoft solutions.

Which Way Is Better?

As we see, there are some interesting issues associated with the economic model offered by open source software. But we are here to talk about security, not pure economic theory, thank goodness. We will look at the question of whether open source software is inherently more or less secure than the closed source solutions. People on either side of this issue have heated philosophical debates regarding this question. Supporting one side of the issue, there are idealistic open source mavens arguing with religious fervor about their favorite software model to a press corps that thinks this angle is sexy. On the other side, there are the large software development houses, supporting their arguments with significant marketing expenditures. Opinions in this argument are often strong, indicating yet another religious war in the technology industry.

Why This Matters

> Most software sucks.
>
> **— Jim McCarthy**
> *Founder of a software quality training company*

Software quality problems have plagued the information technology industry for decades. With the introduction of higher-density chips, fiber-optic technology, and better hard drives, hardware continues to get more reliable over time. Software, on the other hand, remains stubbornly flawed. Watts Humphrey, a software quality guru and researcher from Carnegie Mellon University, has conducted surveys into the number of errors software developers commonly make when writing code.* Various analyses have revealed that, on average, a typical developer accidentally introduces between 100 and 150 defects per thousand lines of code.

Although many of these errors are simple syntactical problems easily discovered by a compiler, a good deal of the remaining defects often open gaping security holes. In fact, if you think about it, a security vulnerability is really just the very controlled exploitation of a bug to achieve an attacker's specific goal. If the attacker can

* "Bugs or Defects?" Watts S. Humphrey, http://interactive.sei.cmu.edu/news@sei/columns/watts_new/1999/March/watts-mar99.htm#humphrey.

make the program fail in a way that benefits him (by crashing, yielding access, or displaying confidential information), he wins. Estimating very conservatively, if only one in ten of the defects in software has security implications, that leaves between 10 and 15 security defects per thousand lines of code. These numbers just do not look very heartening.

A complex operating system like Microsoft Windows XP has approximately 45 million lines of code, and this gigantic number is growing as new features and patches are released.* Doing the multiplication, there may be 450,000 security defects in Windows XP alone. Ouch! Indeed, the very same day that Windows XP was launched in October 2001, Microsoft released a whopping 18 MB of patches for it. And this is touted by Microsoft personnel as the most secure version of Windows ever.

Do not misunderstand; I love Windows XP. It is far more reliable and easier to use than previous releases of Windows. It is definitely a move in the right direction from these perspectives. However, this is just an illustration of the security problem inherent in large software projects. It is not just a Microsoft issue; the entire software industry is introducing larger, more complex, ultra-feature-rich (and sometimes feature-laden) programs with gobs of security flaws.

A Clear and Present Danger: Why?

> Don't worry, be crappy.
>
> **— Guy Kawasaki**
> *IT pundit, commenting on general software quality*

These concerns about shoddy software have potentially enormous impact. Because our economy relies on software for conducting most business transactions, these software glitches could result in major economic damage. Worse yet, with software-controlled embedded systems running automobiles, aircraft, ships, and other heavy machinery, software flaws could be life threatening. Sadly, software bugs have already been implicated in some fatal injuries. One of the most notable cases occurred in December 2000, when four U.S. Marines were tragically killed in their Osprey helicopter. The tragedy started with a hardware failure — the hydraulic system burst. The software was supposed to handle this issue by running through emergency procedures. However, the emergency software malfunctioned, resulting in the fatal crash.** According to Marine General Martin R. Berndt, "This hydraulic failure alone would not normally have caused an aircraft mishap." Software mistakes are a very serious problem indeed.

Although nowhere near as serious, I was once on an airplane that was delayed at the gate due to technical problems. As we waited, patiently buckled in our seats, the pilot announced over the plane's intercom, "Folks, we're having a technical glitch. It's just a software problem in the engine. But the hardware is just fine, so there's nothing to worry about. We've got to reboot, and then we'll be ready to fly!" This pilot assumed that a hardware problem would be much more serious than a software problem. Although I am no aircraft pilot, I do not agree. Before takeoff, hardware can be thrown away and replaced with a spare part. A software problem is much more difficult to find, understand, and repair. Sometimes, just rebooting does not fix it. Happily, after the reboot, the flight was safe and smooth, transporting this white-knuckled flyer across the continent.

So, why is software so flawed, even as our hardware gets better and better? There are numerous reasons, including:

- *Detailed testing is really, really hard, even with simple programs.* Software testing just is not like any other engineering profession. Suppose you are a civil engineer designing and building a bridge over a river. To test your bridge, you drive a five-ton and then a ten-ton truck on the bridge and it does not fall. It is pretty darn safe to assume that any of the weights in between will not break your structure. Not so with software. If user inputs of five and ten both work properly, an input of seven could make your program careen off in some bizarre fashion, to say nothing of user input such as 3.1415926 or even "%90%EF."

* "Software Firms Need to Plug Security Holes, Critics Contend," Kathryn Balint, *San Diego Union-Tribune,* http://www.signonsandiego.com/news/computing/personaltech/20020128-9999_mz1b28securi.html.

** "Hydraulic, Software Failures Downed Osprey, Marines Say," Gerry J. Gilmore, American Forces Press Service, http://www.defenselink.mil/news/Apr2001/n04092001_200104093.html.

- *Many programs are not built with the mindset of being put into a hostile, networked environment.* Heck, even the protocol that underlies the entire Internet (IP) was not designed for exposure to computer attackers around the world. Instead, the protocol has been patched and security has been retrofitted as we have asked IP to do things it was never planned to do.
- *Software development tools and environments often do not check for simple security errors, forcing the programmer to understand security issues and actively avoid making mistakes.* Many programming languages allow software developers to shoot themselves in the foot and write highly insecure code without any warning from the development tools.
- *Consumers buy features, not quality or security.* Therefore, there is little economic motivation for vendors to do security properly. Security issues easily get moved to the back burner, and will be fixed (or even tested) after the product has shipped.
- *Perhaps the single most important reason software is so full of defects is that we let the software vendors get away with writing garbage code!* Customers do not demand better code. On a related note, as a society we do not hold software vendors liable for the damage caused by their flaws. In the physical world, if an auto manufacturer sold you a car that crashed every 24 hours, you would file suit. In the software world, it is your own darn fault for agreeing to the license and using the vendor's shoddy product.

In an excellent article titled "Why Software Is So Bad," Charles C. Mann explores a few of these issues in far more detail.*

So, software quality definitely matters. What can we do? Adherents of open source software often tout the improved security offered by their favorite software development model. We would be wise to listen to and analyze their arguments carefully. If the open source software model can lower the number of defects even slightly, software will be more secure and we will all be better off. Of course, opponents argue that open source software is actually less secure, offering attackers an ideal environment for exploitation. Both sides regularly release white papers and studies by various gurus to underscore their own biases in the debate. We will explore the arguments on both sides of this issue.

The Case for Open Source Software Being More Secure

> We have confidence (a confidence justified by the track record of Linux, the BSD operating systems, and Apache) that our security holes will be infrequent, the compromises they cause will be relatively minor, and fixes will be rapidly developed and deployed.
>
> **— Eric Raymond****

Many people have the strong belief that open source software is just plain more secure than closed source solutions, but why? The arguments in this camp often start with the intuitive observation that, with more people looking at code, more bugs will be found and fixed. Heck, even the Gartner Group, a business and technology analysis and research organization, has argued that the open source model offers more security. Gartner's opinions on IT trends are quite highly regarded in the industry, with some managers taking every utterance of Gartner as the gospel truth. Gartner weighed in on this debate in May 2002 by stating that

> Gartner believes that open documentation and public review of program interfaces between OSs and applications will lead to stronger security mechanisms over the longer term.***

Now we will zoom in on these arguments to see what is behind them.

---

* "Why Software Is So Bad," Charles C. Mann, *Technology Review Magazine,* August 2002.

** "If You Can't Stand the Heat, 2001," http://newsforge.com/article.pl?sid = 01/10/20/1341225&mode = thread.

*** "Microsoft Sends Mixed Signals about Software Security," John Pescatore, May 12, 2002, http://www3.gartner.com/DisplayDocument?doc_cd = 106790.

More Eyeballs Find More Holes and Fix More Problems

> With many eyeballs, all bugs are shallow.*

With source code available to the general public, many thousands of people around the world can scour that code looking for flaws. These people come from a variety of software disciplines and backgrounds, and can apply their own specific knowledge to finding and solving problems. Security is a distributed systems problem — the careful scrutiny of eyes and brains around the planet is a distributed solution. The benefits even extend beyond people looking at code within their own area of expertise. Because the code is so widely available, an expert in kernel development may periodically check out some device drivers, just to make sure everything looks right. A device driver expert may need to spend some time tweaking the features of a mail server, and might find and correct issues there. The mail server expert may have a need to poke around in the kernel to squeeze out additional performance. While looking over the kernel software, he may just find a problem and offer the solution. If everyone can look for bugs, we can quickly hunt them down to extinction, and we will all be more secure.

Furthermore, beyond the sheer number of eyes looking at the problem, we also need to consider the depth to which problems get explored. Many open source developers are deeply passionate about their projects, going beyond someone who simply puts in a 9-to-5 day slinging code for a living. Most open source developers care intensely about their code, knowing that it will get exposure in front of a worldwide body of their peers. They are, therefore, far more careful than someone desperately trying to meet an arbitrary marketing deadline set by a closed source commercial firm.

Additionally, do not fall into the trap of thinking that all open source developers are just wild-eyed, amateur hobbyists. Several open source projects are funded by major companies, including IBM and Sun Microsystems, who view open source software as an integral component of their future software strategies. Both IBM and Sun have on-staff developers who work exclusively on open source software, focusing their eyes in helping make bugs shallow. With this corporate backing, the entire open source community benefits from independent hobbyists, as well as major corporate dollars.

The "many eyeballs" argument also has a good historical basis. Consider the cryptographic community, where peer review is like breathing — an absolute necessity that you do not even think about not doing. When a new crypto algorithm is created, it is widely published, giving other cryptographers a chance to rip it apart and find flaws. If they find holes in the algorithm, it is either thrown out or improved. If some of the smartest minds on the planet, along with a few cranks who just love math puzzles, and everyone in between, get a chance to beat up on a cryptographic algorithm, the results are much more trustworthy. Without this solid scrutiny, algorithms just cannot be trusted.

Only after this baptism by fire is the algorithm ready for a hostile environment. This same argument applies to software. Public scrutiny of source code helps battle-harden the software, making it ready to face the bad guys. Bruce Schneier, founder and CTO of Counterpane™ Internet Security, sums it up well by asserting:

> In the cryptography world, we consider open source necessary for good security; we have for decades. Public security is always more secure than proprietary security. It's true for cryptographic algorithms, security protocols, and security source code. For us, open source isn't just a business model; it's smart engineering practice.**

Problems Get Fixed Faster

Beyond just finding problems more efficiently, some argue that those problems get fixed faster with open source software. Because everyone has the source, a single organization can create a fix and use it quickly, rather than waiting on a vendor. The developer who fixes a problem can then share that code with everyone else, again showing the power of a distributed approach to developing patches. Additionally, if there is a bug that only impacts your company, you will have difficulty getting the attention of a vendor with thousands or millions

---

*An open source community rallying cry, sometimes called "Linus's Law," originally penned by Eric Raymond in his article, "The Cathedral and the Bazaar."

** Bruce Schneier, Crypto-Gram Newsletter, September 15, 1999, http://www.counterpane.com/crypto-gram-9909.html.

of clients, and your problem may never get resolved. With open source, you can fix the problem yourself, or pay an independent software development firm to fix the problem quickly.

Many open source supporters just have a feeling deep in their gut that problems get fixed faster by the open source community. Ron Ritchey, a security guru from Booz Allen Hamilton, wanted to test this gut feel by subjecting the abstract notion to real-world quantitative study. His formal study focused on three issues: (1) the sheer number of vulnerabilities discovered, (2) the level of risk those holes posed to users of the software, and (3) the time that elapsed between disclosure of the problem and the release of a patch.* This last element is of paramount importance because it represents the duration that users are exposed to attack without any defense. If attackers know about a hole, but the vendor has not provided a fix yet, you are in trouble! The shorter the exposure time, the better, as far as product users are concerned.

To bite off a reasonable chunk of the problem to measure, Ritchey focused on comparing two very popular Web servers: the open source Apache Project and the closed source Internet Information Server (IIS) Web server from Microsoft. Apache is the single most widely used Web server today, with over 66 percent of total market share, according to the regular Netcraft Web survey statistics of August 2002.** IIS is no slouch either, as it holds 25 percent of the market, making it the most widely used commercial Web server. The survey used publicly disclosed vulnerabilities over the period 1996 to 2001, taken from the incredibly useful SecurityFocus.com Web site. Ritchey sorted various reported IIS and Apache vulnerabilities into three risk classes:

1. Vulnerabilities that lead to critical compromise or denial of service
2. Bugs that let an attacker read or write files
3. Vulnerabilities with minor impact

Ritchey's results were startling. Apache had far fewer vulnerabilities in each category. Furthermore, Apache also consistently exposed its users to risk for lower periods of time before a patch was released.

Admittedly, Ritchie's study focused on only two products (Apache and IIS) in one category (Web servers). However, his findings are entirely consistent with an earlier study.*** Additionally, further studies into this interesting phenomenon are being planned as of this writing.

Closed Source Is Not as Closed as You Might Think

> He searches the sources of the rivers and brings hidden things to light.
>
> **— Job 28: 10, 11**

Another argument in favor of open source software is the observation that all source code is really in some way exposed to possible attackers. Getting to the heart of the matter, there really is no such thing as absolutely closed source software. Even when a vendor works diligently to protect source code, hundreds or even thousands of eyes are picking through that code every day. Closed source vendors expose their source code to employees, partners, and possibly to attackers themselves.

First, consider the employees of a closed source software development company. They have widespread access to this supposedly secret source code. A malicious employee could view the code, leak it, and possibly even plant backdoors in it. If you were waging cyber warfare against a large country incredibly dependent on its computer infrastructure, it would make a lot of sense to infiltrate the software companies in your target with bogus employees. Or, if you are not into cyber-war conspiracy theories, consider a single, very gifted computer attacker just hiring on to a large software firm with the intention of getting access to source code. Such employees could steal the source or even alter it with hidden functionality. It would be the ultimate Trojan horse, distributed by the software company itself!

Even in a company with very trustworthy employees, source code is often shared with business partners and joint ventures. Sometimes, to advance research and mindshare in a cost-effective manner, vendors even

---

* "Open Sources versus Closed Sources: Which is More Secure?," presentation by Ron Ritchey, http://www.isse.gmu.edu/~ofut/classes/763/studtalks/Ritchey.pdf.

** Netcraft survey on Web server usage, http://www.netcraft.com/survey.

*** "Does Open Source Improve System Security?" Brian Witten, Carl Landwehr, Michael Caloyannides, *IEEE,* September/October 2001, http://www.computer.org/software/so2001/s5057abs.htm.

share source code with universities, environments not known for their high degree of security or confidentiality. Source code could easily leak and might mysteriously pop up anywhere.

Beyond the insider and partner threats, attackers outside the company may simply steal the source code from the vendors, distributing it freely on the Internet. Microsoft has confirmed that, in October 2000, attackers broke into its corporate network and stole the source code to future versions of Windows.* As of this writing, these attackers have never been apprehended. That is pretty darn spooky, but it goes even further. Publicly available Web sites contain the source code to various versions of Cisco's Internetwork Operating System (IOS), the underlying code that runs a majority of the routers in the world.** Here are two of the most widely used closed source products available today, Windows and IOS, each of which has inadvertently had its source code exposed to malicious attackers.

But it gets even worse for the closed source supporters. An attacker does not even have to steal source code to be able to carefully scrutinize software for bugs. Over the past year, we have seen a revolution in the number and quality of sourceless debugging programs, as shown in Exhibit 97.1. Enormous advances are being made in these tools so that even an attacker with moderate skills can reverse engineer executable programs to find major vulnerabilities, ripe for the picking, without even glancing at the source code. The source code is not needed to tear software apart anymore, as these tools allow an attacker to carefully comb through the executable program's code at a microscopic level to find and exploit defects. Some of the tools allow a user to walk through all of the program's function calls step-by-step to see the flow of the program and determine how to break it. Other tools let the attacker step through the raw machine language code, examining each instruction one by one to find flaws. Some let the attacker manipulate the data structures in the running program to change any parameters, so an attacker can inject faults into the program to see how it bleeds. A few of the tools use a technique called "fuzzing," which allows an attacker to inject random-looking data into a program to see if it can cause it to crash. With all of these tools at an attacker's disposal, keeping the source code secret really does not help mask vulnerabilities.

So, consider the fact that closed source products are exposed to employees, business partners, and sometimes even attackers through outright theft or reverse engineering. You can see that pro-closed source arguments simply amount to security-through-obscurity. According to security-through-obscurity advocates, if we carefully hide our gaping vulnerabilities from our enemies, the bad guys will give up in frustration when they cannot easily find holes. The security community generally considers security-through-obscurity a no-no. Some of the bad guys will be sufficiently motivated to get around our obfuscation, and therefore security-through-obscurity is just not real security at all.

In our debate, if attackers spend enough time trying to steal the source code or even analyzing raw executable program, they will find vulnerabilities. Hiding the source code gives us a false sense of security, when we are really exposed to all kinds of problems. Burying our heads in the sand will not fix this inherent flaw in the security of the closed source software development model.

Fear and Loathing in Redmond (and Elsewhere)

> Author 1: I hear if you play the Windows NT 4.0 CD backwards, you get a Satanic message.
>
> Author 2: That's nothing. If you play it forward, it installs NT 4.0.
>
> **— Jay Dyson**
> *As quoted on Rain Forest Puppy's Web site****

So, if security-through-obscurity is really a bogus argument, one wonders what closed source vendors are really hiding under their sheets. If someone looked through the source code of these products, would there be a cornucopia of problems, just ready to be exploited by eager hordes of hackers?

It would appear to be so. In May 2002, Jim Allchin, Group Vice President for Platforms at Microsoft, testified before a federal court regarding the security of Windows itself. Among some rather fascinating commentary,

---

* "Hackers Burgle Microsoft Source Code," Matthew Broersma, ZDNet UK News, October 27, 2000, http://news.zdnet.co.uk/story/0,,s2082221,00.html.

** I advise you against trolling the Internet for this IOS source code. You will likely be violating some sort of law, and the code could have been laced with malicious backdoors by the attackers who stole it.

***http://www.wiretrip.net/rfp.

EXHIBIT 97.1 A Complete Arsenal of Tools for Finding Security Bugs in Software (which Work with or without Source Code)

Tool Name	Summary	Where to Get It
Free		
APISpy32, by Yariv Kaplan	On Windows systems, this tool monitors all API calls, showing the value of all variables passed along the way	http://www.internals.com/utilities_main.htm
Sharefuzz, by Dave Aitel	On UNIX machines, this program can be used to find holes from local accounts on a machine	http://freshmeat.net/projects/sharefuzz/?topic_id = 43
SPIKE, by Dave Aitel	On UNIX machines, this tool can be used to find flaws in network protocol handling, especially in Web servers and remote procedure calls	http://www.immunitysec.com/spike.html
Heap Debugger, by Anonymous	On Windows systems, this tool lists all memory locations not properly released by an application	http://www.programmersheaven.com/zone24/cat277/4136.htm
Electric Fence, by Bruce Perens	On UNIX machines, this tool can find flaws with the way the system frees memory, which could lead to security exposures	http://perens.com/FreeSoftware/
APIHooks, by EliCZ	On Windows systems, this tool intercepts API calls, allowing an attacker to analyze or even manipulate the flow of data through a program	http://www.anticracking.sk/EliCZ/
Fenris, by Michal Zalewski	Multipurpose tracer, stateful analyzer, and partial decompiler	http://razor.bindview.com/tools/fenris/
Feszer, by Frank Swiderski	This Windows tool is used to analyze problems in string handling functions	http://www.atstake.com/research/tools/index.html
Commercial		
IDA Pro, by Data Rescue	This program is the premier code disassembler tool for both Windows and Linux; extremely powerful and very widely used to find security flaws	http://www.datarescue.com
Cenzic's Hailstorm	This powerful tool allows for finding defects by injecting faults into software	http://www.cenzic.com/
Boundschecker, by Compuware Corporation	On Windows systems, this tool finds errors in C++ programs that could lead to security vulnerabilities	http://www.compuware.com/products/devpartner/bounds/

Allchin claimed that exposing the source code and details of the application programming interfaces (APIs) for Microsoft products would represent a threat to national security. Apparently, there are problems so significant in Windows that mere disclosure of the source would threaten us all. When asked about which areas were of most concern, Allchin mentioned Microsoft's message queuing functionality. This capability supports retrieving user input from the keyboard and mouse and passing that input to applications. Allchin did not want to divulge details, and admitted, "The fact that I even mentioned the message queuing thing bothers me."*

As can be expected, within months of this inadvertent disclosure, the computer underground released some attacks against — you guessed it — message queuing. In his paper, "Shattering Windows," a researcher using the name Foon describes a method for gaining privileged access to a Windows machine by exploiting the message queue.** The paper describes techniques for sending messages to applications running with higher privileges, essentially hijacking the permissions, and using them to accomplish the attacker's own goals. Foon took his inspiration from Allchin's comments, and claims, "Given the quantity of research currently taking place around the world after Mr. Allchin's comments, it is about time the

* "Allchin: Disclosure May Endanger U.S.," Caron Carlson, *eWeek,* May 13, 2002, available at http://www.eweek.com/article2/0,3959,5264,00.asp.

** "Exploiting Design Flaws in the Win32 API for Privilege Escalation … or … Shatter Attacks — How to Break Windows," by Foon, August 2002, http://security.tombom.co.uk/shatter.html.

white hat community saw what is actually possible." Although Microsoft dismisses the originality of Foon's attack, his paper opened up new avenues to a large number of computer attackers.

So, loose lips can sink programs. If a stray comment from an executive of a closed source company can bring lots of attacks, perhaps the underlying philosophy of closed source software is just plain broken. It appears that commercial software vendors' lack of source code scrutiny has allowed them to write sloppy, insecure code. With closed source software, security issues are hidden, while the vendors (and everyone else who relies on the code) keep their fingers crossed that attackers do not stumble across a gaping hole. This state of affairs almost guarantees that knowledgeable and well-funded adversaries can still discover problems.

The open source community simply does not have the "luxury" of hiding its dirty laundry, which forces it to implement security more carefully. If the code is really bad, people will easily see that and not use it.

Even Microsoft Is Starting to Share Source

In March 2002, Microsoft itself released approximately one million lines of code for components for its .NET tools, C# (pronounced, "C sharp") development language and Common Language interface. According to Microsoft, this release was designed especially to support academic and research institutions.* Some have pointed out that, with this release, Microsoft is beginning to grudgingly admit that the open source philosophy has significant benefits. Although there were no hints that Microsoft released the source to help improve security, you had better believe this code has gotten a careful run-through by black hats and white hats around the world looking for security flaws! Also, Microsoft itself probably spent significant time combing through this code, looking for security holes before releasing it on an often-vicious world of software reviewers and malicious attackers.

So, from the open source supporter's point of view, this is definitely a step in the right direction. However, releasing only a part of the source code does not dramatically improve security. Even if Microsoft releases all code associated with security functions, there could still be major holes in other parts of the code. Sure, a developer will be able to comb through a certain set of features of the code released by the vendor. However, using reverse engineering techniques, an attacker may still take over the system by finding and exploiting a gaping hole in the code that the vendor keeps to itself. The flaw could be in a seemingly innocuous piece of the code, perhaps the program's help screens; but even there, a buffer overflow could allow an attacker to completely compromise the system. Without fully releasing source code, vendors cannot receive the security benefits of open source software.

Custom Tailoring at a Fine-Grained Level

Another argument of this camp involves the great deal of customization afforded by wide-open source code distribution. With access to the source code, users can customize their programs, adding or removing features to achieve exactly the mix needed for their businesses. With this flexibility, system hardening is possible at a much more fine-grained level than is possible with closed source solutions. Rather than having everything activated in a default installation, open source users can turn off specific services at will. But it goes farther than that. With access to the source code, open source users can disable specific functions within services, to achieve a much greater level of customization than is possible with closed source solutions. If I do not want to have certain risky functions in my production environment, I can use the source code to strip out those features. Separating the software wheat from the chaff really helps to improve security.

There is also a biological analogy to this argument. With more developers creating customized tweaks of their open source programs, we have many different versions of a given piece of code running on the Internet. Suppose an attacker can compromise one of these versions. However, other versions, which were customized by their users, may not be vulnerable, helping to isolate the problem. In nature, a greater bio-diversity helps to stem the spread of nasty pathogens. A pathogen that can successfully infect some of the population will not be able to harm others because they have enough genetic differences to stop the attacker. Given more differences within a species, pandemic plagues can be more easily thwarted. Given the diversity that open source software allows in deployed systems, this model should help us fight off attackers even better.

* http://www.entmag.com/news/article.asp?EditorialsID=5281.

Economics Matter to Security

A final argument bolstering the security claims of open source supporters is based on the economics of the software industry. Unless you have been living in a cave in recent years, you have probably heard reports about the total cost of ownership for open source software being measurably lower than the costs of commercial software. Of course, if you consider the software itself, many open source products are available in low-cost packages or even for free download. But, even beyond the costs of the code itself, support costs are reportedly lower for open source products. It is believed that the availability of source code, as well as a large and healthy community of developers supporting that code, keeps maintenance costs lower as the overall product is more easily adapted to organizations' changing needs. So, what the heck does this have to do with security?

Well, if you had not heard, money matters. It does not take an Alan Greenspan to realize that if the costs of open source software are lower, then some level of remaining funds can be used to improve security. For organizations developing software, some savings can be channeled into improving the security of the code. For companies that use open source software, the savings can be applied to additional time and energy in securely configuring the software or into the general security budget of the company. Because it has an improved impact on the bottom line, more funds are available for end-user security awareness, computer incident response team activities, and other important security initiatives.

The Case for Closed Source Software Being More Secure

> We can build a better product than Linux.
>
> **— Jim Allchin**
> *Microsoft executive, February 2001*

As the open source cheerleaders put their pom-poms away, we will analyze the opposing viewpoint in detail. Is it possible that closed source solutions have security benefits? We will look at each of the open source arguments, one by one, and see how closed source supporters would respond.

Many Eyes Seem to Miss Many Holes and Some of Those Eyes Are Evil

Is source code really reviewed by lots of eyes, as proponents of open source security sometimes attest? Actually, most often, just a small handful of volunteers look at the code, while the rest of the masses trust these anointed few. Worse yet, the open source philosophy can lead to a false sense of security, as everyone assumes that everyone else is reviewing the code. In a thought-provoking paper on this phenomenon, John Viega asserts

> Currently, however, the benefits open source provides in terms of security are vastly overrated, because there isn't as much high-quality auditing as people believe, and because many security problems are much more difficult to find than people realize.[*]

With their hands on the source code, why do more people *not* pour through it to find flaws? After all, it is in their own self-interest to do so, discovering and solving problems before the bad guys do. There are several reasons code is not reviewed in detail, including:

- Some of the source code is simply ugly, having been glommed together from a bunch of various components over the years. Developers sometimes call this "spaghetti code," and unraveling its messy complexity can be rather like sorting out text written onto individual strands of pasta.
- Even the relatively cleaner code is necessarily very complex, requiring great skill and enormous amounts of time to review and master. It is often better left to professionals paid to do just this task.
- In a related way, a code reviewer must have a holistic view of the entirety of the software, not just one or two piece-parts, to find flaws. Sometimes, a few low-impact vulnerabilities from several widely separated areas of code can be exploited together to create a high-risk vulnerability.
- Code review is a mind-numbingly dull task, perhaps less exciting than watching grass grow on a lazy Sunday afternoon. So, here we have a task that requires great skill, extensive expertise, and super attention to detail, but at the same time, it is just plain boring.

[*] "The Myth of Open Source Security," John Viega, http://www.earthweb.com/article/0,,10455_626641_1,00.html.

- Documentation for open source projects is often quite sparse, a situation only compounded by limited comments in the code itself. For anyone but the original developer, understanding how the code functions at a sufficient level to spot defects is excruciatingly difficult.
- Most of the cream-of-the-crop developers are creating new features and plowing new ground, not looking for holes in the work already completed. Checking for problems is often left to second-tier programmers, if it occurs at all.
- Code gets reviewed unevenly. Certain parts of the code that are sexier, such as widely used features, get lots of attention. Other less interesting parts of the code, which may have major security ramifications, are simply orphaned by developers.
- Many developers might be virtuosos at writing code, but they often do not understand security at a deep enough level to find problems.

So, while the good guys do not review the code, attackers can pour through it and find new flaws quickly. Sure, there are lots of eyes, but many of those eyes belong to highly motivated attackers who want to rip the lungs out of the code and will spend enormous amounts of time finding flaws. They can look through the code at a much deeper level than they can with closed source solutions. All of the highly touted sourceless debuggers do not even the score. With access to the source, attackers can find holes they otherwise would not be able to discover just by poking through the executable.

Consider one very startling flaw in a particular open source product: the Apache chunk handling problem widely publicized in June 2002. This vulnerability was very subtle, involving the way the Web server handles requests when data is grouped in separate chunks for more efficient transmission across the network. By creating these chunks in an unexpected fashion, an attacker can exploit a flaw in the Web server. At first, by carefully analyzing the source code, many security experts believed this flaw would only result in a denial-of-service attack, allowing a bad guy to remotely crash the Web server. Many also believed that only the Windows version of Apache could be successfully exploited. Unfortunately, this analysis just was not accurate.

With the full Apache source code available, a computer underground research group calling itself Gobbles zoomed in on the issue. Within a week of initial disclosure, Gobbles had figured out how to turn this problem into a full-blown remote compromise against a bunch of types of systems. They wrote some code containing their results and unleashed it publicly. Using Gobble's code, an attacker with minimal skills could launch an attack and gain root-level privileges on systems. The day this exploit was released, hundreds of systems around the world were compromised by attackers. Furthermore, it is believed that some attacks over the two months prior to the Gobbles release were based on this fundamental vulnerability. So, even before we knew about this flaw, it is possible that attackers were using it to take over systems. Surely, the open source nature of the code helped Gobbles and perhaps many others to analyze the problem and develop their exploits. All the while, the rest of us blithely relied on the open source model of review to find this exact type of problem.

Furthermore, attackers sometimes have far greater motivation than the defenders in this cat-and-mouse game. If an attacker finds a major security flaw, he or she can use it to exploit systems around the world, potentially for significant financial gain. An attacker could even sell exploited code to the criminal underground, governments, or security companies for big dollars. Even for the less criminally-minded attackers, a fresh vulnerability in a widely used system can generate fame, if not fortune. If you break a big product in a big way, you will get media attention and people will listen to your ranting, when they otherwise would not give you the time of day. Fluffy Bunny,[*] an attacker who broke into the SANS Institute Web site in July 2001, summarized this instant notoriety well. SANS, an organization that offers security training around the world, had its Web page altered to exclaim, "Look Mommy, I'm on SANS!" Fluffy Bunny was seeking attention, and that is just what he got.

Some people think that this problem with open source software is temporary, and now that bugs like the Apache chunk handling problem have been identified, we are all safe. *Au contraire!* Before discovering this problem, Apache was a very mature product, having been initially developed in 1995. These types of flaws impact even mature products. As long as new features are being added, there is a constant supply of new code. New code includes its concomitant brand-spanking-new vulnerabilities. Compounding the problem, with full access to the source, attackers can discover very significant flaws in creaky, old code that has been widely overlooked.

[*] Don't you just love these hacker names? Fluffy Bunny, Gobbles, and even Rain Forest Puppy were certainly inspired when they chose their nifty handles.

Finally, beyond looking for software vulnerabilities, lots of evil eyes with widespread access to source code will build on that code to create even more sinister tools. Consider this: A majority of computer attack tools are developed on open source operating systems, especially Linux and OpenBSD. Because they have the source code to the operating system itself, attackers love to bend the operating system to implement their attacks, with far less work than is required in a closed source solution. The flexibility inherent in open source solutions can be easily hijacked. From creating bizarrely mangled packets to designing difficult-to-detect backdoors, an open source operating system sure helps attackers.

Given this control into the very guts of the operating system itself, the most powerful RootKit tools are found on open source operating systems. RootKits are popular computer attack tools that allow a black hat to maintain backdoor access to a system while hiding from the system administrator. They accomplish this feat by replacing good operating system programs with evil variations that lie about who is logged in, which programs are running, and how the network is being used. Without this critical information, the system administrator cannot detect the attacker's presence. The attackers develop these malicious programs by starting out with the source code for the operating system, and then tweaking it to achieve their goals. Is it any wonder that the best RootKits appear on a system where attackers can use the open source code as a starting point for writing their malicious wares? While RootKits do exist for closed source operating systems, they are invariably less sophisticated than the RootKits in widespread use on open source platforms.

Not All Problems Get Fixed Faster or Very Well

Open source software fans point out the rapidity with which they release patches for security flaws as a virtue of their model. However, this speed often masks the fact that some of these fixes do not adequately eliminate the vulnerability. Instead of highly controlled releases, sometimes the open source community shoots from the hip, getting an inadequate and possibly even damaging patch out very fast. If you send out garbage extremely rapidly, it is still garbage, and you are not doing your users any favors.

Consider the Apache chunk-handling vulnerability discussed previously. The first patch to be released came from the ISS X-Force, a team of high-skilled security professionals. Unfortunately, this patch did not solve the entire problem. Even if you were diligent in assessing this patch, you still would have had a vulnerability that allowed an attacker to take over your system.

Compounding this problem, there is no obvious clearinghouse for vulnerability and patch information in the open source world. Sure, a single company can fix a problem it finds, but who is going to check that solution and distribute it to the entire user base? As shown in Exhibit 97.2, we see a variety of researchers, software firms, consultants, hobbyists, and even riff-raff finding flaws and sometimes releasing patches. These patches may work, or they may cause even bigger problems. Someone could even release a patch, duping users into applying a "fix" that really opens their systems up to attack. Sure, there is usually some core team of developers or foundation standing behind an open source product, but they are often slower to react to

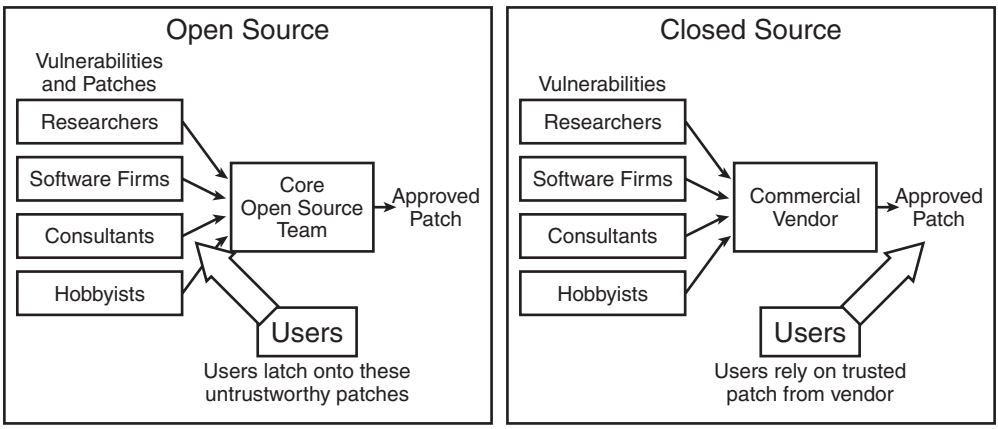


EXHIBIT 97.2 Open source versus closed source patch distribution.

problems. They have to comb through and test the patches discovered by the rest of the world before integrating them into their own code base. This delay eliminates much of the highly vaunted speed of the open source model.

In the closed source software model, on the other hand, the software vendor is clearly the one-stop shop for vulnerability reporting, fix development, and even potential liability if problems do not get fixed. Through its mailing list of customers, the vendor can responsibly disclose the problem, distribute the patch, and even offer various test cases to make sure the patch is functioning properly. Rather than potentially having several competing patches, a single fix by the vendor will efficiently and effectively solve the problem.

Additionally, consider the voluntary nature of many open source contributors. They volunteer their time to support the code, and often are not available on a moment's notice to review a reported problem and release a patch. Unlike these volunteers, closed source commercial software is written by dedicated professionals. Their time often is not sliced as thin as open source volunteers, and they can be dedicated to solving problems. In fact, most large closed source vendors such as Microsoft have teams of individuals waiting for reports of security vulnerabilities. When vulnerabilities are discovered and responsibly reported, the team verifies the problem and interacts with developers to make sure a solution is devised. This centralized approach is much more careful and controlled, two very important characteristics of sound security practices. It also scales better. Although the open source model may allow for solutions to small problems to be fixed by users themselves, the open source model does not necessarily scale particularly well to industrywide software products used by thousands or millions of people.

Reasonable Controls Are in Place Protecting Closed Source

It is indisputable that some closed source software has leaked, including Cisco's core operating system, IOS, and Microsoft Windows. However, despite this fact, we have not seen attackers use this code to create a bunch of new attacks against these platforms. Why? Likely, this abuse has not been seen because these events are so rare, and even when they do occur, the software changes rapidly enough to limit any damage due to exposure of older source code.

Although there have been high-profile cases of source code theft, they are extremely rare. Nearly every script kiddie hacker on the planet, as well as certain highly motivated skilled attackers, has taken a crack at stealing the Windows source. With a product as valuable as the Windows source code to have only been stolen once, and then to have never been released, it appears that the protections used by Microsoft in limiting access to the source code are, for the most part, effective. Certainly, after the October 2001 pilfering, Microsoft beefed up security even more to prevent further problems with the source code leaking out.

Furthermore, the software itself is a moving target. When an attacker steals and distributes an old version of the source code, it does not reveal very many cutting-edge attacks that can be used against recently patched systems. Even if an old version of the source code is stolen, many customers have moved on to newer and better versions. The perpetual upgrade and patch cycle renders this partially exposed source code of very limited use to attackers in undermining the program.

Fear (and Even Loathing) Is Okay if It Is Justified

> Terrorists trying to hack or disrupt U.S. computer networks might find it easier if the federal government attempts to switch to open source, as some groups propose.
>
> **— The Alexis de Tocqueville Institution**
>
> Press release regarding its May 2002 white paper, *"Opening the Open Source Debate."*[*]

In May 2002, the Alexis de Tocqueville Institute, a prestigious Washington, D.C., think tank, released a study on the security issues associated with open source software. This study was certainly a thought-provoking challenge to the assumptions of open source supporters. However, it must be noted that a certain closed source software company provides funding for the Institute.[**] This company, which publicly verified its

[*] White paper available at http://www.adti.net/html_files/defense/opensource_pressrelease_05_30_2002.html.

[**] "Did MS Pay for Open Source Scare?" Michelle Delio, Wired News, June 5, 2002, http://www.wired.com/news/linux/0,1411,52973,00.html.

financial support for the think tank, has a name that is an anagram of the phrase Storm Foci, or if you prefer, Comfort IS.*

However, despite concerns about where the funding comes from, the Institute's white paper is a strong warning for government institutions thinking about moving to open source products. The Alexis de Tocqueville Institute's guiding principles involves studying the spread and perfection of democracy around the world. In this role, the Institute is concerned about both freedom and national security in existing democracies, and views open source as a potential threat to both. According to the Institute's paper, in the aftermath of the September 11, 2001, attacks, terrorists could more easily disrupt the U.S. government and civilian computer networks if they are based on open source software. Because attackers have the source code to work from, they could infiltrate components of critical infrastructure in a far stealthier manner. The paper outlines "how open source might facilitate efforts to disrupt or sabotage electronic commerce, air traffic control or even sensitive surveillance systems." The arguments in the paper go beyond security issues, also citing economic and legal concerns associated with open source software.

Beyond the threats posed by open source solutions, we need to consider the ramifications of distributing source code of currently closed source solutions. If Microsoft purposely placed the source code for Windows on a publicly available Web server and shouted, "Come and get it," would we be safer? Open source proponents frequently brag about Microsoft's assertions that widely releasing the Windows source code would damage national security. Yes, Jim Allchin, a Microsoft executive, did submit testimony to that effect. Yet, pointing this out is not really an argument for exposing the Windows source code, as some open source fans would have it.

If we take Microsoft at its word, and assume that exposing the source for Windows and other products would damage national security, that does not mean we should punish Microsoft and other vendors by pushing them to embrace an open source model. We would be cutting off our nose to spite our face. If such a release would compromise national security, we should not do it. Sometimes, security-through-obscurity is not such a bad thing after all. Keeping the source code out of the hands of the bad guys prevents them from finding problems and developing super nasty tools. Sure, you do not want to rely only on obscurity-for-security. But a dash of obscurity added to an overall security recipe (which includes protection of the source code, secure configuration, and user awareness) can make things even stronger.

Microsoft Is Starting to Share Source Simply to Woo Developers

Some claim that even Microsoft is being dragged to the open source party, as evidenced by its release of a million lines of code for .NET. However, this argument is a red herring, as the release of the .NET source code has nothing at all to do with security. Microsoft is releasing .NET code to woo software developers to adopt Microsoft's framework for developing Web applications. The released source code neither improves nor hurts security in any way.

Too Much Custom Tailoring Can Be Dangerous

Another argument trotted out by open source fans involves the high degree of customization possible with open source solutions. However, this customization is a double-edged sword, and if they are not careful, users could badly cut themselves. If users change the code to shut off individual features without some coherent overall plan, they could inadvertently be weakening security. Similarly, if users start adding features or otherwise tweaking the code, they could very easily inadvertently undermine system security. Even a modification to code that does not have any inherent security functionality could introduce a bug that weakens the overall security of a system. Secure coding is a difficult task, often best left to professionals who understand the code in its entirety.

Going back to the biological analogy of strength through genetic diversity, if there are a bunch of different strong genotypes in a population, a pathogen will be more quickly thwarted. However, some individuals in a diverse population could be swimming in the shallow end of the gene pool. They could certainly have genetic differences, but will likely be far weaker than the original single species. If their differences were developed in a ham-fisted fashion, they could easily be conquered by infection. The same concepts apply to open source software. When users create custom variations, they are quite likely decreasing the security of their system, unless they understand code security at a deep level.

* If you enjoy anagrams, as a lot of computer geeks go, check out the fun, online anagram generator at http://mmm.mbhs.edu/~bconnell/anagrams.html. I use it all the time.

Economics Matter to Security

> Thou source of all my bliss, and all my woe,
>
> That found'st me poor at first, and keep'st me so.
>
> **— Oliver Goldsmith, *The Deserted Village,* 1770**

The economic model of open source software does not necessarily mean that there will be additional funds available for security. Open source software is not like some giant Pez dispenser, shooting out cash that companies will spend on security. The additional support required for the care and feeding of open source software helps to even out its overall cost of ownership, leaving precious little extra money for additional goodies, such as security. Even if there were extra dollars available from open source solutions, these funds would in all likelihood be directed to items other than security.

However, taking the entire IT industry into account, there may not be more money available for security with open source solutions at all. Consider the macroeconomic case over the entire industry. With most open source solutions, there are developers working for a variety of companies around the world, including banks, law firms, and department stores. To realize the benefits of the many eyeballs argument, each of these different entities has to spend some amount of money in helping to secure open source solutions. Adding up all of these costs industrywide raises the overall price of security for open source software.

Now, consider the most common closed source economic model of centralized software development by commercial companies. Experienced, professional programmers work at these commercial software companies, devising patches for software for millions of users. These programmers realize economies of scale in devising security solutions for a wider base of users. Instead of having open source developers around the planet time-sliced, working on security, a smaller centralized group of programmers focused on security could do a better job more cost effectively in the grand scheme of things. By considering the entire universe of software development, the closed source model of patch development and distribution could be more cost effective overall, freeing up funds industrywide to spend on improving security.

Looking at the open source economic model even more closely, there is often little direct financial motivation or legal teeth to getting an open source developer to move in creating a fix for a problem. Suppose a malicious hacker discovers and widely publicizes a vulnerability, but due to your configuration and mix of features, it impacts only your organization and a handful of others. Motivating the open source community to fix it could be difficult, and hiring your own software development firm to address the issue is onerous. Your business is business, not writing software or hiring software development firms. With commercial closed source software, you can rely on and even push a vendor to release fixes. Unlike the typical open source world, if the commercial vendor is hesitant, you can threaten to stop using the products or even send nasty letters from your lawyers explaining how the vendor is increasing your risk. The vendor may be liable for negligence in not addressing your issue. With commercial closed source solutions, you have recourse to get action from the vendor, which you often do not have in the open source space.

Sorting It All Out

> WIRED: Linux fans believe their OS is secure because the code is reviewed by developers worldwide. Do more eyes mean more security?
>
> DE RAADT: I've been disagreeing with this point of view since the first time I heard it. The "more eyes" statement is like saying, "When more people walk the streets, there will be less crime." That only works when the crimes are obvious, like muggings, and when those people are cops. The little things get glossed over by the large number of eyes.
>
> **Theo De Raadt**
> *Founder and lead developer of the OpenBSD Operating System*[*]

So, where does my opinion fall in this high-stakes computer poker game, where powerful forces on either side vie for supremacy? On the one hand, we have the caricature of the entrenched, rich, and often imperial

[*] *Wired* interview, September 2002.

commercial closed source software companies, with enough additional money to fund think tanks. On the other side, we have the image of the ragtag open source zealots, with focus and drive rarely seen in the software industry. Although neither image is completely fair, these stereotypes often lead people to reach drastic conclusions about whom to trust in solving security issues. We need to look beyond the stereotypes while considering the arguments discussed throughout this chapter.

Carefully weighing the arguments, in my opinion, for all practical purposes, it is a wash, a dead tie. Of course, stating that opinion means that adherents of both sides of this issue will disagree with me. Such is life, I suppose. As is evidenced by the numerous notes to this chapter, both closed source and open source supporters are feverishly trying to drag security into their fight. I find it fascinating that both sides have recently zoomed in on security topics to help them win the debate in favor of their own ideal software model.

However, security is almost always independent of whether a product is closed source or open source. Some open source software is very vulnerable, and some has exemplary security. Some closed source solutions completely stink, while others are rock solid. What really matters here is the quality of the software development process and the conscientiousness of development team members. The old-fashioned issues of solid software design, careful implementation, and comprehensive testing are what matters, not whether the source code is available to the user base. Additionally, independent of the software development economic model, carefully configuring and maintaining the system are incredibly important to keeping it secure.

The Tie Will Remain for Quite a While

> The constant demand for novelty means that software is always in the bleeding-edge phase, when products are inherently less reliable.
>
> **— Charles C. Mann**[*]

This opinion of balance between the two sides is further bolstered by the current state of maturity of many widely used software products. Vendors (both open and closed source) are continuously releasing new and complex features every single day for operating systems, servers, browsers, and other tools. With this constant introduction of new features, we get a continual release of fresh security bugs in both open and closed source solutions. The many eyeballs of the open source community have a lot to look over, as do the closed source development teams. In this environment, security will continue to be a challenge, regardless of whether we use open or closed source products. We should continue to listen to the arguments on both sides of the issue. But keep in mind that they often cancel each other out under the huge load of new vulnerabilities discovered in tools released through each model, as well as the poor administration and maintenance found on many systems today.

[*] "Why Software Is So Bad," *Technology Review Magazine,* August 2002.

PeopleSoft Security

Satnam Purewal

SECURITY WITHIN AN ORGANIZATION'S INFORMATION SYSTEMS ENVIRONMENT IS GUIDED BY THE BUSINESS AND DRIVEN BY AVAILABLE TECHNOLOGY ENABLERS. Business processes, functional responsibilities, and user requirements drive security within an application. This chapter highlights security issues to consider in a PeopleSoft 7.5 client/server environment, including the network, operating system, database, and application components.

Within the PeopleSoft client/server environment, there are several layers of security that should be implemented to control logical access to PeopleSoft applications and data: network, operating system, database, and PeopleSoft application security. Network, operating system, and database security depend on the hardware and software selected for the environment (Windows NT, UNIX, and Sybase, respectively). User access to PeopleSoft functions is controlled within the PeopleSoft application.

1. Network security controls:
   a. who can log on to the network
   b. when they can log on (via restricted logon times)
   c. what files they can access (via file rights such as execute-only, read-only, read/write, no access, etc.)
2. Operating system security controls:
   a. who can log on to the operating system
   b. what commands can be issued
   c. what network services are available (controlled at the operating system level)
   d. what files/directories a user can access
   e. the level of access (read, write, delete)
3. Database security controls:
   a. who can log on to a database
   b. which tables or views users can access
   c. the commands users can execute to modify the data or the database
   d. who can perform database administration activities

4. PeopleSoft online security controls:
   a. who can sign-on to PeopleSoft (via operator IDs and passwords)
   b. when they can sign-on (via operator sign-on times)
   c. the panels users can access and the functions they can perform
   d. the processes users can run
   e. the data they can query/update

NETWORK SECURITY

The main function of network security is to control access to the network and its shared resources. It serves as the first line of defense against unauthorized access to the PeopleSoft application.

At the network security layer, it is important to implement login controls. PeopleSoft 7.5 delivers limited authentication controls. If third-party tools are not going to be used to enhance the PeopleSoft authentication process, then it is essential that the controls implemented on this layer are robust.

The network servers typically store critical application data like client-executable programs and management reports. PeopleSoft file server directories should be set up as read-only for only those individuals accessing the PeopleSoft application (i.e., access should not be read-only for everyone on the network). If executables are not protected, unauthorized users could inadvertently execute programs that result in a denial-of-service. For this reason, critical applications used to move data should be protected in a separate directory. Furthermore, the PeopleSoft directories containing sensitive report definitions should be protected by only granting read access to users who require access.

DATABASE MANAGEMENT SYSTEM SECURITY

The database management system contains all PeopleSoft data and object definitions. It is the repository where organizational information resides and is the source for reporting. Direct access to the database circumvents PeopleSoft application security and exposes important and confidential information.

All databases compatible with the PeopleSoft applications have their own security system. This security system is essential for ensuring the integrity and accuracy of the data when direct access to the database is granted.

To reduce the risk of unauthorized direct access to the database, the PeopleSoft access ID and password must be secured, and direct access to the database should be limited to the database administrators (DBAs).

The access ID represents the account that the application uses to connect to the underlying database in order to access PeopleSoft tables. For

the access ID to update data in tables, the ID must have read/write access to all PeopleSoft tables (otherwise, each individual operator would have to be granted access to each individual table). To better understand the risk posed by the access ID, it helps to have an understanding of the PeopleSoft sign-on (or logon) process:

1. When PeopleSoft is launched on the user workstation, the application prompts for an operator ID and password. The ID and password input by the operator is passed to the database (or application server in three-tier environments).
2. The operator ID and password are validated against the PSOPRDEFN security table. If both are correct, the access ID and password are passed back to the workstation.
3. PeopleSoft disconnects from the DBMS and reconnects using the access ID and password. This gives PeopleSoft read/write access to all tables in the database.

The application has full access to all PeopleSoft tables, but the access granted to the individual operator is restricted by PeopleSoft application security (menu, process, query, object, and row-level security). Users with knowledge of the access ID and password could log on (e.g., via an ODBC connection) directly to the database, circumventing application security. The user would then have full access privileges to all tables and data, including the ability to drop or modify tables.

To mitigate this risk, the following guidelines related to the access ID and password should be followed:

- Procedures should be implemented for regularly changing the access ID password (e.g., every 30 days). At a minimum, the password must be changed anytime someone with knowledge of it leaves the organization.
- Ownership of the access ID and password should be assigned, preferably to a DBA. This person would be responsible for ensuring that the password is changed on a regular interval, and for selecting strong passwords. Only this person and a backup should know the password. However, the ID should never be used by the person to log on to the database.
- Each database instance should have its own unique access ID password. This reduces the risk that a compromised password could be used to gain unauthorized access to all instances.
- The access ID and password should not be hard-coded in cleartext into production scripts and programs. If a batch program requires it, store the ID and password in an encrypted file on the operating system and "point" to the file in the program.

- Other than DBAs and technical support personnel, no one should have or need a database ID and direct connectivity to the database (e.g., SQL tools).

OPERATING SYSTEM SECURITY

The operating system needs to be secured to prevent unauthorized changes to source, executable, and configuration files. PeopleSoft and database application files and instances reside on the operating system. Thus, it is critical that the operating system environment be secure to prevent unauthorized changes to source, executable, and configuration files.

PEOPLESOFT APPLICATION SECURITY

To understand PeopleSoft security, it is first essential to understand how users access PeopleSoft. To access the system, an operator ID is needed. The system will determine the level of access for which the user is authorized and allow the appropriate navigation to the panels.

Many organizations have users with similar access requirements. In these situations, an "operator class" can be created to facilitate the administration of similar access to multiple users. It is possible to assign multiple operator classes to users. When multiple operator classes are used, PeopleSoft determines the level of access in different ways for each component. The method of determining access is described below for each layer when there are multiple operator classes.

PeopleSoft controls access to the different layers of the application using operator classes and IDs. The term "operator profile" is used to refer, in general, to both operator IDs and classes. Operator profiles are used to control access to the different layers, which can be compared to an onion. Exhibit 12-1 shows these layers: Sign-on security, panel security, query security, row-level security, object security, field security, and process security. The outer layers (i.e., sign-on security and panel security) define broader access controls. Moving toward the center, security becomes defined at a more granular level.

The layers in Exhibit 12-1:

- Sign-on security provides the ability to set up individual operator IDs for all users, as well as the ability to control when these users can access the system.
- Panel security provides the ability to grant access to only the functions the user requires within the application.
- Query security controls the tables and data users can access when running queries.
- Row-level security defines the data that users can access through the panels they have been assigned.

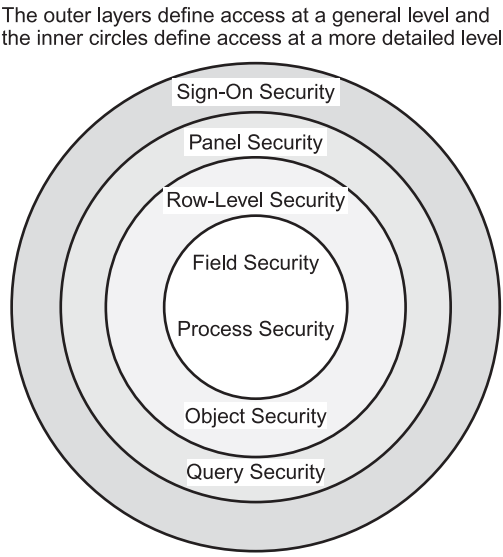


Exhibit 12-1. PeopleSoft security onion.

- Object security defines the objects that users can access through the tools authorized through panel security.
- Field security is the ability to restrict access to certain fields within a panel assigned to a user.
- Process security is used to restrict the ability to run jobs from the PeopleSoft application.

Sign-on Security

PeopleSoft sign-on security consists of assigning operator IDs and passwords for the purpose of user logon. An operator ID and the associated password can be one to eight characters in length. However, the delivered sign-on security does not provide much control for accessing the PeopleSoft application.

PeopleSoft (version 7.5 and earlier) modules are delivered with limited sign-on security capabilities. The standard features available in many applications are not available within PeopleSoft. For example, there is no way to limit the number of simultaneous sessions a user can initiate with an operator ID. There also are no controls over the types of passwords that can be chosen. For example, users can choose one-character passwords or they can set the password equal to their operator ID. Users with passwords equal to the operator ID do not have to enter passwords at logon. If these users are observed during the sign-on process, it is easy to determine their passwords.

Many organizations have help desks for the purpose of troubleshooting common problems. With PeopleSoft, password maintenance cannot be decentralized to the help desk without also granting the ability to maintain operator IDs. This means that the help desk would also have the ability to change a user's access as well as the password. Furthermore, it's not possible to force users to reset passwords during the initial sign-on or after a password reset by the security administrator.

There are no intrusion detection controls that make it possible to suspend operator IDs after specified violation thresholds are reached. Potentially, intruders using the brute-force method to enter the system will go undetected unless they are caught trying to gain access while at the workstation.

Organizations requiring more robust authentication controls should review third-party tools. Alternatively, PeopleSoft plans to introduce password management features in version 8.0.

Sign-on Times. A user's session times are controlled through the operator ID or the operator class(es). In either case, the default sign-on times are 24 hours a day and 7 days a week. If users will not be using the system on the weekend or in the evening, it is best to limit access to the known work hours.

If multiple operator classes are assigned to operator IDs, attention must be given to the sign-times. The user's start time will be the earliest time found in the list of assigned operator classes. Similarly, the user's end time will be the latest time found in the list of assigned operator classes.

Delivered IDs. PeopleSoft is delivered with operator IDs with the passwords set equal to the operator ID. These operator IDs should be deleted because they usually have full access to business panels and developer tools. If an organization wishes to keep the delivered operator IDs, the password should be changed immediately for each operator ID.

Delivered Operator Classes. PeopleSoft-delivered operator classes also have full access to a large number of functional and development menus and panels. For example, most of these operator classes have the ability to maintain panels and create new panels. These operator classes also have the ability to maintain security.

These classes should be deleted in order to prevent them from being assigned accidentally to users. This will prevent users from getting these operator classes assigned to their profile in error.

Panel Security

There are two ways to grant access to panels. The first way is to assign menus and panels directly to the operator ID. The second way is to assign menus/panels to an operator class and then assign the operator class to

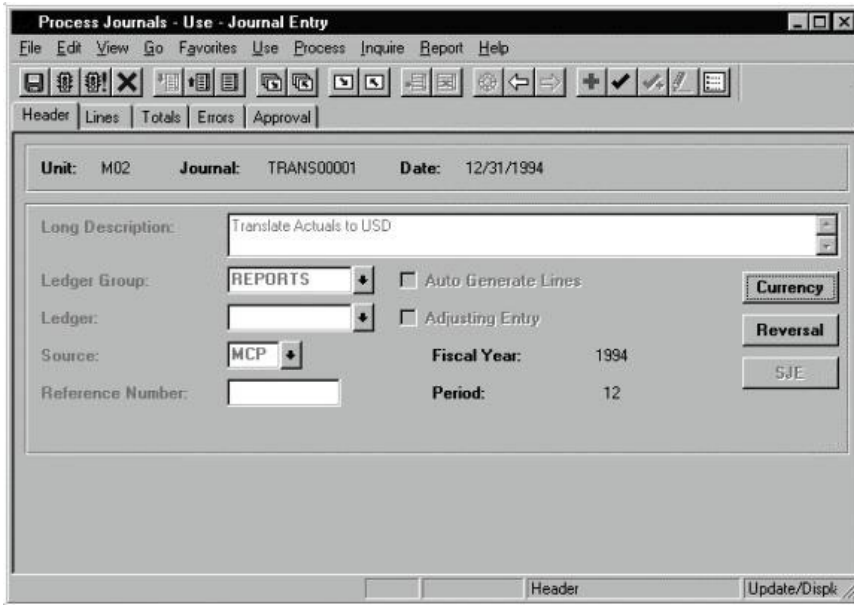


Exhibit 12-2. The PeopleSoft journal entry panel.

the operator ID. When multiple operator classes are assigned to a user, the menus granted to a user are determined by taking a union of all the menus and panels assigned from the list of operator classes assigned to the user. If a panel exists in more than one of the user's operator classes with different levels of access, the user is granted the greater access. This means if in one operator class the user has read-only access and in the other the user has update access, the user is granted update access. This capability allows user profiles to be built like building blocks. Operator classes should be created that reflect functional access. Operator classes should then be assigned according to the access the user needs.

Panel security is essentially column security. It controls access to the columns of data in the PeopleSoft tables. This is best described with an example. The PeopleSoft Journal Entry panel (see Exhibit 12-2) has many fields, including Unit, Journal, Date, Ledger, Long Description, Ledger Group, Ledger, Source, Reference Number, and Auto Generate Lines.

Exhibit 12-3 shows a subset of the columns in the table JRNL_HEADER. This table is accessible from the panel **Process Journals – Use – Journal Entry Headers** panel. The fields in this panel are only accessible by the user if they are displayed on the panel to which the user has access.

When access is granted to a panel, it is also necessary to assign *actions* that a user can perform through the panel. Exhibit 12-4 shows the actions that are

Exhibit 12-3. A subset of the columns in the table JRNL_HEADER.

Unit	Journal	Date	Long Descr	Ledger Grp	Ledger	Source	Ref No	Auto Gen
M02	TRANS0001	1994-12-31	Translate Actuals to USD	REPORTS		MCP		N
M02	TRANS0001	1995-12-31	Translate Actuals to USD	REPORTS		MCP		N
M02	TRANS0001	1996-01-01	Translate Actuals to USD	REPORTS		MCP		N
M04	0000005185	1995-12-27	Adjusting entries for unexpected Production Scrap - not to be repeated.	ACTUALS		ADJ		N
M04	0000005197	1998-03-13	Inventory Transactions	ACTUALS		INV	INV100	N
M04	0000005259	1998-03-19	Inventory Transactions	ACTUALS		INV	INV100	N
M04	0000005271	1998-01-31		BUDGETS		CFO		N
M04	0000005272	1998-01-01	Budget Journals	BUDGETS		CFO		N

Exhibit 12-4. Common actions in panels.

Action	Capability
Add	Ability to insert a new row
Update/Display	Ability to access present and future data
Update/Display All	Ability to access present, future, and historical data; updates to historical data are not permitted
Correction	Ability to access present, future, and historical data; updates to historical data are permitted

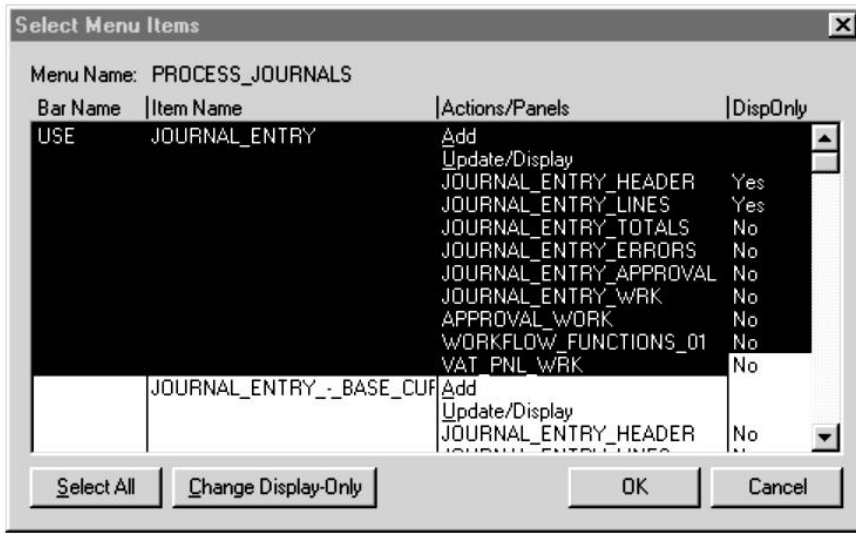


Exhibit 12-5. Assigning read-only access.

common to most panels. This table only shows a subset of all the actions that are available. Furthermore, not all of these actions are available on all panels.

From a security standpoint, correction access should be limited to select individuals in an organization because users with this authority have the ability to change historical information without maintaining an audit trail. As a result, the ability to change historical information could create questions about the integrity of the data. Correction should be used sparingly and only granted in the event that an appropriate process is established to record changes that are performed.

The naming convention of two of the actions (Update/Display, Update/Display All) is somewhat misleading. If a user is granted access to one or both of these actions, the user does not necessarily have update access. Update access also depends on the "Display Only" attribute associated with each panel. When a panel is assigned to an operator ID or operator class, the default access is update. If the user is to have read-only access to a panel, then this attribute must be set to "Y" for yes (see Exhibit 12-5 for an example). This diagram shows that the user has been assigned read-only access to the panels "JOURNAL_ENTRY_HEADER" and "JOURNAL_ENTRY_LINES." For the other highlighted panels, the user has been granted update capabilities.

The panels that fall under the menu group PeopleTools provide powerful authority (see Exhibit 12-6 for a list of PeopleTools menu items). These panels should only be granted to users who have a specific need in the production environment.

Exhibit 12-6. PeopleTools menu items.

APPLICATION DESIGNER
SECURITY ADMINISTRATOR
OBJECT SECURITY
APPLICATION REVIEWER
UTILITIES
IMPORT MANAGER
PROCESS SCHEDULER
EDI MANAGER
nVISION
REPORT BOOKS
TREE MANAGER
QUERY
APPLICATION ENGINE
MASS CHANGE
WORKFLOW ADMINISTRATOR
PROCESS MONITOR
TRANSLATE
CUBE MANAGER

Query Security

Users who are granted access to the **Query** tool will not have the capability to run any queries unless they are granted access to PeopleSoft tables. This is done by adding *Access Groups* to the user's operator ID or one of the operator classes in the user's profile. Access Groups are a way of grouping related tables for the purposes of granting query access.

Configuring query security is a three-step process:

1. Grant access to the **Query** tool.
2. Determine which tables a user can query against and assign **Access Groups**.
3. Set up the Query Profile.

Sensitive organizational and employee data is stored within the PeopleSoft application and can be viewed using the **Query** tool. The challenge in setting up query security is consistency. Many times, organizations will spend a great deal of effort restricting access to panels and then grant access to view all tables through query. This amounts to possible unauthorized access to an organization's information. To restrict access in query to the data accessible through the panels may not be possible using the PeopleSoft delivered access groups. It may be necessary to define new access groups to enable querying against only the tables a user has been authorized to view. Setting up customized access groups will facilitate an organization's objective to ensure consistency when authorizing access.

The **Query Profile** helps define the types of queries a user can run and whether the user can create queries. Exhibit 12-7 displays an example of a profile. Access to the Query tool grants users the ability to view information that resides within the PeopleSoft database tables. By allowing users to create ad hoc queries can require high levels of system resources in order to run complex queries. The Query Profile should be configured to reduce the risk of overly complex queries from being created without being tuned by the database administrators.

The Query Profile has several options to configure. In the **PS/Query Use** box, there are three options. If a user is not a trained query user, then access should be limited to *Only Allowed to run Queries.* Only the more experienced users should be given the authority to create queries. This will reduce the likelihood that resource intensive queries are executed.

Row-level Security

Panel security controls access to the tables and columns of data within the tables but a user will be able to access all data within the columns of the tables on the panel. To restrict user access to data on a panel, row-level security should be established. Access is granted to data using control fields. For example, in Exhibit 12-8 the control field is "Unit" (or Business Unit). If a user is assigned to only the M02 business unit, that user would only be able to see the first four lines of data.

Row-level security is implemented differently in HRMS and Financials.

Human Resource Management System (HRMS) Row-level Security. In HRMS, the modules are delivered with row-level security activated. The delivered row-level security is based on a Department Security Tree and is hierarchical (see Exhibit 12-9). In this example, if a user is granted access to ABC manufacturing department, then the user would have access to the ABC manufacturing department and all of the child nodes. If access is granted to the department Office of the Director Mfg, then the user would have access to the Office of the Director Mfg as well as Corporate Sales, Corporate Marketing, Corporate Admin/Finance, and Customer Services. It is also possible to grant access to the department Office of the Direct Mfg. and then deny access to a lower level department such as Corporate Marketing.

It is important to remember that the organizational tree and the security tree in HRMS need not be the same. In fact, they should not be the same. The organizational tree should reflect the organization today. The security tree will have historical nodes that may have been phased out. It is important to keep these trees in order to grant access to the associated data.

Financials Row-level Security. In the Financials application, row-level security is not configured in the modules when it is delivered. If row-level

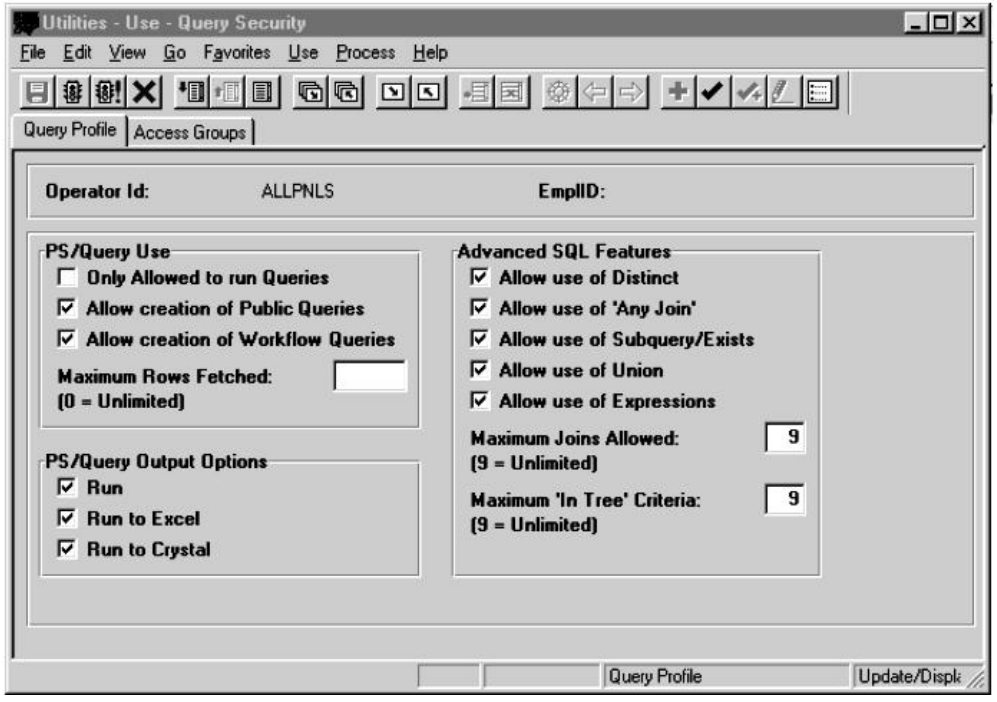


Exhibit 12-7. Query profile.

Exhibit 12-8. Row-level security.

Unit	Journal	Date	Ledger	Unit	Currency	Foreign Curr.	Debits	Credits
M02	AP00005168	1995-12-31	ACTUALS	M02	CAD	CAD	50000.00	50000.00
M02	BI00005216	1998-03-16	ACTUALS	M02	CAD	USD	10149.30	10149.30
M02	BI00005258	1998-03-18	ACTUALS	M02	CAD	USD	20298.60	20298.60
M02	TRANS00001	1995-12-31	REPORTS	M02	USD	USD	3470257761.27	3470257761.27
M04	0000005185	1995-12-27	ACTUALS	M04	USD	CAD	60362.91	60362.91
M04	0000005185	1995-12-27	ACTUALS	M04	USD	USD	6345.00	6345.00
M04	0000005197	1998-03-13	ACTUALS	M04	USD	USD	525145.27	525145.27
M04	0000005271	1998-01-31	BUDGETS	M04	USD	CAD	69075.08	69075.08

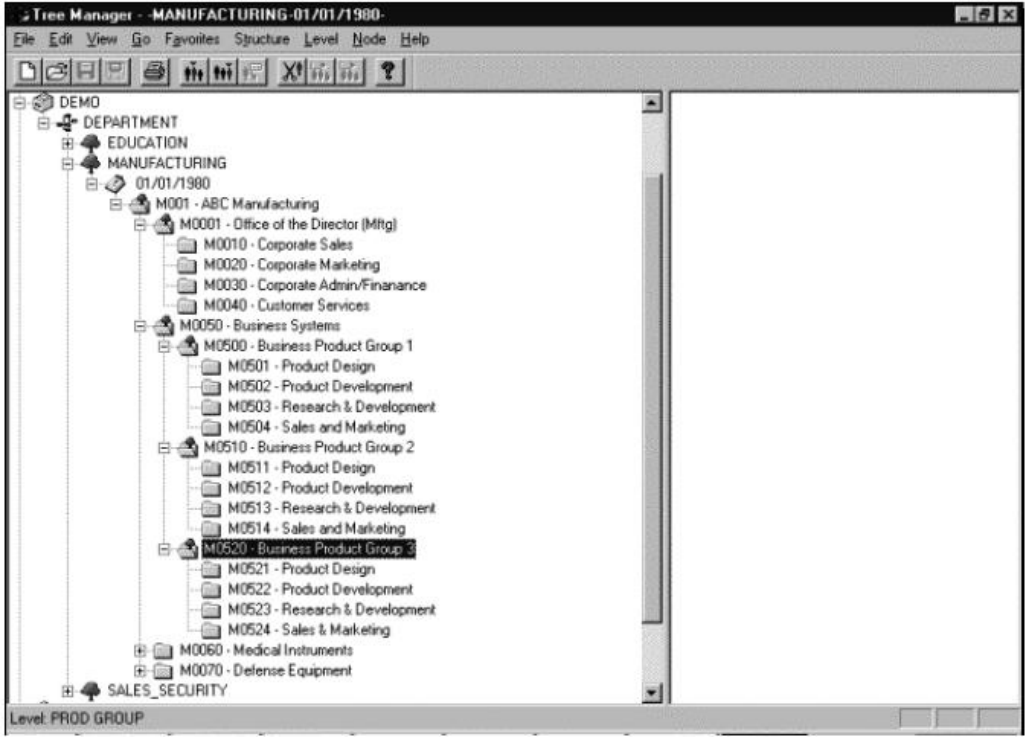


Exhibit 12-9. Department security tree.

security is desired, then it is necessary to first determine if row-level security will be implemented at the operator ID or operator class level. Next, it is necessary to determine the control fields that will be used to implement row-level security. The fields available for row-level security depend on the modules being implemented. Exhibit 12-10 shows which module the options are available in.

Exhibit 12-10. Modules of available options.

Field	Module
Business Unit	General Ledger
SetID	General Ledger
Ledger	General Ledger
Book	Asset Management
Project	Projects
Analysis Group	Projects
Pay Cycle	Accounts Payable

Object Security

In PeopleSoft, an object is defined as a menu, a panel, or a tree. For a complete list of objects, see Exhibit 12-11. By default, all objects are accessible to users with access to the appropriate tools. This should not always be the case. For example, it is not desirable for the security administrator to update the organization tree, nor is it appropriate for an HR supervisor to update the department security tree. This issue is resolved through object groups. Object groups are groups of objects with similar security privileges. Once an object is assigned to an object group, it is no longer accessible unless the object group is assigned to the user.

Exhibit 12-11. PeopleSoft objects.

Import Definitions (I)
Menu Definitions (M)
Panel Definitions (P)
Panel Group Definitions (G)
Record Definitions (R)
Trees (E)
Tree Structure Definitions (S)
Projects (J)
Translate Tables (X)
Query Definitions
Business Process Maps (U)
Business Processes (B)

In production, there should not be any access to development-type tools. For this reason, the usage of object security is limited in production. It is mainly used to protect trees. When users are granted access to the Tree Manager, the users have access to all the available trees. In production HRMS, this would mean access to the organization tree, the department security tree, and query security trees. In Financials, this means access to the query security trees and the reporting trees. To resolve this issue, object security is used to ensure that the users with access to Tree Manager are only able to view/update trees that are their responsibility.

Field Security

The PeopleSoft application is delivered with a standard set of menus and panels that provides the functionality required for users to perform their job functions. In delivering a standard set of menus and panels, there are occasions in which the access to data granted on a panel does not coincide with security requirements. For this reason, field-level security may need to be implemented to provide the appropriate level of security for the organization.

Field security can be implemented in two ways; either way, it is a customization that will affect future upgrades. The first option is to implement field security by attaching PeopleCode to the field at the table or panel level. This is complicated and not easy to track. Operator IDs or operator classes are hard-coded into the code. To maintain security on a long-term basis, the security administrator would require assistance from the developers.

The other option is to duplicate a panel, remove the sensitive field from the new panel, and secure access through panel security to these panels. This is the preferred method because it allows the security administrator control over which users have access to the field and it is also easier to track for future upgrades.

Process Security

For users to run jobs, it is necessary for them to have access to the panel from which the job can be executed. It is also necessary for the users to have the process group that contains the job assigned to their profile.

To simplify security administration, it is recommended that users be granted access to all process groups and access be maintained through panel security. This is only possible if the menus/panels do not contain jobs with varying levels of sensitivity. If there are multiple jobs on a panel and users do not require access to all jobs, then access can be granted to the panel and to the process group that gives access to only the jobs required.

SUMMARY

Within the PeopleSoft client/server environment, there are four main layers of security that should be implemented to control logical access to PeopleSoft applications: network, operating system, database, and application security. Network security is essential to control access to the network and the PeopleSoft applications and reports. Operating system security will control access to the operating system as well as shared services. Database security will control access to the database and the data within the database. Each layer serves a purpose and ignoring the layer could introduce unnecessary risks.

PeopleSoft application security has many layers. An organization can build security to the level of granularity required to meet corporate requirements. Sign-on security and panel security are essential for basic access. Without these layers, users are not able to access the system. Query security needs to be implemented in a manner that is consistent with the panel security. Users should not be able to view data through query that they cannot view through their authorized panels. The other component can be configured to the extent that is necessary to meet the organization's security policies.

Individuals responsible for implementing security need to first understand the organization's risk and the security requirements before they embark on designing PeopleSoft security. It is complex, but with planning it can be implemented effectively.

World Wide Web Application Security

Sean Scanlon

DESIGNING, IMPLEMENTING, AND ADMINISTERING APPLICATION SECURITY ARCHITECTURES THAT ADDRESS AND RESOLVE USER IDENTIFICATION, AUTHENTICATION, AND DATA ACCESS CONTROLS, HAVE BECOME INCREASINGLY CHALLENGING AS TECHNOLOGIES TRANSITION FROM A MAINFRAME ARCHITECTURE, TO THE MULTIPLE-TIER CLIENT/SERVER MODELS, TO THE NEWEST WORLD WIDE WEB-BASED APPLICATION CONFIGURATIONS. Within the mainframe environment, software access control utilities are typically controlled by one or more security officers, who add, change, and delete rules to accommodate the organization's policy compliance. Within the n-tier client/server architecture, security officers or business application administrators typically share the responsibility for any number of mechanisms, to ensure the implementation and maintenance of controls. In the Web application environment, however, the *application user* is introduced as a co-owner of the administration process.

This chapter provides the reader with an appreciation for the intricacies of designing, implementing, and administering security and controls within Web applications, utilizing a commercial third-party package. The manuscript reflects a real-life scenario, whereby a company with the need to do E-business on the Web goes through an exercise to determine the cost/benefit and feasibility of building in security versus adding it on, including all of the considerations and decisions made along the way to implementation.

HISTORY OF WEB APPLICATIONS: THE NEED FOR CONTROLS

During the last decade or so, companies spent a great deal of time and effort building critical business applications utilizing client/server architectures. These applications were usually distributed to a set of controlled, internal users, usually accessed through internal company resources or dedicated, secured remote access solutions. Because of the limited set of users and respective privileges, security was built into the applications or provided by third-party utilities that were integrated with the application. Because of the centralized and limited nature of these applications,

Exhibit 13-1. Considerations for large Web-based application development.

- Authenticating and securing multiple applications, sometimes numbering in the hundreds
- Securing access to applications that access multiple systems, including legacy databases and applications
- Providing personalized Web content to users
- Providing single sign-on access to users accessing multiple applications, enhancing the user experience
- Supporting hundreds, thousands, and even millions of users
- Minimizing the burden on central IT staffs and facilitating administration of user accounts and privileges
- Allowing new customers to securely sign-up quickly and easily without requiring phone calls
- Scalability to support millions of users and transactions and the ability to grow to support unforeseen demand
- Flexibility to support new technologies while leveraging existing resources like legacy applications, directory servers, and other forms of user identification
- Integration with existing security solutions and other Internet security components

management of these solutions was handled by application administrators or a central IT security organization.

Now fast-forward to current trends, where the Web and Internet technologies are quickly becoming a key component for companies' critical business applications (see Exhibit 13-1). Companies are leveraging the Web to enhance communications with customers, vendors, subcontractors, suppliers, and partners, as well as utilizing technologies to reach new audiences and markets. But the same technologies that make the Web such an innovative platform for enhancing communication also dictates the necessity for detailed security planning. The Web has opened up communication to anyone in the world with a computer and a phone line. But the danger is that along with facilitating communication with new markets, customers, and vendors, there is the potential that anyone with a computer and phone line could now access information intended only for a select few.

For companies that have only a few small applications that are accessed by a small set of controlled users, the situation is fairly straightforward. Developers of each application can quickly use directory- or file-level security; if more granular security is required, the developers can embed security in each application housing user information and privileges in a security database. Again, within this scenario, management of a small set of users is less time-consuming and can be handled by a customer service group or the IT security department.

However, most companies are building large Web solutions, many times providing front-end applications to multiple legacy systems on the back

end. These applications are accessed by a diverse and very large population of users, both internal and external to the organization. In these instances, one must move to a different mindset to support logon administration and access controls for hundreds, thousands, and potentially millions of users.

A modified paradigm for security is now a requirement for Web applications: accommodating larger numbers of users in a very noninvasive way. The importance of securing data has not changed; a sure way to lose customers is to have faulty security practices that allow customer information to be accessed by unauthorized outside parties. Further, malicious hackers can access company secrets and critical business data, potentially ruining a company's reputation. However, the new security challenge for organizations now becomes one of transitioning to electronic business by leveraging the Web, obtaining and retaining external constituents in the most customer-intimate and customer-friendly way, while maintaining the requirement for granular access controls and "least privilege."

HOW WEB APPLICATION SECURITY IT FITS INTO AN OVERALL INTERNET SECURITY STRATEGY

Brief Overall Description

Building a secure user management infrastructure is just one component of a complete Internet Security Architecture. While a discussion of a complete Internet Security Architecture (including network security) is beyond the scope of this chapter, it is important to understand the role played by a secure user management infrastructure. The following is a general overview of an overall security architecture (see Exhibit 13-2) and the components that a secure user management infrastructure can help address.

	Management	Security
End User 	• Reporting/Statistics • User Administration • Delegation • Self-Management	• Identification • Authentication
Application 	• Clustering • Policies & Profiles	• Access Controls • Content Filtering • Proxy Services
Data 	• Fault Tolerance • Reporting/Statistics	• Encryption • Auditing
Network 	• Fault Tolerance • Traffic Reporting • Intrusion Detection	• Authentication • Encryption • Auditing • Non-Repudiation

Exhibit 13-2. Internet Security Architecture.

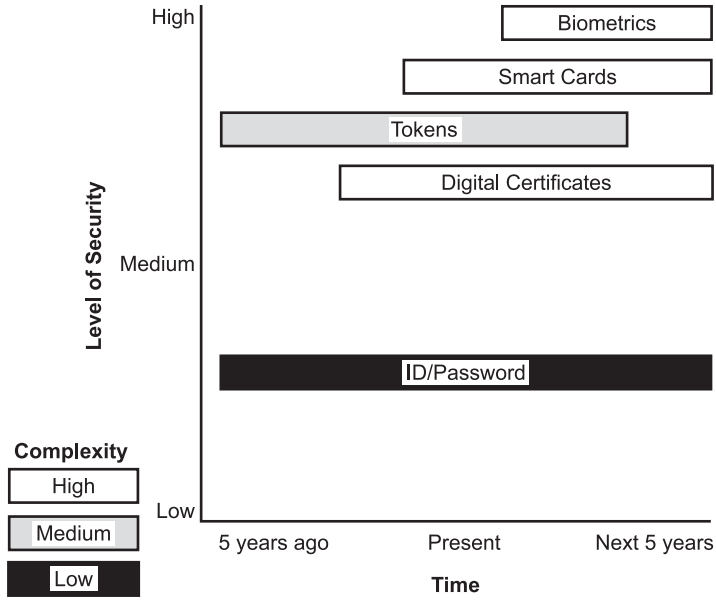


Exhibit 13-3. Authentication time chart.

Authentication

A wide range of authentication mechanisms are available for Web systems and applications. As the Internet matures, more complex and mature techniques will evolve (see Exhibit 13-3). With home-grown developed security solutions, this will potentially require rewriting applications and complicated migrations to new authentication techniques as they become available.

The implementation of a centralized user management architecture can help companies simplify the migration of new authentication techniques by removing the authentication of users from the Internet applications. As new techniques emerge, changes can be made to the user management infrastructure, while the applications themselves would not need major updates, or updates at all.

WHY A WEB APPLICATION AUTHENTICATION/ACCESS CONTROL ARCHITECTURE?

Before deciding whether or not it is necessary to implement a centralized authentication and access control architecture, it is helpful to compare the differences between developing user management solutions for each application and building a centralized infrastructure that is utilized by multiple applications.

Characteristics of decentralized authentication and access control include:

- low initial costs
- quick to develop and implement for small-scale projects
- each application requires its own security solution (developers must build security into each new application)
- user accounts are required for each application
- user must log in separately to each application
- accounts for users must be managed in multiple databases or directories
- privileges must be managed across multiple databases or directories
- inconsistent approach, as well as a lower security level, because common tasks are often done differently across multiple applications
- each system requires its own management procedures increasing administration costs and efforts
- custom solutions may not be scalable as users and transactions increase
- custom solutions may not be flexible enough to support new technologies and security identification schemes
- may utilize an existing directory services infrastructure

Characteristics of centralization authentication and access control include:

- higher start-up costs
- more upfront planning and design required
- a centralized security infrastructure is utilized across multiple applications and multiple Web server platforms
- a single account can be used for multiple applications
- users can log in one time and access multiple applications
- accounts for multiple applications can be managed in a single directory; administration of accounts can easily be distributed to customer service organizations
- privileges can be managed centrally and leveraged over multiple applications
- consistent approach to security, standards are easily developed and managed by a central group and then implemented in applications
- developers can focus on creating applications without having to focus on building security into each application
- scalable systems can be built to support new applications, which can leverage the existing infrastructure
- most centralized solutions are flexible enough to support new technologies; as new technologies and security identification schemes are introduced, they can be implemented independent of applications

Exhibit 13-4. Project phases.

Phase	Tasks
Project planning and initiation	• Develop project scope and objectives • Outline resources required for requirements and design phase • Roles and responsibilities
Requirements	• Develop business requirements • Develop technical requirements • Develop risk assessment • Develop contingency plans • Prioritize requirements and set selection criteria • Roles and responsibilities
Product strategy and selection	• Decide on centralized versus decentralized strategy • Make or buy • Product evaluation and testing • Product selection • License procurement
Design	• Server architecture • Network architecture • Directory services • Directory services strategy • Architecture • Schema • Development environment standards • Administrative responsibilities • Account • Infrastructure
Implementation	• Administrative tools development • Server Implementation • Directory services implementation • Integration
Testing	• Functionality • Performance • Scalability and failover • Testing strategies • Pilot test
Post-implementation	• Ongoing support

PROJECT OVERVIEW

Purpose

Because of the diverse nature of users, data, systems, and applications that can potentially be supported by the centralized user management infrastructure, it is important to ensure that detailed requirements and project plans are developed prior to product selection and implementation (see Exhibit 13-4). Upfront planning will help ensure that all business and

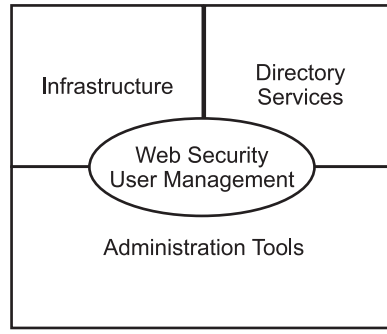


Exhibit 13-5. Web secure user management components.

technical requirements are identified and prioritized, potentially helping prevent serious schedule issues and cost overruns.

PROJECT PLANNING AND INITIATION

Project Components

There are three key components that make up developing an enterprise-wide Web security user management infrastructure (see Exhibit 13-5). While there is significant overlap between components, and each component will affect how the other components will be designed, breaking the project into components makes it more manageable.

Infrastructure. The infrastructure component involves defining the back-end networking and server components of the user management infrastructure, and how that infrastructure integrates into overall Web and legacy data system architecture.

Directory Services. The directory services component involves defining where the user information will be stored, what type of information will be stored, and how that information will be synchronized with other data systems.

Administration Tools. The administration tools component defines the processes and procedures that will be used to manage user information, delegation of administration, and business processes and rules. The administration tools component also involves developing the tools that are used to manage and maintain information.

Roles and Responsibilities

Security. The security department is responsible for ensuring that the requirements meet the overall company security policies and practices.

Security should also work closely with the business to help them identify business security requirements. Processes and procedures should be updated in support of the new architecture.

Business. The business is responsible for identifying the business requirements associated with the applications.

Application Developers. Application developers are responsible for identifying tool sets currently in place, information storage requirements, and other requirements associated with the development of the applications that will utilize the infrastructure.

Infrastructure Components. It is very important for the infrastructure and networking groups to be involved. Infrastructure for support of the hardware, webservers, and directory services. Networking group to ensure that the necessary network connections and bandwidth is available.

REQUIREMENTS

Define Business Requirements

Before evaluating the need for, selecting, and implementing a centralized security authentication infrastructure, it is critical to ensure that all business requirements are thoroughly identified and prioritized. This process is no different than building the business and security requirements for client/server and Internet applications. Identifying the business requirements will help identify the following key issues:

1. What existing security policies and processes are in place?
2. Is the cost of implementing a single centralized infrastructure warranted, or is it acceptable to implement decentralized security in each application?
3. What data and systems will users be accessing? What is the confidentiality of the data and systems being accessed?
4. What are the business security requirements for the data and systems being accessed? Are there regulations and legal issues regarding the information that dictate specific technologies or processes?
5. What type of applications will require security? Will users be accessing more than one application? Should they be allowed single sign-on access?
6. What type of auditing is required? Is it permissible to track user movements in the Web site?
7. Is user personalization required?
8. Is self-registration necessary, or are users required to contact a customer service organization to request a name and password?
9. Who will be responsible for administering privileges? Are there different administration requirements for different user groups?

10. What are the projected numbers of users?
11. Are there password management requirements?
12. Who will be accessing applications/data? Where are these users located? This information should be broken down into groups and categories if possible.
13. What are the various roles of people accessing the data? Roles define the application/data privileges users will have.
14. What is the timeframe and schedules for the applications that the infrastructure will support?
15. What are the cost constraints?

Define Technical Requirements

After defining the business requirements, it is important to understand the existing technical environment and requirements. This will help determine the size and scope of the solution required, what platforms need to be supported, and the development tools that need to be supported by the solution.

Identifying the technical requirements will help identify the following key issues:

1. What legacy systems need to be accessed?
2. What platforms need to be supported?
3. Is there an existing directory services infrastructure in place, or does a new one need to be implemented?
4. What Web development tools are utilized for applications?
5. What are the projected number of users and transactions?
6. How granular should access control be? Can users access an entire Web site or is specific security required for single pages, buttons, objects, and text?
7. What security identification techniques are required: account/password, biometrics, certificates, etc.? Will new techniques be migrated to as they are introduced?
8. Is new equipment required? Can it be supported?
9. What standards need to be supported?
10. Will existing applications be migrated to the new infrastructure, including client/server and legacy applications?
11. What are the cost constraints?

Risk Assessment

Risk assessment is an important part of determining the key security requirements (see Exhibit 13-6). While doing a detailed analysis of a security risk assessment is beyond the scope of this chapter, it is important to understand some of the key analyses that need to be done.

Exhibit 13-6. Risk assessment.

- What needs to be protected?
  - — Data
  - — Systems
- Who are the potential threats?
  - — Internal
  - — External
  - — Unknown
- What are the potential impacts of a security compromise?
  - — Financial
  - — Legal
  - — Regulatory
  - — Reputation
- What are the realistic chances of the event occurring?
  - — Attempt to determine the realistic chance of the event occurring
  - — Verify that all requirements were identified

The benefits of risk assessment include ensuring that one does not spend hundreds of thousands of dollars to protect information that has little financial worth, as well as ensuring that a potential security compromise that could cause millions of dollars worth of damage, in both hard dollars and reputation, does not occur because one did not spend what in hindsight is an insignificant investment.

The most difficult part of developing the risk assessment is determining the potential impacts and the realistic chances of the event occurring. In some cases, it is very easy to identify the financial impacts, but careful analysis must be done to determine the potential legal, regulatory, and reputation impacts. While a security breach may not have a direct financial impact if user information is lost, if publicized on the front page of the business section, the damage caused to one's reputation and the effect that has on attracting new users could be devastating.

Sometimes, it can be very difficult to identify the potential chance of a breach occurring. Threats can come from many unforeseen directions and new attacks are constantly being developed. Steps should be taken to ensure that detailed processes, including monitoring and reviews of audit logs, are done on a regular basis. This can be helpful in identifying existing or potential threats and analyzing their chance of occurrence. Analysis of threats, new and existing, should be performed routinely.

Prioritization and Selection Criteria

After defining the business and technical requirements, it is important to ensure that the priorities are discussed and agreed upon. Each group

should completely understand the priorities and requirements of the other groups. In many cases, requirements may be developed that are nice to have, but are not a priority for implementing the infrastructure. One question that should be asked is: is one willing to delay implementation for an extended amount of time to implement that requirement? For example, would the business group wait an extra six months to deliver the application so that it is personalized to the user, or are they willing to implement an initial version of the Web site and upgrade it in the future? By clearly understanding the priorities, developing selection criteria will be much easier and products can be separated and evaluated based on how well they meet key criteria and requirements.

Selection criteria should be based on the requirements identified and the priorities of all parties involved. A weight should be given to each selection criterion; as products are analyzed, a rating can be given to each selection criterion and then multiplied against the weight. While one product may meet more requirements, one may find that it does not meet the most important selection criterion and, therefore, is not the proper selection.

It is also important to revisit the requirements and their priorities on a regular basis. If the business requirements change during the middle of the product, it is important to understand those changes and evaluate whether or not the project is still moving in the right direction or whether modifications need to be made.

PRODUCT STRATEGY AND SELECTION

Selecting the Right Architecture

Selecting the right infrastructure includes determining whether centralized or decentralized architecture is more appropriate and whether to develop the solution in-house or purchase/implement a third-party solution.

Centralized or Decentralized. Before determining whether to make or buy, it is first important to understand if a centralized or decentralized infrastructure meets the organization's needs (see Exhibit 13-7). Based on the requirements and priorities identified above, it should become obvious as to whether or not the organization should implement a centralized or decentralized architecture. A general rule of thumb can be identified.

Make or Buy. If one has determined that a centralized architecture is required to meet one's needs, then it is realistic to expect that one will be purchasing and implementing a third-party solution. For large-scale Web sites, the costs associated with developing and maintaining a robust and scalable user management infrastructure quickly surpass the costs associated with purchasing, installing, and maintaining a third-party solution.

Exhibit 13-7. Centralized or decentralized characteristics.

Centralized	Decentralized
Multiple applications	Cost is a major issue
Supports large number of users	Small number of applications
Single sign-on access required	One authentication technique
Multiple authentication techniques	Minimal audit requirements
Large-scale growth projected	Manageable growth projected
Decentralized administration	Minimal administration requirements
Detailed audit requirements	

If it has been determined that a decentralized architecture is more appropriate, it is realistic to expect that one will be developing one's own security solutions for each Web application, or implementing a third-party solution on a small scale, without the planning and resources required to implement an enterprisewide solution.

Product Evaluation & Testing. Having made a decision to move forward with buying a third-party solution, now the real fun begins — ensuring that one selects the best product that will meet one's needs, and that can be implemented according to one's schedule.

Before beginning product evaluation and testing, review the requirements, prioritization, and selection criteria to ensure that they accurately reflect the organization's needs. A major determination when doing product evaluation and testing is to define the following:

What are the time constraints involved with implementing the solution? Are there time constraints involved? If so, that may limit the number of tools that one can evaluate or select products based on vendor demonstrations, product reviews, and customer references. Time constraints will also identify how long and detailed one can evaluate each product. It is important to understand that implementing a centralized architecture can be a time-consuming process and, therefore, detailed testing may not be possible. Top priorities should be focused on, with the evaluation of lower priorities based on vendor demonstrations and other resources.

- *Is there an in-house solution already in place?* If there is an in-house solution in place, or a directory services infrastructure that can be leveraged, this can help facilitate testing.
- *Is hands-on testing required?* If one is looking at building a large-scale solution supporting millions of users and transactions, one will probably want to spend some time installing and testing at least one tool prior to making a selection.
- *Are equipment and resources available?* While one might like to do detailed testing and evaluation, it is important to identify and locate

the appropriate resources. Hands-on testing may require bringing in outside consulting or contract resources to perform adequate tests. In many cases, it may be necessary to purchase equipment to perform the testing; and if simultaneous testing of multiple tools is going to occur, then each product should be installed separately.

Key points to doing product evaluation and testing include:

- To help facilitate installation and ensure proper installation, either the vendor or a service organization familiar with the product should be engaged. This will help minimize the lead time associated with installing and configuring the product.
- Multi-function team meetings, with participants from Systems Development, Information Security and Computer Resources, should occur on a regular basis, so that issues can be quickly identified and resolved by all stakeholders.
- If multiple products are being evaluated, each product should be evaluated separately and then compared against the other products. While one may find that both products meet a requirement, it may be that one product meets it better.

Product Selection. Product selection involves making a final selection of a product. A detailed summary report with recommendations should be created. The summary report should include:

- business requirements overview
- technical requirements overview
- risk assessment overview
- prioritization of requirements
- selection criteria
- evaluation process overview
- results of evaluation and testing
- risks associated with selection
- recommendations for moving forward

At this point, one should begin paying special attention to the risks associated with moving forward with the selected product and begin identifying contingency plans that need to be developed.

License Procurement. While selecting a product, it is important to understand the costs associated with implementing that product. If there are severe budget constraints, this may have a major impact on the products that can be implemented. Issues associated with purchasing the product include:

1. How many licenses are needed? This should be broken out by timeframes: immediate (3 months), short term (6 to 12 months), and long term (12 months+).

2. How is the product licensed? Is it a per-user license, site license? Are transaction fees involved? What are the maintenance costs of the licenses? Is there a yearly subscription fee for the software?
3. How are the components licensed? Is it necessary to purchase server licenses as well as user licenses? Are additional components required for the functionality required by the infrastructure?
4. If a directory is being implemented, can that be licensed as part of the purchase of the secure user management product? Are there limitations on how that directory can be used?
5. What type of, if any, implementation services are included in the price of the software? What are the rates for implementation services?
6. What type of technical support is included in the price of the software? Are there additional fees for the ongoing technical support that will be required to successfully maintain the product?

DESIGN

The requirements built for the product selection should be reevaluated at this stage, especially the technical requirements, to ensure that they are still valid. At this stage, it may be necessary to obtain design assistance from the vendor or one of its partner service organizations to ensure that the infrastructure is designed properly and will meet both immediate and future usage requirements. The design phase can be broken into the following components.

Server Infrastructure

The server infrastructure should be the first component analyzed.

- What is the existing server infrastructure for the Internet/intranet architecture?
- What components are required for the product? Do client agents need to be installed on the Web servers, directory servers, or other servers that will utilize the infrastructure?
- What servers are required? Are separate servers required for each component? Are multiple servers required for each component?
- What are the server sizing requirements? The vendor should be able to provide modeling tools and sizing requirements.
- What are the failover and redundancy requirements? What are the failover and redundancy capabilities of the application?
- What are the security requirements for the information stored in the directory/databases used by the application?

Network

The network should next be analyzed.

- What are the network and bandwidth requirements for the secure user management infrastructure?

- What is the existing Internet/intranet network design? Where are the firewalls located? Are traffic load balancers or other redundancy solutions in place?
- If the Internet servers are hosted remotely, what are the bandwidth capabilities between the remote site and one's internal data center?

Directory Services

The building of a complete directory infrastructure in support of a centralized architecture is beyond the scope of this chapter. It is important to note that the directory services are the heart and soul of one's centralized architecture. The directory service is responsible for storing user-related information, groups, rights and privileges, and any potential personalization information. Here is an overview of the steps that need to be addressed at this juncture.

Directory Services Strategy.

- What is the projected number of users?
- The projected number of users will have a major impact on the selection of a directory solution. One should break projections into timeframes: 1 month, 6 months, 1 year, and 2 years.
- Is there an existing directory service in place that can be utilized?
- Does the organization have an existing directory service that can be leveraged? Will this solution scale to meet long-term user projections? If not, can it be used in the short term while a long-term solution is being implemented? For example, the organization might already have a Windows NT domain infrastructure in place; but while this would be sufficient for five to 10,000 users, it cannot scale to meet the needs of 100,000 users.
- What type of authentication schemes will be utilized?
- Determining the type of authentication schemes to be utilized will help identify the type of directory service required. The directory requirements for basic account/password requirements, where one could get away with using a Windows NT domain infrastructure or maybe an SQL infrastructure, are much different than the requirements for a full-scale PKI infrastructure, for which one should be considering a more robust solution, like an LDAP directory service.

Directory Schema Design.

- What type of information needs to be stored?
- What are the namespace design considerations?
- Is only basic user account information being stored, or is additional information, like personal user information and customization features, required? Using a Windows NT domain infrastructure limits the type of information that can be stored about a user, but using an LDAP

or NDS infrastructure allows one to expand the directory schema and store additional information that can be used to personalize the information provided to a user.
- What are the administration requirements?
- What are the account creation and maintenance requirements?

Development Environment

Building a development environment for software development and testing involves development standards.

Development Standards. To take advantage of a centralized architecture, it is necessary to build development security processes and development standards. This will facilitate the design of security into applications and the development of applications (see Exhibit 13-8). The development security process should focus on helping the business and development team design the security required for each application. Exhibit 13-8 is a sample process created to help facilitate the design of security requirements for Web-based applications utilizing a centralized authentication tool.

Administrative Responsibilities

There are multiple components of administration for a secure user management infrastructure. There is administration of the users and groups that will be authenticated by the infrastructure; there is administration of the user management infrastructure itself; and there is the data security administration that is used to develop and implement the policies and rules used to protect information.

Account Administration. Understanding the administration of accounts and user information is very important in developing the directory services architecture. The hierarchy and organization of the directory will resemble how the management of users is delegated.

If self-administration and registration are required, this will impact the development of administrative tools.

Infrastructure Administration. As with the implementation of any enterprisewide solution, it is very important to understand the various infrastructure components, and how those will be administered, monitored, and maintained. With the Web globalizing applications and being "always on," the user management infrastructure will be the front door to many of the applications and commerce solutions that will require 24×7 availability and all the maintenance and escalation procedures that go along with a 24×7 infrastructure.

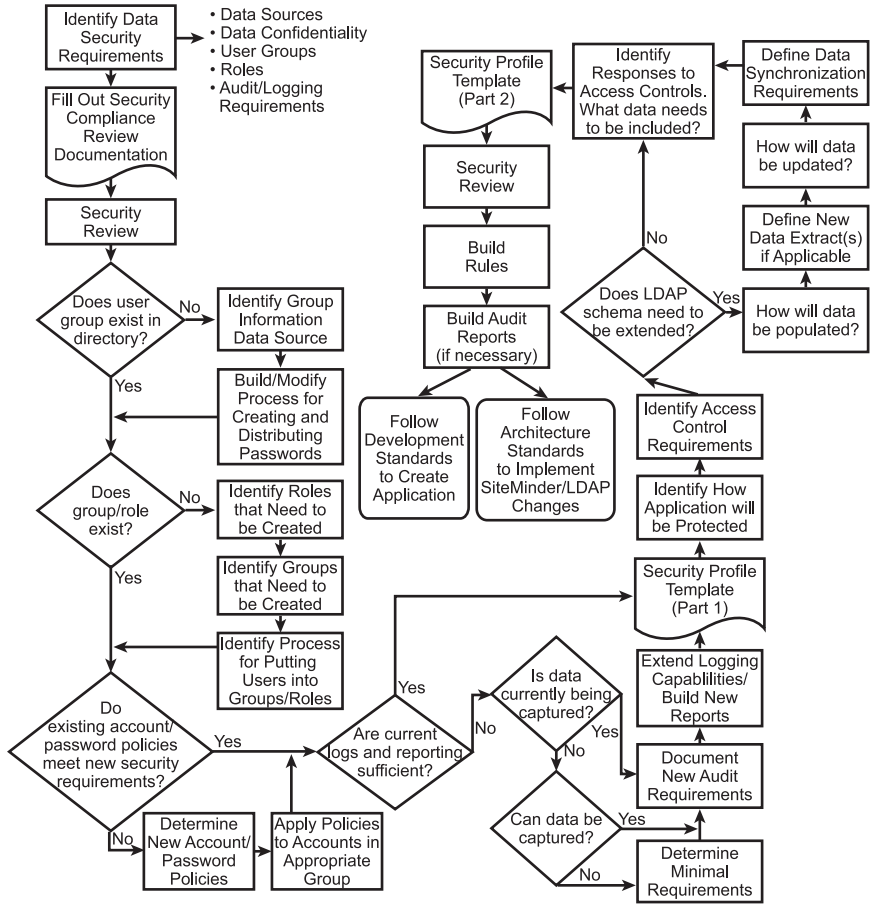


Exhibit 13-8. Application security design requirements.

Data Security Administration. A third set of administrators is required. The role of data security administrators is to work with data owners to determine how the information is to be protected, and then to develop the rules and policies that will be used by the management infrastructure and developers to protect the information.

TESTING

The testing of one's centralized architecture will resemble that of any other large-scale enterprisewide or client/server application. The overall test plan should include all the features listed in Exhibit 13-9.

Exhibit 13-9. Testing strategy examples.

Test	Purpose
Functionality	To ensure that the infrastructure is functioning properly. This would include testing rules and policies to ensure that they are interacting correctly with the directory services. If custom administrative tools are required for the management of the directory, this would also include detailed testing to ensure that these tools were secure and functioning properly.
Performance	Because the centralized infrastructure is the front end to multiple applications, it is important to do performance and scalability testing to ensure that the user management infrastructure does not become a bottleneck and adversely affect the performance and scalability of applications. Standard Internet performance testing tools and methods should be utilized.
Reliability and failover	An important part of maintaining 24×7 availability is built-in reliability, fault tolerance, and failover. Testing should occur to ensure that the architecture will continue to function despite hardware failures, network outages, and other common outages.
Security	Because one's user management infrastructure is ultimately a security tool, it is very important to ensure that the infrastructure itself is secure. Testing would mirror standard Internet and server security tests like intrusion detection, denial-of-service, password attacks, etc.
Pilot test	The purpose of the pilot is to ensure that the architecture is implemented effectively and to help identify and resolve any issues in a small, manageable environment. Because the user management architecture is really a tool used by applications, it is best to integrate the pilot testing of the infrastructure into the roll-out of another application. The pilot group should consist of the people who are going to be using the product. If it is targeted toward internal users, then the pilot end-user group should be internal users. If it is going to be the general internet population, one should attempt to identify a couple of best customers who are willing to participate in a pilot test/beta program. If the member services organization will be administering accounts, then they should be included as part of the pilot to ensure that that process has been implemented smoothly. The pilot test should focus on all aspects of the process, not just testing the technology. If there is a manual process associated with distributing the passwords, then this process needs to be tested as well. One would hate to go live and have thousands of people access accounts the first day only to find out that one has never validated that the mailroom could handle the additional load.

SUMMARY

This chapter is intended to orient the practitioner to the multiple issues requiring attention when an organization implements secure Web applications using third-party commercial software. Designing, implementing, and administering application security architectures, which address and resolve user identification, authentication, and data access controls, have become increasingly popular. In the Web application environment, the author introduces the complexity of *application user* as co-owner of the administration process.

This chapter reflects a real-life scenario, whereby a company with the need to do E-business on the Web, goes through an exercise to determine the cost/benefit and feasibility of building in security versus adding it on, including all of the considerations and decisions made along the way to implementation. For the readers' reference, several products were evaluated, including "getAccess" from EnCommerce and Netegrity's SiteMinder.

98

Reflections on Database Integrity

William Hugh Murray, CISSP

This chapter discusses the concept of database integrity. It contrasts this concept to those of data integrity and database management system integrity. The purpose of the discussion is to arrive at a set of recommendations for the owners and operators of such databases on how to preserve that integrity.

Concepts and Descriptions

This section sets forth some definitions and concepts that describe and bound the issue of database integrity.

Integrity

Integrity is the property of being whole, complete, and unimpaired; free from interference or contamination; unbroken; in agreement with requirements or expectations.

Data can be said to have integrity when it is internally consistent (e.g., the books are in balance) and when it describes what it intends (e.g., the books accurately reflect the performance and condition of the business). A system can be said to have integrity when it performs according to a complete specification most of the time, fails in a predictable manner, presents sufficient evidence of its failure to permit timely and effective corrective action, and permits orderly recovery.

Database

For purposes of this discussion, a database can be defined as a monolithic collection of related or interdependent data elements. Alternatively, it is a monolithic collection of information represented in coded data elements and specific relationships between those data elements. A database is usually intended to be shared across users, uses, or applications.

The abstraction of database is relatively novel, no older than the modern computer. Until the appearance of database management software for the microcomputer, perhaps a decade ago, it was esoteric. Analogous collections of data, such as the books of account for a business, existed before the computer. The term can properly be applied to most of the data that is usually recorded on such media as ledger cards or 3×5 cards. However, it is usually reserved for the most formal, rigorous, and systematic of such collections.

Information in a database can be explicitly represented in the form of coded data elements; employee name is a common example. However, there is other information in the database in the form of associations, both explicit and implicit, between the data elements.

Relationships are special kinds of associations between the data elements. For example, the various fields in an employee database record are related logically in much the same way as they are related on a piece of paper. The meaning and identity of each field is determined, in part, by this context. This information is at least as important as that in the data elements themselves.

The relationships can be expressed in the data itself (relational), in the arrangement or order of the elements within the database (structured), or in metadata, data about the data, that explicitly describes or encodes the relationships (e.g., indexed or object oriented). While databases can be characterized by how the relationships are primarily expressed, in practice, all databases use a combination of these mechanisms. For example, in those databases known as relational, some relationships are expressed in the structure (i.e., tables and views), some in the data (i.e., references to other tables), and some in metadata (the names of the columns).

Database Integrity

A database can be said to have integrity when it preserves the information in the data, that is, when both the data and the relationships are maintained. Database integrity is about the integrity of the records. The integrity of the database is separate from, and can be contrasted to, that of the data, on the one hand, and of the database management system on the other.

Database Management System

For our purposes, a database management system is a generalized, abstract, and automated mechanism for creating, maintaining, storing, preserving, and presenting a database to, and on behalf of, applications.

Database managers are often characterized by the name of the mechanism on which they primarily rely to describe the relationships among the data elements. Thus, database managers in which the relationship between two data elements is normally implied in the data itself, for example, the content of a data element (two employee records have the same department number), or the ordering of the data (employee A precedes B in the sort order of the name field) can be called *relational database managers*. Those in which the relationship is implied by how the two elements are physically stored, (for example, all employees in the same department are stored together, or employee A is always stored before B) can be referred to as *structured database managers.*

Relational Integrity

Relational integrity is the aspect of database integrity that deals with the preservation of the special relationships between the data elements.

Referential integrity is an example and a special case of relational integrity. A reference is a relationship in which a value in one record points to another record, usually of another record type. For our purposes, it is an example and illustration of what it might mean to say that a database has integrity to the extent that relationships are preserved.

Consider the case of an employee record with a department number in it that refers to a department record. If the department number in the employee record is *N*, then referential integrity requires that there be a department record for department *N*. It would prohibit the creation of an employee record with a department number for which there was no corresponding department record, the deletion of department record *N* as long as any employee record pointed to it, and more than one department record *N* for the employee record to point to.

It should be noted that this kind of integrity is optional. That is, the condition could exist, coincidentally or accidentally, without any declaration, commitment, or enforcement. Likewise, it can be implemented and enforced either by using applications or the database management system. As a rule, it is preferable to have it implemented in the database management system so that the mechanism can be shared across applications and so that one application need not rely on another.

Methods

This section discusses some of the methods for implementing database managers and preserving the integrity of the database.

Localization

By definition, a database is a monolith. That is, all of its elements and all of its relationships are essential to its identity. If any element or relationship is lost or broken, then the identity and the integrity are destroyed.

Of course, this is separate from the physical database manager, which might contain two or more independent databases. However, all other things being equal, keeping the elements of the database together helps preserve its integrity. Therefore, most database managers strive to keep the database together.

Single Owning Process

An important form of localization is the single owning process. Because a database is a monolith, there must be a single process that can see all of it, manage it, and have responsibility for its integrity. This owning process is usually the database manager. An implication is that a database manager is usually a single process.

Redundancy

To make the database more reliable than the media and devices on which it is stored, most database managers apply some kind of redundant data. The data is recorded in more than the minimum number of bits otherwise required to express it.

Dynamic Error Detection and Correction

Often, redundancy takes the form of error detection and correction codes. The data is recorded in codes that make the alteration of a bit obvious and its timely and automatic correction possible. One such code is parity, in which an additional bit is added to each frame of 7 or 8 bits to make the frame conform to some arbitrary rule such as odd or even. A variance from the rule signals the alteration of a bit. Some codes are so powerful as to permit the automatic detection and correction of multiple bit errors. These codes can be implemented in both the storage device (i.e., below the line) or in the database manager (above the line between software and hardware-only mechanisms).

Duplication

Redundancy can be carried as far as one or more complete copies of the database or its elements. Such copies can be either inside or outside the database manager. Because relationships are usually best known to the database manager, they are best preserved using the duplication facilities that are provided by it.

Mirroring

One form of duplication is mirroring, in which two synchronized copies of the data are maintained. Mirroring is done internal to a mechanism; the copy is not visible from outside. For example, a file manager can mirror files. It will apply changes to both copies, satisfy requests from either, but conceal the existence of the second copy to processes outside itself. Mirroring can be done on the same device or on a different one. When done on a single device, mirroring protects against a media failure or a limited failure of the device (e.g., a bad track). When done across devices, it protects against a general device failure.

Backup

Backup copies of the database are made independent of the database manager. Among other losses, these copies are specifically intended to protect against damage that might occur to the data if the manager should fail or become corrupt.

Such copies can be prepared automatically by the database manager, or by using utilities or other program processes that are independent of the mechanism itself. Of course, although intended to protect against database manager failures, the use of an independent backup system may itself be a threat to the integrity of the database. It is difficult for an independent system to know and enforce the rules that the database manager itself enforces.

Checkpoints and Journals

A checkpoint is a special case of a backup copy. It is taken at a particular point in time. For example, the initial state of the database, even if empty, is a checkpoint. Checkpoints are used in conjunction with a journal or

log of all update activity subsequent to the checkpoint to reconstruct the database. This mechanism preserves both integrity and currency.

Reconstruction

Such secondary copies can be employed to reconstruct the database, even from massive failures. However, this means that, at least under some circumstances, the integrity of the database will depend on the integrity of these copies.

Compartmentation

To compartmentalize is to place things into segregated compartments. The intent is to contain the effects of what happens in one compartment in such a way as to limit the impact on other compartments. For example, one might run multiple small database managers, in preference to a single large one, so as to limit the impact of a failure.

Segregation and Independence

Database management systems often implement segregation and independence of sub-processes to preserve integrity. For example, they may isolate the process that does an update from that which checks to see that it was done correctly and from the one that attempts corrective action. The purpose is to minimize the chances that the same fault will affect all three.

Encapsulation

The database manager can be viewed as a package, container, or capsule, one role of which is to protect the database from any outside interference or contamination. Encapsulation can be either physical or logical. For a database manager, physical encapsulation might be provided by placing it in a separate computer. Logical encapsulation might be provided by placing it in an isolated and protected process within an environment provided by a shared computer and its operating system. Logical encapsulation may also be provided, in part, and in static conditions, by the use of secret codes.

Most database management systems provide some encapsulation of the databases they contain. Object-oriented database management systems do so, by definition, explicitly and globally. Increasingly, one sees database managers themselves being encapsulated in their own hardware.

Hiding

Capsules hide or conceal their contents so that they cannot be seen or addressed from the outside. While this does not make the database safer from destruction, it does protect it from unauthorized disclosure and from malicious, but covert, change. Hiding can be implemented in many ways; the most common are by means of process-to-process isolation, data typing and type managers, and by the use of secret codes.

Binding

Binding is used to resolve and fix, for example, a data characteristic or reference, so as to resist later change. In computer science, one speaks of early and late binding. For example, in some programming, symbolic names are bound, that is, resolved so as to resist later change at compile time, while in others the same characteristic may not be bound until execution time.

Many structured database management systems can bind relationships in the database at programming time or at load time. This tends to improve both the integrity and performance at the expense of loss of flexibility and increased maintenance cost. Relational database managers also employ binding of table existence at creation time.

Binding applies only within the environment in which it takes place. If data or databases are removed from the database manager, then characteristics are no longer bound or reliable.

Atomic Update

Atomic update means that any change to the database takes place completely or not at all. There are no partial updates. This includes both data elements and relationships. Most database managers implement this by maintaining the ability to "roll back" any partial updates that they are unable to complete.

Locking

One potential threat to the integrity of a database results from concurrent use by two or more processes. For example, where two users make changes to a database, there is some potential that the second change will overwrite the first. Database management systems are expected to provide mechanisms, such as locking, that resist such problems.

Locking is a mechanism that database managers employ to ensure that partially updated elements and relationships are not used. It involves marking the element as "in use" or "asking for the lock" for all elements involved in an update. The mechanism will not permit a second use of an element that is in use and will not begin an update until it can obtain the locks for all elements involved. However, locking is ordinarily a logical, rather than physical, mechanism. It is usually just a bit or flag that is set by locking or unlocking.

Locking may come in several levels of transparency and granularity. Ideally, locking would be automatic and transparent to all users or using processes. However, this might have unnecessary performance impact. For example, for maximum transparency, a database management system might restrict access from application B to any data that A is looking at, on the assumption that A might elect to update it. Thus, B will see a performance penalty even if he does not care about potential updates.

Performance might also require that B's access be limited to only the smallest element that A might update. B should not be restricted from an entire table simply because A is interested in a single row of the table. Thus, maximum performance requires that both A and B declare their intent.

Access Control

Access control is a mechanism provided by the database management system to enable the owners and managers of the database to control which users or using processes can alter the database, its elements, or its relationships. These controls are most likely to be included in database management systems intended for use by multiple users. It is an integrity mechanism in that it reduces the size of the population that can alter the database to the intended population. It can also be used to enforce dual controls intended to resist errors and malice.

Privileged Controls

Most database management systems, particularly those that provide access controls, provide what can be referred to as privileged controls. These controls are intended for use by the managers of the system. They are intended for use to exercise ultimate control, particularly to remedy unusual situations. Two unusual situations are of particular interest. The first is to override the access controls. This capability may be necessary to avoid a deadlock situation. The second is the use of such privilege to repair the database itself. In the early days of structured databases, such controls were frequently used to "repair broken chains."

It should be noted that such privilege includes the ability to contaminate or interfere with the database.

Reconciliation

Reconciliation refers to an act or process that brings the database into harmony or consistency; that is, the act or process of checking the database against expectation and correcting for variances. Normally, database management systems perform this kind of checking on a routine, automatic, frequent, and repetitive, if not quite continuous, basis. For example, after making a `WRITE` request to another process (e.g., the file system), the database manager can make an immediate inspection to satisfy itself that the request completed correctly. The routine and automatic nature of this activity, among other things, distinguishes it from recovery. Another is that it relies almost exclusively on internal resources.

Recovery

Recovery is the integrity mechanism of last resort, the one that is used when the database is broken beyond the ability of any other mechanism to repair it. It is usually externally invoked and relies on external resources such as backup copies of the data. While it must bring the database back to a state of integrity, it may do so at the expense of currency or even lost data.

Conclusions

Database integrity is essential. If one cannot rely on the data, it is useless. Integrity is easier to preserve than to recreate. No single tool or mechanism is sufficient unto itself. Database management systems will employ a variety of tools, and owners and managers will compensate for the inherent limitations of the database managers by employing tools that are completely external to it.

At least four things are necessary to preserve the integrity of a database:

1. One must preserve both the data elements and the relationships among them.
2. One must understand and exploit the mechanisms provided by the database management systems.
3. One must not compromise any of these mechanisms, either in the way one uses them or external to them.
4. One must understand the limitations of the database management system and compensate for them.

A simple copy of the data elements may not preserve the information contained in the relationships. For example, if a structured database contains information about the relationships in the physical location of the data within the device, then a copy of the data can preserve the relationships only if it is on an identical device.

Because all database management systems employ a combination of mechanisms to implement relationships and because most of these mechanism are concealed, management or operational procedures that bypass the database management system are suspect. On the other hand, if there are no measures taken to preserve integrity that are independent of the database management system, then a failure of the mechanism can destroy the database.

It should be noted that the most robust database managers so encapsulate the database that they cannot be bypassed. Any attempt to do so will result, at best, in the distortion of the database, and, at worst, in the destruction of the database and the database management system. Most of these systems will also provide one or more built-in mechanisms for creating external representations of the database.

One final issue is that of scale. Most databases are relatively small when compared to the systems and devices on which they reside. However, many of the most important databases are very large and span tens or even hundreds of devices. In such databases, information about relationships can span many devices. The integrity of the database requires the preservation of the devices and their relationship to each other.

On the other hand, it is common in these databases to create external copies by backing up the devices rather than the database or even the files. Such backups are device and device-field dependent. While they provide adequate protection against the failure of one or two devices, recovery from the destruction of the entire environment might require the complete replication of the environment. Timeliness may require that this be done in days or even hours. Thus, in exactly the databases in which it may be most urgent to have device-independent backups, it may be least likely to have them.

Recommendations

This section sets forth recommendations for preserving the integrity of databases. These include some recommendations for using the database management system and some for compensating for its limitations.

1. Choose a database manager whose characteristics, features, and properties are sufficiently robust for the intended application and environment. Consider the size of the database and its importance to the enterprise.
2. Use the database management system according to directions. Note and respect all limitations.
3. Place the database and its manager in a robust environment.
4. Provide adequate resources (e.g., mirror files, devices, and control units) as indicated by the application and environment.

5. Prefer monolithic databases for integrity. Use distributed database managers only to the extent justified by major differences in performance.
6. For integrity, prefer a one-to-one relationship between a database, a database management system, and a processor. Share only to the extent indicated by major economies of scale. Keep in mind that today's computer systems can be more readily scaled to their applications. Large-scale sharing no longer offers the economies that it used to.
7. Prefer relational and object-oriented databases for integrity. Prefer structured databases for performance.
8. Applications and users should check those behaviors of the database manager that they rely on.
9. Limit access to the database and to elements within it to the minimum number of known users and processes consistent with the application.
10. Apply access controls in such a way as to involve multiple people in sensitive updates to the database.
11. Involve multiple people in the use of privileged or potent controls.
12. Keep multiple backup copies and generations of the data, including checkpoints and journals of update activity.
13. Prefer device-independent backups, particularly for databases that span multiple devices.
14. For device independence, prefer to make backups with services provided by the database manager. Use independent mechanisms for performance.
15. Prefer to make backups with services provided by the database manager for preservation of relationships. Prefer backups made by other means for independence and to protect against failure in the mechanism.
16. To protect external copies of the database, involve multiple people in their custody.
17. Check integrity after recovery and before use. Remember that even normal use of a corrupt database may spread the damage and that using bad data may result in serious damage to the enterprise.

Data Marts and Data Warehouses: Keys to the Future or Keys to the Kingdom?

M. E. Krehnke
D. K. Bradley

WHAT DO YOU THINK WHEN YOU HEAR THE TERM "DATA MART" OR "DATA WAREHOUSE"? CONVENIENCE? AVAILABILITY? CHOICES? CONFUSION FROM OVERWHELMING OPTIONS? POWER? SUCCESS? Organizational information, such as marketing statistics or customer preferences, when analyzed, can mean power and success in today's and future markets. If it is more convenient for a customer to do business with a "remembering" organization — one that retains and uses customer information (e.g., products used, sales trends, goals) and does not have to ask for the information twice — then that organization is more likely to retain and grow that customer's base.[1] There are even organizations whose purpose is to train business staff to acquire competitor's information through legal, but espionage-like techniques, calling it "corporate intelligence."[2]

DATA WAREHOUSES AND DATA MARTS: WHAT ARE THEY?

Data warehouses and data marts are increasingly perceived as vital organizational resources and — given the effort and funding required for their creation and maintenance, and their potential value to someone inside (or outside) the organization — they need to be understood, effectively used, and protected. Several years ago, one data warehouse proponent suggested a data warehouse's justification that includes support for

"merchandising, logistics, promotions, marketing and sales programs, asset management, cost containment, pricing, and product development," and equated the data warehouse with "corporate memory."[3]

The future looked (and still looks) bright for data warehouses, but there are significant implementation issues that need to be addressed, including scalability (size), data quality, and flexibility for use. These are the issues highlighted today in numerous journals and books — as opposed to several years ago when the process for creating a data warehouse and its justification were the primary topics of interest.

Data Warehouse and Data Mart Differences

Key differences between a data warehouse and data mart are size, content, user groups, development time, and amount of resources required to implement. A data warehouse (DW) is generally considered to be organizational in scope, containing key information from all divisions within a company, including marketing, sales, engineering, human resources, and finance, for a designated period of time. The users, historically, have been primarily managers or analysts (aka power users) who are collecting and analyzing data for planning and strategy decisions. Because of the magnitude of information contained in a DW, the time required for identifying what information should be contained in the warehouse, and then collecting, categorizing, indexing, and normalizing the data, is a significant commitment of resources, generally taking several years to implement.

A data mart (DM) is considered to be a lesser-scale data warehouse, often addressing the data needs of a division or an enterprise or addressing a specific concern (e.g., customer preferences) of a company. Because the amount and type of data are less varied, and the number of users who have to achieve concurrence on the pertinent business goals is fewer, the amount of time required to initiate a DM is less. Some components of a DM can be available for use within nine months to a year of initiation, depending on the design and scope of the project. If carefully planned and executed, it is possible for DMs of an enterprise to actually function as components of a (future) DW for the entire company. These DMs are linked together to form a DW via a method of categorization and indexing (i.e., metadata) and a means for accessing, assembling, and moving the data about the company (i.e., middleware software). It is important to carefully plan the decision support architecture, however, or the combination of DMs will result in expensive redundancy of data, with little or no reconciliation of data across the DMs. Multiple DMs within an organization cannot replace a well-planned DW.[4]

Data Warehouse and Data Mart Similarities

Key similarities between a DW and DM include the decisions required regarding the data before the first byte is ever put into place:

- What is the strategic plan for the organization with regard to the DW architecture and environment?
- What is the design/development/implementation process to be followed?
- What data will be included?
- How will the data be organized?
- How and when will the data be updated?

Following an established process and plan for DW development will help ensure that key steps are performed — in a timely and accurate manner by the appropriate individuals. (Unless noted otherwise, the concepts for DWs also apply to DMs.) The process involves the typical development steps of requirements gathering, design, construction, testing, and implementation.

The DW or DM is not an operational database and, as such, does not contain the business rules that can be applied to data before it is presented to the user by the original business application. Merely dumping all the operational data into the DW is not going to be effective or useful. Some data will be summarized or transformed, and other data may not be included. All data will have to be "scrubbed" to ensure that quality data is loaded into the DW. Careful data-related decisions must be made regarding the following:[5]

- business goals to be supported
- data associated with the business goals
- data characteristics (e.g., frequency, detail)
- time when transformation of codes is performed (e.g., when stored, accessed)
- schedule for data load, refresh, and update times
- size and scalability of the warehouse or mart

Business Goals Identification. The identification of the business goals to be supported will involve the groups who will be using the system. Because DWs are generally considered to be nonoperational decision support systems, they will contain select operational data. This data can be analyzed over time to identify pertinent trends or, as is the case with data mining, be used to identify some previously unknown relationship between elements that can be used to advance the organization's objectives. It is vital, however, that the DW be linked to, and supportive of, the strategic goals of the business.

Data Associated with Business Goals. The data associated with the identified business goals may be quantitative (e.g., dollar amount of sales) or qualitative (i.e., descriptive) in nature. DWs are not infinite in nature, and decisions must be made regarding the value of collecting, transforming, storing, and updating certain data to keep it more readily accessible for analysis.

Data Characteristics. Once the data has been identified, additional decisions regarding the number of years to be stored and the level of frequency to be stored have to be made. A related, tough decision is the level of detail. Are item sales needed: by customer, by sale, by season, by type of customer, or some other summary? Resources available are always going to be limited by some factor: funding, technology, or available support.

Data Transformation and Timing. Depending on the type of data and its format, additional decisions must be made regarding the type and timing of any transformations of the data for the warehouse. Business applications usually perform the transformations of data before they are viewed on the screen by the user or printed in a report, and the DW will not have an application to transform the data. As a result, users may not know that a certain code means engineering firm (for example) when they retrieve data about XYZ Company to perform an analysis. Therefore, the data must be transformed prior to its presentation, either before it is entered into the database for storage or before the user sees it.

Data Reloading and Updating. Depending on the type and quantity of data, the schedules for data reloading or data updating may require a significant amount of time. Decisions regarding the reload/update frequency will have to be made at the onset of the design because of the resources required for implementing and maintaining the process. A crucial decision to be made is: will data be reloaded en masse or will only changed data be loaded (updated)? A DW is nonoperational, so the frequency for reload/update should be lower than that required for an operational database containing the same or similar information. Longer reload and update times may impact users by limiting their access to the required information for key customer-related decisions and competition-beating actions. Data maintenance will be a substantial component of ongoing costs associated with the DW.

Size and Scalability. Over time, the physical size of the DW increases because the amount of data contained increases. The size of the database may impact the data updating or retrieval processes, which may impact the usage rate; as well, an increase in the number of users will also impact the retrieval process. Size may have a strongly negative impact on the cost, performance, availability, risk, and management of the DW. The ability of a DW to grow in size and functionality and not affect other critical factors is called scalability, and this capability relies heavily on the architecture and technologies to be used, which were agreed upon at the time the DW was designed.

DATA QUALITY

The quality of data in a DW is significant because it contains summarized data, addresses different audiences and functions than originally intended,

and depends on other systems for its data. The time-worn phrase "garbage in, garbage out" is frequently applied to the concept of DW data. Suggested ways to address data quality include incorporating metadata into the data warehouse structure, handling content errors at load time, and setting users' expectations about data quality. In addition, "it is mandatory to track the relationships among data entities and the calculations used over time to ensure that essential referential integrity of the historical data is maintained."[6]

Metadata Incorporation into the DW Design

Metadata is considered to be the cornerstone of DW success and effective implementation. Metadata not only supports the user in the access and analysis of the data, but also supports the data quality of the data in the warehouse.

The creation of metadata regarding the DW helps the user define, access, and understand data needed for a particular analysis or exploration. It standardizes all organizational data elements (e.g., the customer number for marketing and finance organizations), and acts as a "blueprint" to guide the DW builders and users through the warehouse and to guide subsequent integration of later data sources.

Metadata for a DW generally includes the following[7]:

1. organizational business models and rules
2. data view definitions
3. data usage model
4. report dictionary
5. user profiles
6. physical and logical data models
7. source file data dictionaries
8. data element descriptions
9. data conversion rules

Standardization of Metadata Models

The importance of metadata to the usefulness of a DW is a concept mentioned by most of the authors reviewed. Metadata and its standardization are so significant that Microsoft has joined the Metadata Coalition (MDC) consortium. Microsoft turned its metadata model, the Open Information Model (OIM), over to the MDC for integration into the MDC Metadata Interchange Specification (MDIS). This standard will enable various vendors to exchange metadata among their tools and databases, and support proprietary metadata.[8] There are other vendors, however, that are reluctant to participate and are maintaining their own versions of metadata management.[9] But this present difference of opinions does not diminish the need for comprehensive metadata for a DW.

Setting User Expectations Regarding Data Quality

Metadata about the data transformations can indicate to the user the level of data quality that can be expected. Depending on the user, the availability of data may be more significant than the accuracy, and this may be the case for some DMs. But because the DW is intended to contain significant data that is maintained over the long term and can be used for trend analysis, data quality is vital to the organization's DW goals. In a "Report from the Trenches," Quinlan emphasizes the need to manage user expectations and identify potential hardships as well as benefits.[10] This consideration is frequently mentioned in discussions of requirements for a successful DW implementation.

Characteristics of data quality are[11]:

- *accuracy:* degree of agreement between a set of data values and a corresponding set of correct values
- *completeness:* degree to which values are present in the attributes that require them
- *consistency:* agreement or logical coherence among data that frees them from variation or contradiction
- *relatability:* agreement or logical coherence that permits rational correlation in comparison with other similar or like data
- *timeliness:* data item or multiple items that are provided at the time required or specified
- *uniqueness:* data values that are constrained to a set of distinct entries, each value being the only one of its kind
- *validity:* conformance of data values that are edited for acceptability, reducing the probability of error

DW USE

The proposed use of a DW will define the initial contents, and the initial tools and analysis techniques. Over time, as users become trained in its use and there is proven applicability to organizational objectives, the content of a DW generally expands and the number of users increases. Therefore, developers and management need to realize that it is not possible to create the "perfect warehouse." Users cannot foresee every decision that they are going to need to make and define the information they need to do so. Change is inevitable. Users become more adept at using the DW and want data in more detail than they did initially; users think of questions they had not considered initially; and business environments change and new information is needed to respond to the current marketplace or new organizational objectives.[12] This is why it is important to plan strategically for the DW environment.

Types of Users

DWs are prevalent today in the retailing, banking, insurance, and communications sectors; and these industries tend to be leaders in the use of

business intelligence/data warehouse (BI/DW) applications, particularly in financial and sales/marketing applications.[13] Most organizations have a customer base that they want to maintain and grow (i.e., providing additional products or services to the same customer over time). The use of DWs and various data exploration and analysis techniques (such as data mining) can provide organizations with an extensive amount of valuable information regarding their present or potential customer base. This valuable information includes cross-selling and up-selling, fraud detection and compliance, potential lifetime customer value, market demand forecasting, customer retention/vulnerability, product affinity analysis, price optimization, risk management, and target market segmentation.

Techniques of Use

The data characteristics of the DW are significantly different from those of a transactional or operational database, presenting large volumes of summary data that address an extensive time period, which is updated on a periodic (rather than daily) basis. The availability of such data, covering multiple areas of a business enterprise over a long period of time, has significant value in organizational strategic marketing and planning. The availability of metadata enables a user to identify useful information for further analysis. If the data quality is high, the user will have confidence in the results.

The type of analysis performed is determined, in part, by the capabilities of the user and the availability of software to support the analysis. The usefulness of the data can be related to the frequency of updates and the level of detail provided in the DW. There are three general forms of study that can be performed on DW data[14]:

1. *analysis:* discovering new patterns and hypotheses for existing, unchanging data by running correlations, statistics, or a set of sorted reports
2. *monitoring:* automatic detection of matches or violations of patterns to provide a timely response to the change
3. *discovery:* interactive identification, a process of uncovering previously unknown relationships, patterns, and trends that would not necessarily be revealed by running correlations, statistics, or a set of sorted reports

The DW is more applicable for the "monitoring" and "discovery" techniques because the resources available are more fully utilized. It is possible that ad hoc analysis may be accepted in such a positive manner that scheduled reports are then performed as a result of that analysis, in which case the method changes from "discovery" to simply "analysis." However, the discovery of patterns (offline) can then be used to define a set of rules that will automatically identify the same patterns when compared with new, updated data online.

Data Mining. Data mining is a prevalent DW data analysis technique. It can be costly and time-consuming, because the software is expensive and may require considerable time for the analyst to become proficient. The benefits, however, can be quite remarkable. Data mining can be applied to a known situation with a concrete, direct question to pursue (i.e., reactive analysis) or to an unknown situation with no established parameters. The user is "seeking to identify unknown patterns and practices, detect covert/unexplained practices, and have the capability to expose organized activity (i.e., proactive invigilation)."[14]

Data mining is an iterative process, and additional sources can be introduced at any time during the process. It is most useful in exploratory analysis scenarios with no predetermined expectations as to the outcome. Data mining is not a single-source (product/technology) solution, and must be applied, as any tool, with the appropriate methodological approach. When using data mining, the analyst must consider:

- organizational requirements
- available data sources
- corporate policies and procedures

There are questions that have to be answered to determine if the data mining effort is worthwhile, including[15]:

1. Are sufficient data sources available to make the effort worthwhile?
2. Is the data accurate, well coded, and properly maintained for the analyst to produce reasonable results?
3. Is permission granted to access all of the data needed to perform the analysis?
4. Are static extractors of data sufficient?
5. Is there an understanding of what things are of interest or importance to set the problem boundaries?
6. Have hypothetical examples been discussed beforehand with the user of the analysis?
7. Are the target audience and the intent known (e.g., internal review, informational purposes, formal presentation, or official publication)?

Activities associated with data mining are[16]:

- *classification:* establishing a predefined set of labels for the records
- *estimation:* filling in missing values in a particular field
- *segmentation:* identification of subpopulations with similar behavior
- *description:* spotting any anomalous or "interesting" information

Data mining goals may be[17]:

- *predictive:* models (expressed as executable code) to perform some form of classification or estimation

- *descriptive:* informational by uncovering patterns and relationships

Data to be mined may be[17]:

- *structured:* fixed length, fixed format records with fields that contain numeric values, character codes, or short strings
- *unstructured:* word or phrase queries, combining data across multiple, diverse domains to identify unknown relationships

The data mining techniques (and products) to be used will depend on the type of data being mined and the end objectives of the activity.

Data Visualization. Data visualization is an effective data mining technique that enables the analyst and the recipients to discern relationships that may not be evident from a review of numerical data by abstracting the information from low-level detail into composite representations. Data visualization presents a "top-down view of the range of diversity represented in the data set on dimensions of interest."[18]

Data visualization results depend on the quality of data. "An ill-specified or preposterous model or a puny data set cannot be rescued by a graphic (or by calculation), no matter how clever or fancy. A silly theory means a silly graphic."[19] Data visualization tools can, however, support key principles of graphical excellence[20]:

- a well-designed presentation of interesting data through "substance, statistics, and design"
- communication of complex ideas with "clarity, precision, and efficiency"
- presentation of the "greatest number of ideas in the shortest time with the least ink in the smallest space"

Enterprise Information Portals. Extended use of the Internet and Web-based applications within an enterprise now supports a new form of access, data filtering, and data analysis: a personalized, corporate search engine — similar to the Internet personalized search engines (e.g., My Yahoo) — called a corporate portal, enterprise information portal, or business intelligence portal. This new tool provides multiple characteristics that would be beneficial to an individual seeking to acquire and analyze relevant information[21]:

- ease of use through a Web browser
- filtering out of irrelevant data
- integration of numerical and textual data
- capability of providing alerts when certain data events are triggered.

Enterprise information portals (EIPs) can be built from existing data warehouses or from the ground up through the use of Extensible Markup Language (XML). XML supports the integration of unstructured data resources (e.g., text documents, reports, e-mails, graphics, images, audio,

and video) with structured data resources in relational and legacy databases.[22] Business benefits associated with the EIP are projected to include[23]:

- leverage of DW, Enterprise Resource Planning (ERP), and other IT systems
- transforming E-commerce business into "true" E-business
- easing reorganization, merger, and acquisition processes
- providing improved navigation and access capabilities

But it is emphasized that all of the design and implementation processes and procedures, network infrastructures, and data quality required for successful DWs s must be applied to ensure an EIP's potential for supporting enterprise operations and business success.

Results

The results of data mining can be very beneficial to an organization, and can support numerous objectives: customer-focused planning and actions, business intelligence, or even fraud discovery. Examples of industries and associated data mining uses presented in *Data Mining Solutions, Methods and Tools for Real-World Problems*[18] include:

- *pharmaceuticals:* research to fight disease and degenerative disorders by mapping the human genome
- *telecommunications:* customer profiling to provide better service
- *retail sales and marketing:* managing the market saturation of individual customers
- *financial market analysis:* managing investments in an unstable Asian banking market
- *banking and finance:* evaluation of customer credit policy and the reduction of delinquent and defaulted car loans
- *law enforcement and special investigative units:* use of financial reporting regulations and data to identify money-laundering activities and other financial crimes in the United States by companies

Other examples are cited repeatedly throughout data management journals, such as *DM Review*. The uses of data mining continue to expand as users become more skilled, and as the tools and techniques increase in options and capabilities.

RETURNS ON THE DW INVESTMENT

Careful consideration and planning are required before initiating a DW development and implementation activity. The resources required are substantial, although the benefits can surpass the costs many times.

Costs

The DW design and implementation activity is very labor intensive, and requires the involvement of numerous business staff (in addition to Information Technology staff) over the entire life cycle of the DW, in order for the project to be successful by responding to organizational information needs. Although technology costs over time tend to drop, while providing even greater capabilities, there is a significant investment in hardware and software. Administration of the DWs is an ongoing expense. Because DWs are not static and will continue to grow in terms of the years of data and the types of data maintained, additional data collection and quality control are required to ensure continued viability and usefulness of the corporate information resource.

Costs are incurred throughout the entire DW life cycle; some costs are one-time costs, others are recurrent costs. One-time costs and a likely percentage of the total DW budget (shown in parentheses) include[24]:

- *hardware:* disk storage (30%), processor costs (20%), network communication costs (10%)
- *software:* database management software (10%); access/analysis tools (6%); systems management tools: activity monitor (2%), data monitor (2%); integration and transformation(15%); interface creation, metadata creation and population (5%)

Cost estimates (cited above) are based on the implementation of a centralized (rather than distributed) DW, with use of an automated code generator for the integration and transformation layer.

Recurrent costs include[24]:

- refreshment of the data warehouse data from the operational environment (55%)
- maintenance and update of the DW and metadata infrastructure (3%)
- end-user training (6%)
- data warehouse administration — data verification of conformance to the enterprise data model (2%), monitoring (7%), archiving (1%), reorganization/restructuring (1%); servicing DW requests for data (21%); capacity planning (1%); usage analysis (2%); and *security administration* (1%) [emphasis added]

The recurrent costs are almost exclusively associated with the administrative work required to keep the DW operational and responsive to the organization's needs. Additional resources may be required, however, to upgrade the hardware (e.g., more storage) or for the network to handle an unexpected increase in the volume of requests for DW information over time. It is common for the DW budget to grow an order of magnitude per

year for the first two years that the DW is being implemented. After the first few years, the rate of growth slows to 30 or 40 percent growth per year.[24]

The resources that should be expended for any item will depend on the strategic goals that the DW is intended to support. Factors affecting the actual budget values include[24]:

- size of the organization
- amount of history to be maintained
- level of detail required
- sophistication of the end user
- competitive marketplace participant or not
- speed with which DW will be constructed
- construction of DW is manual or automated
- amount of summary data to be maintained
- creation of integration and transformation layer is manual or automated
- maintenance of the integration and transformation layer is manual or automated

MEASURES OF SUCCESS

The costs for a DW can be extraordinary. Bill Inmon shows multiple DMs costing in the tens of millions in the graphics in his article on metadata and DMs.[4] Despite the costs, William McKnight indicates that that a recent survey of DW users has shown a range of return on investment (ROI) for a three-year period between 1857 percent and 16,000 percent, with an average annual ROI of 401 percent.[25] However, Douglas Hackney cautions that the sample sets for some DW ROI surveys were self-selected and the methodology flawed. Hackney does say that there are other ROI measures that need to be considered: "pure financial ROI, opportunity cost, 'do nothing' cost and a 'functional ROI'. In the real world, your financial ROI may be 0 percent, but the overall return of all the measures can easily be over 100 percent."[26] So, the actual measures of success for the DW in an organization, and the quantitative or qualitative values obtained, will depend on the organization.

Internal customers and their focus should be considered when determining the objectives and performance measures for the DW ROI, including[25]:

- sales volume (sales and marketing)
- reduced expenses (operations)
- inventory management (operations)
- profits (executive management)
- market share (executive management)
- improved time to market (executive management)
- ability to identify new markets (executive management)

DWs respond to these objectives by bringing together, in a cohesive and manageable group, subject areas, data sources, user communities, business rules, and hardware architecture.

Expanding on the above metrics, other benefits that can significantly impact the organization's well-being and its success in the marketplace are[25]:

- reduced losses due to fraud detection
- reduced write-offs because of (previous) inadequate data to combat challenges from vendors and customers
- reduced overproduction of goods and commensurate inventory holding costs
- increased metrics on customer retention, targeted marketing and an increased customer base, promotion analysis programs with increased customer numbers and penetration, and lowering time to market

Mergers by companies in today's market provide an opportunity for cross-selling by identifying new, potential customers for the partners or by providing additional services that can be presented for consideration to existing customers. Responsiveness to customers' needs, such as speed (submitting offers to a customer prior to the customer making a decision) and precision (tailoring offerings to what is predicted the customer wants), can be facilitated with a well-designed and well-utilized DW. Associated actions can include the automatic initiation of marketing activity in response to known buying or attrition triggers, or tools that coordinate a "continuous customized communication's stream with customers." Data mining expands the potential beyond query-driven efforts by identifying previously unknown relationships that positively affect the customer base.[27]

MISTAKES TO AVOID

The Data Warehousing Institute conducted meetings with DW project managers and Information Systems executives in 1995 to identify the "ten mistakes to avoid for data warehousing managers" and created a booklet (Ten Mistakes Booklet) that is available from the institute.[28] Time has not changed the importance or the essence of the knowledge imparted through the experienced contributors. Although many authors have highlighted one or more topics in their writings, this source is very succinct and comprehensive. The "Ten Data Warehousing Mistakes to Avoid" and a very brief explanation are noted below.[29]

1. *Starting with the wrong sponsorship chain.* Supporters of the DW must include an executive sponsor with funding and an intense interest in the effective use of information, a project "driver" who keeps the

project moving in the right direction with input from appropriate sources, and the DW manager.

2. *Setting expectations that one cannot meet and frustrating executives at the moment of truth.* DWs contain a select portion of organizational information, often at a summary level. If DWs are portrayed as "the answer" to all questions, then users are going to be disappointed. User expectations must be managed.
3. *Engaging in politically-naïve behavior.* DWs are a tool to support managers. To say that DWs will "help managers make better decisions" can alienate potential supporters (who may have been performing well *without* a DW).
4. *Loading the warehouse with information just because it was available.* Extraneous data makes it more difficult to locate the essential information and slows down the retrieval and analysis process. The data selected for inclusion in the DW must support organizational strategic goals.
5. *Believing that the data warehousing database design is the same as the transactional database design.* DWs are intended to maintain and provide access to selected information from operational (transactional) databases, generally covering long periods of time. The type of information contained in a DW will cross multiple divisions within the organization, and the source data may come from multiple databases and may be summarized or provided in detail. These characteristics (as well as the database objectives) are substantially different from those of operational or transactional databases.
6. *Choosing a data warehouse manager who is technology oriented rather than user oriented.* Data warehousing is a service business — not a storage business — and making clients angry is a near-perfect method of destroying a service business.
7. *Focusing on traditional internal record-oriented data and ignoring the potential value of external data and text, images, and — potentially — sound and video.* Expand the data warehouse beyond the usual data presentation options and include other vital presentation options. Users may ask: Where is the copy of the contract (image) that explains the information behind the data? Where is the ad (image) that ran in that magazine? Where is the tape (audio or video) of the key competitor at a recent conference talking about its business strategy? Where is the recent product launch (video)? Being able to provide the required reference data will enhance the analysis that the data warehouse designers and sponsors endeavor to support.
8. *Delivering data with overlapping and confusing definitions.* Consensus on data definitions is mandatory, and this is difficult to attain because multiple departments may have different meanings for the same term (e.g., sales). Otherwise, users may not have confidence in

the data they are acquiring. Even worse, they may acquire the wrong information, embarrass themselves, and blame the data warehouse.

9. *Believing the vendor's performance, capacity, and scalability promises.* Planning to address the present and future DW capacity in terms of data storage, user access, and data transfer is mandatory. Budgeting must include unforeseen difficulties and costs associated with less than adequate performance by a product.
10. *Believing that once the data warehouse is up and running, one's problems are finished.* Once they become familiar with the data warehouse and the process for acquiring and analyzing data, users are going to want additional and different types of data than that already contained in the DW. The DW project team must be maintained after the initial design and implementation takes place for on-going DW support and enhancement.
11. *Focusing on ad hoc data mining and periodic reporting.* (Believing there are only ten mistakes to avoid is also a mistake.) Sometimes, ad hoc reports are converted into regularly scheduled reports, but the recipients may not read the reports. Alert systems can be a better approach and make a DW mission-critical, by monitoring data flowing into the warehouse and informing key people with a need-to-know as soon as a critical event takes place.

Responsiveness to key business goals — high-quality data, metadata, and scalable architecture — is emphasized repeatedly by many DW authors, as noted in the next section on suggestions for DW implementation.

DW IMPLEMENTATION

Although the actual implementation of a DW will depend on the business goals to be supported and the type and number of users, there are general implementation considerations and measures of success that are applicable to many circumstances.

General Considerations

As expected, implementation suggestions are (basically) the opposite of the mistakes to avoid. There is some overlap in the suggestions noted because there are multiple authors cited. Suggestions include:

1. Understand the basic requirements.
2. Design a highly scalable solution.
3. Deliver the first piece of the solution into users' hands quickly.[30]
4. Support a business function that is directly related to the company's strategy; begin with the end in mind.
5. Involve the business functions from the project inception throughout its lifecycle.

6. Ensure executive sponsorship understands the DW value, particularly with respect to revenue enhancement that focuses on the customer.
7. Maintain executive sponsorship and interest throughout the project.[31]
8. Develop standards for data transformation, replication, stewardship, and naming.
9. Determine a cost-justification methodology, and charge users for data they request.
10. Allow sufficient time to implement the DW properly, and conduct a phased implementation.
11. Designate the authority for determining data sources and populating the metadata and DW data to Data Administration.
12. Monitor data usage and archive data that is rarely or never accessed.[5]
13. Budget resources for metadata creation. Make metadata population a metric for the development team.
14. Budget resources for metadata maintenance. Any change in the data requires a change in the metadata.
15. Ensure ease of access. Find and deploy tools that seamlessly integrate metadata.[32]
16. Monitor DW storage growth and data activity to implement reasonable capacity planning.
17. Monitor user access and analysis techniques to ensure that they optimize usage of the DW resources.
18. Tune the DW for performance based on usage patterns (e.g., selectively index data, partition data, create summarization, and create aggregations).
19. Support both business metadata and technical metadata.
20. Plan for the future. Ensure that interface between applications and the DW is as automated as possible. Data granularity allows for continuous DW tuning and reorganization, as required to meet user needs and organization strategic goals.
21. Consider the creation of an "exploration warehouse" for the "out-of-the-box thinkers" who may want to submit lengthy resource-consuming queries — if they become a regular request.[33]

Qualitative Measures of DW Implementation Success

In 1994, Sears, Roebuck and Co. (a leading U.S. retailer of apparel, home, and automotive products that operates 3000 department and specialty stores) implemented a DW to address organizational objectives. The eight (qualitative) measures of success presented below are based on the experiences associated with the Sears DW implementation.

1. *Regular implementation of new releases.* The DW and applications are evolving to meet business needs, adding functionality through a phased implementation process.

2. *Users will wait for promised system upgrades.* When phases will deliver the functionality that is promised, at the expected quality level and data integrity level, planned implementation schedules (and possibly slippage) will be tolerated.
3. *New applications use the DW to serve their data requirements.* Increased reliance on the DW provides consistency company-wide and is cost effective.
4. *Users and support staff will continue to be involved in the DW.* Users become reliant on the DW and the part it plays in the performance of their work responsibilities. Therefore, there needs to be a permanent DW staff to support the constantly changing DW and business environment. When product timeliness is crucial to profitability, then designated staff (such as the Sears Business Support Team) and the DW staff can provide additional, specialized support to meet user needs.
5. *The DW is used to identify new business opportunities.* As users become familiar with the DW, they will increasingly pursue new discovery opportunities.
6. *Special requests become the rule, not the exception.* The ability to handle special requests on a routine basis is an example of DW maturity and a positive leverage of DW resources.
7. *Ongoing user training.* New and advanced user training (e.g., troubleshooting techniques, sophisticated functionality) and the provision of updated documentation (highlighting new features) facilitate and enhance DW use in support of business objectives.
8. *Retirement of legacy systems.* Use of legacy systems containing duplicate information will decline. Retirement of legacy systems should follow a planned process, including verification of data accuracy, completeness, and timely posting, with advance notification to identified users for a smooth transition to the DW applications.[34]

DW SECURITY IMPLICATIONS

The benefits of the well-implemented and well-managed DW can be very significant to a company. The data is integrated into a single data source. There is considerable ease of data access that can be used for decision-making support, including trends identification and analysis, and problem-solving. There is overall better data quality and uniformity, and different views of the same data are reconciled. Analysis techniques may even uncover useful competitive information. But with this valuable warehouse of information and power comes substantial risk if the information is unavailable, destroyed, improperly altered, or disclosed to or acquired by a competitor.

There may be additional risks associated with a specific DW, depending on the organization's functions, its environment, and the resources available

for the DW design, implementation, and maintenance — which must be determined on an individual basis — that are not addressed here. Consider the perspective that the risks will change over time as the DW receives increased use by more sophisticated internal and external users; supports more functions; and becomes more critical to organizational operations.

DW Design Review

Insofar as the literature unanimously exhorts the need for upper-management support and applicability to critical business missions, the importance of the system is significant before it is even implemented. Issues associated with availability, integrity, and confidentiality should be addressed in the system design, and plans should include options for scalability and growth in the future. The DW must be available to users when they need the information; its integrity must be established and maintained; and only those with a need-to-know must access the data. Management must be made aware of the security implications and requirements, and security should be built into the DW design.

DW Design is Compliant with Established Corporate Information Security Policy, Standards, Guidelines, and Procedures. During the design phase, certain decisions are being made regarding expected management functions to be supported by the DW: user population (quantity, type, and expertise level); associated network connectivity required; information to be contained in the initial phase of the DW; data modeling processes and associated data formats; and resources (e.g., hardware, software, staff, data) necessary to implement the design. This phase also has significant security implications. The DW design must support and comply with corporate information security policies, including:

- non-disclosure statements signed by employees when they are hired[35]
- installation and configuration of new hardware and software, according to established corporate policies, guidelines, and procedures
- documentation of acceptable use and associated organizational monitoring activities
- consistency with overall security architecture
- avoidance of liability for inadequately addressing security through "negligence, breach of fiduciary duty, failing to use the security measures found in other organizations in the same industry, failing to exercise due care expected from a computer professional, or failure to act after an 'actual notice' has taken place"[45]
- protection from prosecution regarding inappropriate information access by defining appropriate information security behavior by authorized users[46]

DW Data Access Rights Are Defined and modeled for the DW User Population. When determining the access requirements for the DW, your initial users may be a small subset of employees in one division. Over time, it will expand to employees throughout the entire organization, and may include selected subsets of subcontractors, vendors, suppliers, or other groups who are partnering with the organization for a specific purpose. Users will not have access to all DW information, and appropriate access and monitoring controls must be implemented. Areas of security concern and implementation regarding user access controls include:

- DW user roles' definition for access controls (e.g., role-based access controls)
- user access rights and responsibilities documentation
- development of user agreements specifying security responsibilities and procedures[35]
- definition of user groups and their authorized access to specific internal or external data
- user groups and their authorized levels of network connectivity and use definitions
- definition of procedures for review of system logs and other records generated by the software packages[37]

DW Data Content and Granularity Is Defined and Appropriately Implemented in the DW Design. Initially, the DW content may be internal organizational numerical data, limited to a particular department or division. As time passes, the amount and type of data is going to increase, and may include internal organizational textual data, images, and videos, and external data of various forms as well. In addition, the required granularity of the data may change. Users initially may be comfortable with summary data; but as their familiarity with the DW and the analysis tools increases, they are going to want more detailed data, with a higher level of granularity than originally provided. Decisions that affect data content and its integrity throughout the DW life cycle include:

- Data granularity (e.g., summary, detail, instance, atomic) is defined.
- Data transformation rules are documented for use in maintaining data integrity.
- Process is defined for maintaining all data transformation rules for the life of the system.

Data Sensitivity Is Defined and Associated with Appropriate Access Controls Issues associated with data ownership, sensitivity, labeling, and need-to-know will need to be defined so that the data can be properly labeled, and access requirements (e.g., role-based access controls) can be assigned.

Establishment of role-based access controls "is viewed as effective and efficient for general enterprise security" and would allow the organization to expand the DW access over time, and successfully manage a large number of users.[38] Actions required that define and establish the data access controls include:

- determination of user access control techniques, including the methods for user identification, authentication, and authorization
- assignment of users to specific groups with associated authority, capabilities, and privileges[38] for role-based access controls
- determination of database controls (e.g., table and data labeling, encryption)
- establishment of a process for granting access and for the documentation of specified user roles and authorized data access, and a process for preventing the circumvention of the granting of access controls
- establishment of a process for officially notifying the Database Administrator (or designated individual) when an individual's role changes and his or her access to data must be changed accordingly
- establishment of a process for periodically reviewing access controls, including role-based access controls to ensure that only individuals with specified clearances and need-to-know have access to sensitive information

Data Integrity and Data Inference Requirements Are Defined and Associated with Appropriate Access Controls. Data integrity will be reviewed when the data is transformed for the DW, but should be monitored on a periodic basis throughout the life cycle of the DW, in cooperation with the DW database administration staff. Data inference and aggregation may enable an individual to acquire information for which he or she has no need-to-know, based on the capability to acquire other information. "An inference presents a security breach if higher-classified information can be inferred from lower-classified information."[39]

Circumstances in which this action might occur through data aggregation or data association in the DW need to be identified and addressed through appropriate data access controls. Data access controls to prevent or reduce unauthorized access to information obtained through a data inference process (i.e., data aggregation or data association) can include[39]:

- *Appropriate labeling of information:* unclassified information is reclassified (or labeled at a higher level) to prevent unauthorized inferences by data aggregation or data association.
- *Query restriction:* all queries are dominated by the level of the user, and inappropriate queries are aborted or modified to include only authorized data.

- *Polyinstantiation:* multiple versions of the same information item are created to exist at different classification levels.
- *Auditing:* a history of user queries is analyzed to determine if the response to a new query might suggest an inference violation.
- *Toleration of limited inferences:* inferred information violations do not pose a serious threat, and the prevention of certain inferences may be unfeasible.

Operating System, Application, and Communications Security Requirements Are Defined. Many DWs are using a Web-based interface, which provides easy accessibility and significant risk. Depending on the location of the system, multiple security mechanisms will be required. Actions required to define the security requirements should be based on a risk analysis and include:

- determination of mechanisms to ensure operating system and application system availability and integrity (e.g., firewalls, intrusion detection systems)
- determination of any secure communication requirements (e.g., Secure Socket Layer, encryption)

Plans for Hardware Configuration and Backup Must Be Included in the DW Design. The creation of a DW as a separate, nonoperational function will result in a duplication of hardware resources, because the operational hardware is maintained separately. In examples of mature DW utilizations, a second DW is often created for power users for "exploratory research" because the complexity of their analysis requests would take too much time and resources away from the other general users of the initial DW. This is then (possibly) a third set of hardware that must be purchased, configured, maintained, administered, and protected. The hardware investment keeps increasing. Documentation and updating of hardware and backup configurations should be performed as necessary.

Plans for Software Distribution, Configuration, and Use Must Be Included in the DW Design. The creation of one or multiple DWs also means additional operating system, application, middleware, and security software. In addition, as the number of users increases, the number of licensed software copies must also increase. Users may not be able to install the software themselves and so technical support may need to be provided. Distribution activities should ensure that:

- users have authorized copies of all software
- technical support is provided for software installation, use, and troubleshooting to maintain licensing compliance and data integrity

Plans for Continuity of Operations and Disaster Recovery Must Be Included in the DW Design. Capabilities for hardware and software backup, continuity of operations, and disaster recovery options will also have to be considered. The DW is used to implement strategic business goals, and downtime must be limited. As more users integrate the DW data into their routine work performance, more users will be negatively impacted by its unavailability. Activities in support of operations continuity and disaster recovery should include:

- designations from the design team regarding the criticality of data and key functions
- creation of an alternative hardware list
- resource allocations for DW system backups and storage
- resource allocations for business continuity and disaster recovery plans

Plans for Routine Evaluation of the Impact Of Expanded Network Connectivity on Organizational Network Performance Must Be Included in the DW Design. Over time, with the increased number of users and the increased amount and type of data being accessed in the DW and transmitted over the organizational network, network resources are going to be "stressed." Possible options for handling increased network loads will need to be discussed. Network upgrades may be required over the long term and this needs to be considered in the resource planning activities. Otherwise data availability and data integrity may be impacted at crucial management decision times — times when one wants the DW to stand out as the valuable resource it was intended to be. Changes in network configurations must be documented and comply with organizational security policies and procedures. Planning to address DW scalability and the ability of the network to respond favorably to growth should include:

- evaluation of proposed network configurations and the expected service to be provided by a given configuration against DW requirements[40]
- estimation of DW network requirements' impact on existing organizational network connectivity requirements and possible reduction in data availability or integrity
- consideration of network connectivity options and the effects on the implementation of security

DW Security Implementation Review

A security review must be conducted to ensure that all the DW components supporting information security that were defined during the design phase are accurately and consistently installed and configured. Testing must be performed to ensure that the security mechanisms and database processes perform in a reliable manner and that the security mechanisms enforce established access controls. Availability of data must be consistent

with defined requirements. The information security professional, the database administrator, and the network administrator should work together to ensure that data confidentiality, integrity, and availability are addressed.

Monitor the Acquisition and Installation of DW Technology Components in Accordance with Established Corporate Security Policies. When acquired, the hardware and software DW components must be configured to support the corporate security policies and the data models defined during the design phase. During installation, the following actions should take place: (1) documentation of the hardware and software configurations; and (2) testing of the system before operational to ensure compliance with policies.

Review the Creation/Generation of Database Components for Security Concerns. A process should be established to ensure that data is properly labeled, access requirements are defined and configured, and all controls can be enforced. In cooperation with the design team, individuals responsible for security should perform a review of the database configurations for compliance with security policies and defined data access controls. Database processes must enforce the following data integrity principles[39]:

1. *Well-formed transactions:* transactions support the properties of correct-state transformation, serialization, failure atomicity, progress (transaction completion), entity integrity, and referential integrity.
2. *Least privilege:* programs and users are given the minimum access required to perform their jobs.
3. *Separation of duties:* events that affect the balance of assets are divided into separate tasks performed by different individuals.
4. *Reconstruction of events:* user accountability for actions and determination of actions are performed through a well-defined audit trail.
5. *Delegation of authority:* process for acquisition and distribution of privileges is well-defined and constrained.
6. *Reality checks:* cross-checks with an external reality are performed.
7. *Continuity of operations:* system operations are maintained at an appropriate level.

Review the Acquisition of DW Source Data. DW data is coming from other sources; ensure that all internal and external data sources are known and documented, and data use is authorized. If the data is external, ensure that appropriate compensation for the data (if applicable) has been made, and that access limitations (if applicable) are enforced.

Review testing. Configuration settings for the security mechanisms must be verified, documented, and protected from alteration. Testing to ensure that the security mechanisms are installed and functioning properly must be performed and documented prior to the DW becoming operational. A

plan should also be established for the testing of security mechanisms throughout the life cycle of the DW, including the following situations:

- routine testing of security mechanisms on a scheduled basis
- hardware or software configurations of the DW are changed
- circumstances indicate that an unauthorized alteration may have occurred
- a security incident occurs or is suspected
- a security mechanism is not functioning properly

DW Operations

The DW is not a static database. Users and information are going to be periodically changing. The process of data acquisition, modeling, labeling, and insertion into the DW must follow the established procedures. Users must be trained in DW use and updated as processes or procedures change, depending on the data being made available to them. More users and more data mean additional demands will be placed on the organization's network, and performance must be monitored to ensure promised availability and data integrity. Security mechanisms must be monitored to ensure accurate and consistent performance. Backup and recovery procedures must also be implemented as defined to ensure data availability.

Participate as a Co-instructor in DW User Instruction/Training. Training will be required for users to fully utilize the DW. This is also an opportunity to present (and reinforce) applicable information security requirements and the user's responsibility to protect enterprise information and other areas of concern. Activities associated with this include:

- promotion of users' understanding of their responsibilities regarding data privacy and protection
- documentation of user responsibilities and nondisclosure agreements

Perform Network Monitoring for Performance. Document network performance against established baselines to ensure that data availability is being implemented as planned.

Perform Security Monitoring for Access Control Implementation. Review defined hardware and software configurations on a periodic basis to ensure no inappropriate changes have been made, particularly in a distributed DW environment. Security monitoring activities should include:

- review of user accesses to verify established controls are in place and operational, and no unauthorized access is being granted (e.g., individual with role X is being granted to higher level data associated with role Y)

- provision of the capability for the DW administrator to cancel a session or an ID, as might be needed to combat a possible attack[35]
- review of operating system and application systems to ensure no unauthorized changes have been made to the configurations

Perform Software Application and Security Patches in a Timely and Accurate Manner. All patches must be installed as soon as they are received and documented in the configuration information.

Perform Data and Software Backups and Archiving. As data and software are changed, backups must be performed as defined in the DW design. Maintaining backups of the current data and software configurations will support any required continuity of operations or disaster recovery activities. Backups must be stored offsite at a remote location so that they are not subject to the same threats. If any data is moved from the DW to remote storage because it is not currently used, then the data must be appropriately labeled, stored, and protected to ensure access in the event that the information is needed again.

Review DW Data and Metadata Integrity. DW data will be reloaded or updated on a periodic basis. Changes to the DW data may also require changes to the metadata. The data should be reviewed to determine that the updates are being performed on the established schedule, are being performed correctly, and the integrity of the data is being maintained.

DW Maintenance

DW maintenance is a significant activity, because the DW is an ever-changing environment, with new data and new users being added on a routine basis. All security-relevant changes to the DW environment must be reviewed, approved, and documented prior to implementation.

Review and Document the Updating of DW Hardware and Software. Over time, changes will be made to the hardware and software, as technology improves or patches are required in support of functions or security. Associated activities include:

- installation of all software patches in a timely manner and documentation of the software configuration
- maintenance of software backups and creation of new backups after software changes
- ensuring new users have authorized copies of the software
- ensuring that system backup and recovery procedures reflect the current importance of the DW to organizational operations. If DW criticality has increased over time with use, has the ability to respond to this new level of importance been changed accordingly?

Review the Extraction/Loading of Data Process and Frequency to Ensure Timeliness and Accuracy. The DW data that is to be updated will be extracted from a source system, transformed, and then loaded into the DW. The frequency with which this activity is performed will depend on the frequency with which the data changes, and the users' needs regarding accurate and complete data. The process required to *update* DW data takes significantly less time than that required to *reload* the entire DW database, but there has to be a mechanism for determining what data has been changed. This process needs to be reviewed, and adjusted as required, throughout the life cycle of the DW.

Scheduling/Performing Data Updates. Ensure that data updates are performed as scheduled and the data integrity is maintained.

DW Optimization

Once the DW is established within an organization, it is likely that there will be situations in which individuals or organizations are working to make the DW better (e.g., new data content and types), cheaper (e.g., more automated, less labor intensive), and faster (e.g., new analysis tools, better network connectivity and throughput). Optimization will result in changes, and changes need to be reviewed in light of their impact on security. All changes should be approved before being implemented and carefully documented.

Participate in User Refresher/Upgrade Training. Over time, additional data content areas are going to be added to the DW and new analysis tools may be added. Users will need to be trained in the new software and other DW changes. This training also presents an opportunity to present any new security requirements and procedures — and to review existing requirements and procedures associated with the DW.

Review and Update the Process for Extraction/Loading of Data. As new data requirements evolve for the DW, new data may be acquired. Appropriate procedures must be followed regarding the access, labeling, and maintenance of new data to maintain the DW reputation regarding data integrity and availability.

Review the Scheduling/Performance of Data Updates. Over time, users may require more frequent updates of certain data. Ensure that data updates are performed as scheduled and that data integrity is maintained.

Perform Network Monitoring for Performance. Document network performance against established baselines to ensure that data availability is being implemented as planned. An expanded number of users and increased demand for large volumes of data may require modifications to

the network configuration or to the scheduling of data updates. Such modifications may reduce the network traffic load at certain times of the day, week, or month, and ensure that requirements for data availability and integrity are maintained.

Perform Security Monitoring for Access. The DW information can have substantial operational value or exchange value to a competitor or a disloyal employee, as well as to the authorized users. With the use of corporate "portals" of entry, all of the data may be available through one common interface — making the means and opportunity for "acquisition" of information more easily achieved. Implementation of access controls needs to be continually monitored and evaluated throughout the DW life cycle. The unauthorized acquisition or dissemination of business-sensitive information (such as privacy data, trade secrets, planning information, or financial data) could result in lost revenue, company embarrassment, or legal problems. Monitoring access controls should be a continual security procedure for the DW.

Database Analysis. Some existing DW data may not be used with the expected frequency and it may be moved to another storage location, creating space for data more in demand. Changes in DW data configurations and locations should be documented.

There may be additional risks associated with an actual DW, depending on the organization's functions, its environment, and the resources available for the DW design, implementation, and maintenance — which must be determined on an individual basis. But with careful planning and implementation, the DW will be a valuable resource for the organization and help the staff to meet its strategic goals — now and in the future.

CONCLUSION

The security section presented some of the security considerations that need to be addressed throughout the life cycle of the DW. One consideration not highlighted above is the amount of time and associated resources (including equipment and funding) necessary to implement DW security. Bill Inmon estimated Security Administration to be 1 percent of the total warehouse costs, with costs to double the first year and then grow 30 to 40 percent after that. Maintaining adequate security is a crucial DW and organizational concern. The value of the DW is going to increase over time, and more users are going to have access to the information. Appropriate resources must be allocated to the protection of the DW. The ROI to the organization can be very significant if the information is adequately protected. If the information is not protected, then someone else is getting the keys to the kingdom. Understanding the DW design and implementation

process can enable security professionals to justify their involvement early on in the design process and throughout the DW life cycle, and empower them to make appropriate, timely security recommendations and accomplish their responsibilities successfully.

Notes

1. Peppers, Don and Rogers, Martha, Mass Customization: Listening to Customers, *DM Review*, 9(1), 16, January 1999.
2. Denning, Dorothy E., *Information Warfare and Security*, Addison-Wesley, Reading, MA, July 1999, 148.
3. Saylor, Michael, Data Warehouse on the Web, *DM Review*, 6(9), 22–26, October 1996.
4. Inman, Bill, Meta Data for the Data Mart Environment, *DM Review*, 9(4), 44, April 1999.
5. Adelman, Sid, The Data Warehouse Database Explosion, *DM Review*, 6(11), 41–43, December 1996.
6. Imhoff, Claudia and Geiger, Jonathan, Data Quality in the Data Warehouse, *DM Review*, 6(4), 55–58, April 1996.
7. Griggin, Jane, Information Strategy, *DM Review*, 6(11), 12, 18, December 1996.
8. Mimmo, Pieter R., Building Your Data Warehouse Right the First Time, *Data Warehousing: What Works*, Vol. 9, November 1999, The Data Warehouse Institute Web site: www.dw-institute.com.
9. King, Nelson, Metadata: Gold in the Hills, *Intelligent Enterprise*, 2(3), 12, February 16, 1999.
10. Quinlan, Tim, Report from the Trenches, *Database Programming & Design*, 9(12), 36–38, 40–42, 44–45, December 1996.
11. Hufford, Duane, Data Warehouse Quality, *DM Review*, 6(3), 31–34, March 1996.
12. Rudin, Ken, The Fallacy of Perfecting the Warehouse, *DM Review*, 9(4), 14, April 1999.
13. Burwen, Michael P., BI and DW: Crossing the Millennium, *DM Review*, 9(4), 12, April 1999.
14. Westphal, Christopher and Blaxton, Teresa, *Data Mining Solutions, Methods and Tools for Solving Real-World Problems*, Wiley Computer Publishing, New York, 1998, 68–69.
15. Westphal, Christopher and Blaxton, Teresa, *Data Mining Solutions, Methods and Tools for Solving Real-World Problems*, Wiley Computer Publishing, New York, 1998, 19–24.
16. Westphal, Christopher and Blaxton, Teresa, *Data Mining Solutions, Methods and Tools for Solving Real-World Problems*, Wiley Computer Publishing, New York, 1998, xiv–xv.
17. Westphal, Christopher and Blaxton, Teresa, *Data Mining Solutions, Methods and Tools for Solving Real-World Problems*, Wiley Computer Publishing, New York, 1998, xv.
18. Westphal, Christopher and Blaxton, Teresa, *Data Mining Solutions, Methods and Tools for Solving Real-World Problems*, Wiley Computer Publishing, New York, 1998, 35.
19. Tufte, Edward R., *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, CT, 1983, 15.
20. Tufte, Edward R., *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, CT, 1983, 51.
21. Osterfelt, Susan, Doorways to Data, *DM Review*, 9(4), April 1999.
22. Finkelstein, Clive, Enterprise Portals and XML, *DM Review*, 10(1), 21, January 2000.
23. Schroeck, Michael, Enterprise Information Portals, *DM Review*, 10(1), 22, January 2000.
24. Inmon, Bill, The Data Warehouse Budget, *DM Review*, 7(1), 12–13, January 1997.
25. McKnight, William, Data Warehouse Justification and ROI, *DM Review*, 9(10), 50–52, November 1999.
26. Hackney, Douglas, How About 0% ROI?, *DM Review*, 9(1), 88, January 1999.
27. Suther, Tim, Customer Relationship Management, *DM Review*, 9(1), 24, January 1999.
28. The Data Warehousing Institute (TDWI), 849-J Quince Orchard Boulevard, Gaithersburg, MD 20878, (301) 947-3730, www.dw-institute.com.
29. The Data Warehousing Institute, Data Warehousing: What Works?, Gaithersburg, MD, Publication Number 295104, 1995.
30. Rudin, Ken, The Fallacy of Perfecting the Warehouse, *DM Review*, 9(4), 14, April 1999.
31. Schroeck, Michael J., Data Warehouse Best Practices, *DM Review*, 9(1), 14. January 1999.
32. Hackney, Douglas, Metadata Maturity, *DM Review*, 6(3), 22, March 1996.
33. Inmon, Bill, Planning for a Healthy, Centralized Warehouse, Bill Inmon, *Teradata Review*, 2(1), 20–24, Spring 1999.

34. Steerman, Hank, Measuring Data Warehouse Success: Eight Signs You're on the Right Track, *Teradata Review*, 2(1), 12–17, Spring 1999.
35. Fites, Philip and Kratz, Martin, *Information Systems Security: A Practitioner's Reference,* International Thomson Computer Press, Boston, 10.
36. Wood, Charles C., *Information Security Policies Made Easy*, Baseline Software, Sausalito, CA, 6.
37. Wood, Charles C., *Information Security Policies Made Easy*, Baseline Software, Sausalito, CA, 5.
38. Murray, William, Enterprise Security Architecture, *Information Security Management Handbook*, 4th ed., Harold F. Tipton and Micki S. Krause, Eds., Auerbach, New York, 1999, chap. 13, 215–230.
39. Sandhu, Ravi S. and Jajodia, Sushil, Data Base Security Controls, *Handbook of Information Security Management*, Zella G. Ruthberg and Harold F. Tipton, Eds., Auerbach, Boston, 1993, chap. II-3-2, 481–499.
40. Kern, Harris et al., *Managing the New Enterprise*, Prentice-Hall, Sun SoftPress, NJ, 1996, 120.

99

Digital Signatures in Relational Database Applications

Mike R. Prevost

Now that public key encryption and its associated infrastructure (PKI) have become an accepted foundation for securing the electronic world, a wealth of new security products has come on the scene. However, it appears that many of these products are solving security problems related to the infrastructure upon which business applications run rather than the applications themselves. For example, virtual private network (VPN) products are beginning to support certificate-based authentication and public key-based key exchange. SSL is the standard for privacy and authentication on the Web. Although these types of technologies are completely necessary, they are all highly specialized and are invisible to the applications they are securing.

The nature of digital signature technology and its use in database-driven applications require a certain amount of application integration. It is this integration step that has been the primary technical stumbling block to the widespread use of digital signatures. PKI programming is still a "black art" known only to the few who have conquered its formidable layers of complexity. PKI integration projects have proven too costly and too risky for many application owners. As a result, organizations seem to be focusing on ways to add security to applications without performing complex integrations. However, in moving from securing our infrastructure to securing our applications, there is a growing genre of data security products that are making it easier to integrate security features such as digital signatures into the applications themselves.

This chapter discusses the issues associated with integrating digital signature functionality into relational database applications. First, this chapter focuses on some concepts about digital signatures and the role that digital signatures play in an application security strategy, followed by an explanation of why relational database applications are different from other environments and a discussion of some of the pitfalls of the various integration approaches. Finally, the chapter outlines an "application generic" solution to digitally signing data stored in relational databases that is very easy to integrate into applications.

Digital Signature Concepts

In relational database applications, digital signatures are typically used to ensure data integrity or non-repudiation (i.e., proof of origin). Because digital signatures are semantically similar to paper signatures, they are used to streamline business processes by reducing or entirely eliminating the need to print, sign, transfer, and store paper documents. The legal framework for holding signers accountable for documents they digitally sign is beginning to take shape.

Note that digital signature is only one element of a complete application security plan. The focus on digital signature does not at all diminish need for other technologies such as encryption, authentication, authorization, access control, firewalls, and intrusion detection. Digital signature does, however, provide important security services that are not addressed by other technologies.

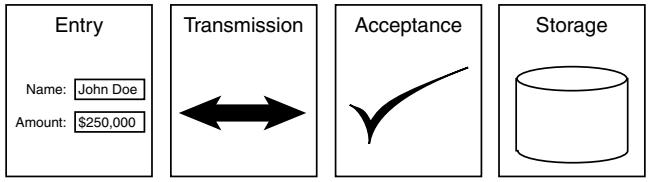


EXHIBIT 99.1 Four steps in a transaction.

The Anatomy of a Transaction

When discussing application security, the term "transaction" is often used. This is a very vague term that brings to mind financial or business transactions. Sometimes, the term "document" is used. For the immediate purposes, a transaction (or document) is any exchange between the user and the application that results in a change to data that is stored by the application. In database applications, the transaction data is stored in a relational database.

Exhibit 99.1 breaks a transaction into four steps. Each step has unique security requirements. This diagram serves as a basis for illustrating how digital signatures fit into the overall security requirements of an application. The order of these steps may be different for some application architectures.

Step 1: Data Entry

Because transactions involve data, the data has to originate somewhere. This usually means that a user enters it on some sort of data entry screen. In this step, the application is probably concerned with data validation: ensuring that all required data fields are populated in a format that the application can understand. Applications may also want to prevent certain users from accessing certain data entry screens.

Step 2: Data Transmission

In many applications, transaction data is transferred across a network to a central application server or database server. Applications may need to ensure that the transaction data is not altered during transmission. Also, the transaction may include sensitive information such as credit card numbers or other private, personal information. It is also likely that applications may require assurance that the data is being transmitted to the intended recipient. The popular SSL protocol satisfies these requirements for Web-based applications. Virtual private networking (VPN) technologies can also provide these services.

Step 3: Acceptance

At some point in the process, the application or application server "accepts" the transaction. That is, the transaction meets all the requirements necessary to be processed. Accepting a transaction can involve several elements:

- *Data validation*. All required fields are entered in a format that the application can understand.
- *Integrity*. The data has not been altered during transmission to the application or database server.
- *Authentication*. The identity of the user has been firmly established.
- *Authorization*. The authenticated user has permission to perform this transaction.

Step 4: Storage

Because a transaction is being defined as an interaction between the user and the application that results in a change to the data stored in the database, the data must be stored. In many cases, a transaction requires that new data be written to the database. However, transactions might only change existing data. In either case, applications may need to ensure that the stored data is not changed, destroyed, or viewed by malicious or unauthorized users. These attacks can often be prevented by a strong access control mechanism and a good backup plan.

Prevention versus Proof

In the previous explanation, there is an element of transaction security that is missing. At the acceptance stage (Step 3), one knows:

- That all the required transaction data is entered in an acceptable format (validation)
- That the data has not been altered during transmission (integrity)
- That no one has viewed the data during transmission (privacy)
- The identity of the user performing the transaction (authentication)
- That the user has permission to perform the transaction (authorization)

It seems like all the major security requirements have been met. The problem is that one only knows these things during the very brief period of time when the transaction is executed. Once the transaction is complete, this knowledge vanishes and cannot be reestablished because it cannot be stored along with the transaction data. However, digital signatures allow some of this knowledge to be captured and stored.

Digital signatures do not protect data in the same way that other cryptographic techniques do. Digital signatures do not hide data from unauthorized viewers. This is provided by data encryption. Digital signatures cannot prevent data from being modified by external hackers or malicious "insiders." This is provided by authentication and access control. Digital signatures simply allow an application to prove two things about the data they "protect":

1. *Integrity*: the data has not been modified since it was signed.
2. *Origin*: the identity of the signer can be cryptographically proven.

There is a significant difference between *preventing* changes to application data and being able to *prove* that the data has not been changed. This may seem like a fine line, but how does one *prove* that one's access control mechanisms have not been compromised? It is much easier to prove that a security violation has occurred than it is to prove that one has not occurred. If attempts to defraud an organization are detected, then the hacker has not done a good enough job.

If the transaction data is digitally signed, applications that rely on that data can prove that it has not changed and that it came from an authorized user. So, although digital signatures cannot prevent fraud from being attempted, they can prevent attempted fraud from succeeding by giving applications the ability to detect fraudulent transactions.

The digital signature itself is a separate piece of data that must be stored with the transaction to facilitate this proof. The fact that digital signature impacts the data storage requirements of the application is another reason why digital signature functionality requires a tighter integration with the application than other security technologies.

Paperless Business Processes

Exhibit 99.2 shows how digital signatures are typically used to implement a paperless process. In each step, the users are using an application that allows them to view and modify data that is stored in a central database. Note that each time a "document" is created or modified within the application, it is digitally signed. Each time that data is used, its signature is verified. This allows the relying user to be confident that the data in the database is genuine and was originated by an authorized user. The application automatically performs the signing and verifying whenever a document is stored or retrieved from the database. This enforces the security policy and prevents users from inadvertently skipping these steps. Because the application must know when to sign documents, when to verify them, and what to do when either of these operations fail, digital signature must be an integral part of the application's workflow logic.

Databases Are Different

Thus far, this chapter has discussed why digital signature technology is different from other security technologies. Relational database applications also have some very unique qualities. These unique qualities require a unique approach to digital signature integration.

What Is a Document?

Digitally signed "transactions" were discussed previously. Often, the term "document" is used to denote the data that is signed (see Exhibit 99.3). Each type of digital signature solution seems to define a document differently. For example, e-mail security products define a document as an e-mail and its attachments. There are security

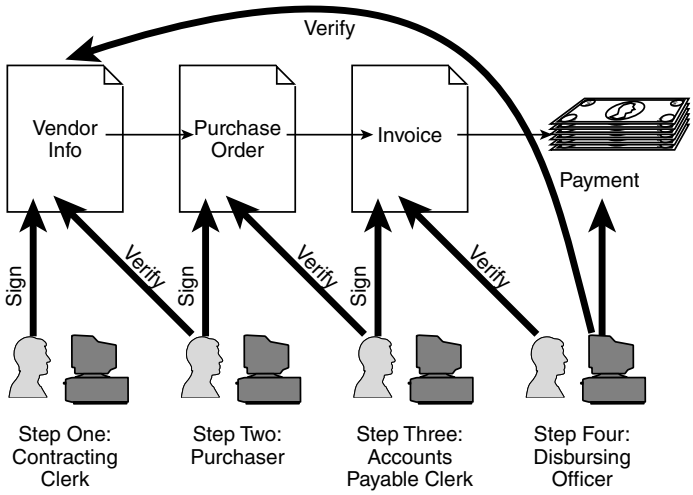


EXHIBIT 99.2. A typical paperless business process.

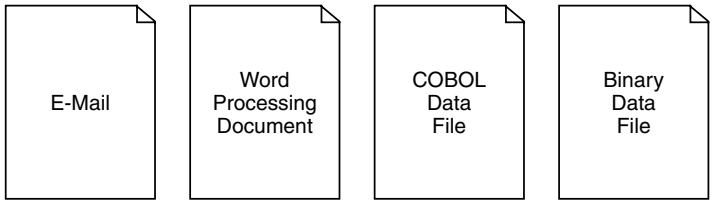


EXHIBIT 99.3 Types of documents.

products that digitally sign word processing documents or spreadsheets. Other products digitally sign any type of file. Note in each of these examples that although a document may internally contain many discrete data elements, the document as a whole can be represented as a contiguous set of bytes.

Relational databases store their data much differently. Databases store structured data as opposed to unstructured data. This means that all of the data elements that compose a document must be known in advance before the first document is created. Databases use a concept called normalization, which allows large amounts of structured data to be stored and searched very efficiently. The data in a document is stored in tables. Tables are composed of rows and columns. The columns define the name (e.g., "PRODUCT_NAME," "INVOICE_NUMBER," or "PURCHASE_DATE") and type (e.g., CHARACTER, NUMBER, and DATE, respectively) of each data element. A row in a table, called a "record," contains the actual data values for each column in the table.

Here, a "document" is defined as the data in one or more rows from one or more columns of one or more tables in a relational database. That is, a document may span multiple database tables and may include only selected columns from those tables and may encompass more than one row per table. This sounds complex and it can be very complex. Databases are designed to efficiently handle large amounts of data that is related in complex ways.

Exhibit 99.4 shows a document in a format that makes sense to people. It is a very simplified purchase order from Gradkell Systems, Inc., to a company named LLED Computer Corporation. A purchase order is usually identified by a purchase order number. This is purchase order #123. It has four line items. Each line item has a quantity, description, and amount. The Purchase Order also has a total amount. Exhibit 99.5 represents how purchase order documents might be stored in a database.

Note that not all columns shown in Exhibit 99.5 are displayed in Exhibit 99.4. This is important because database applications may contain data that is used internally by that application but is not important to the business process. Examples of such data are internal flags that mark a document's position in a workflow (e.g., it has been entered, but is pending approval). It is not usually necessary to sign this type of data because it is

PURCHASE ORDER **#123**

TO: LLED Computer Corporation
From: Gradkell Systems, Inc.
4910 University Place

1	4 Processor 800 MHz Pentium III PowerEdge Server w/Red Hat Linux	$4,750.00
4	512 MB PC-100 DIMM Memory	$250.00
1	SCSI RAID Controller	$1,750.00
3	18 GB 10,000 RPM SCSI Disk Drive	$1,250.00
	Total:	$8,000.00

EXHIBIT 99.4 A database document printed or displayed by an application.

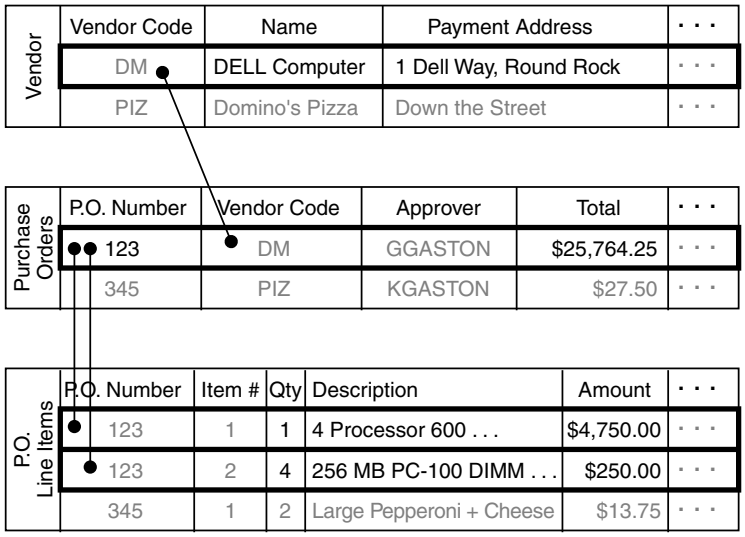


EXHIBIT 99.5 A database document stored in the database. Highlighted rows pertain to Purchase Order #123.

not really part of the document. This data is only used to move the document through a process. If it is signed, the signature will be invalidated when the data changes. Thus, it is important to be able to choose which columns to include in the signature rather than having to sign the entire row.

Note that the data that pertains to Purchase Order #123 is not a contiguous set of bytes. It is intermingled with other purchase orders (e.g., #345, a pizza order). Because digital signature algorithms operate on a contiguous set of bytes, the data must be retrieved from the database and formatted into a contiguous string of characters. This must be done exactly the same way each time. The result must be bit for bit the same every time or the signature will not verify. This is because the digital signature operation is performed on a block of data. At the level in the process where the cryptography is applied, the contents of the data have no meaning. The signing process only sees the data as an ordered collection of bits. The signature verification process simply answers the questions, "Is this the data that was signed?" and "Was it signed by the specified user?"

The exactness with which data must be represented presents some special problems. Databases store numeric and date values in a special way and usually have a default format that is used to display these values. For example, if a date value was signed in the form "11:30 PM on 10 May 1999," but was verified in the form "1999-05-10 23:30:00," the signature will not verify because the data was changed. Actually, only the representation of the data has changed, but that representation was not bit for bit the same as when it was signed. The

same is true of numeric data. The real number 47502.5 can also be represented as "$47,502.50." This becomes an issue when the default format used by the database to represent numeric and date values can be changed by a database administrator. These problems can be avoided if the format of the data is explicitly specified when the data is retrieved from the database.

Integration Approaches: Why Is Application Integration So Problematic?

When adding security features to applications, digital signature is fundamentally different from other security techniques. There are several reasons for this:

- Applications must trigger the signing and verification of documents at the appropriate points in the business process.
- Applications must be able to reject documents or stop processes when signature verification indicates that data has been altered since it was signed.
- The digital signature itself is an additional piece of information that must be stored by the application so that data integrity and non-repudiation can be proven at a later date.

The additional application logic and data storage requirements required to correctly process digital signatures means that digital signature functionality usually cannot be added to applications in a completely transparent manner.

Integration Using Low-Level Cryptographic Toolkits

The nuts and bolts of public key cryptography and PKI are extremely complex. The underlying cryptographic algorithms involve advanced mathematics and absolutely must be implemented correctly. The data formats used to encode data (usually ASN.1, abstract syntax notation) are very complex and require extensive low-level programming experience and a high degree of familiarity with ISO and ANSI standards. The logic associated with building and validating certificate chains presents a substantial learning curve. Fortunately, there are cryptographic toolkits that handle much of this low-level processing.

However, cryptographic toolkits only go so far. Developers must still have a high level of familiarity with the data structures and algorithms used in digital signing and verifying. Most cryptographic toolkits assume that developers are using the C or C++ programming languages. Even when using toolkits such as these, the lack of a comprehensive understanding of what is going on under the hood can result in disastrous security problems.

In addition to security problems, there are a host of other issues that have prevented organizations from taking this approach to application security integration. One reason is high risk. An organization may have plenty of application developers who are proficient in environments such as Visual Basic, Power Builder, Oracle Forms, ColdFusion, JSP, ASP, etc. However, they often do not have very many developers who can be devoted to the task of learning C or C++, PKI programming, and low-level cryptographic toolkits. Even if an organization does have a wealth of "system-level" developers, what are they going to do in six months when the digital signature feature is 90 percent complete and the developer leaves the company? The cost of the integration and maintenance must be weighed against the cost of available third-party solutions that do not require a learning curve that is so steep.

In many cases, "enterprise" databases have several "front ends" to the same data. Data may originate from a Web-based application and be processed internally by an application written in Visual Basic. Often, digital signature integration projects that use low-level toolkits result in a solution that is specific to one application or to one development environment. If the digital signature system only works in the Web interface, other applications may have no way of proving that no one has tampered with the data.

Development Environments with Digital Signature Built In

An alternative approach to using low-level cryptographic toolkits is to completely rewrite the application using tools that have digital signature built in. For new systems, this can work very well. For example, some electronic

forms products have digital signature capabilities built in. These products perform very well when used to directly replace a paper system. The electronic forms can be made to look almost exactly like the paper forms, but do not have to be printed for signature purposes. Many of the packages also integrate with relational databases. They can use the database for both retrieval and storage of form data and they can use the database for form storage. However, these products are not general-purpose database front ends. Some products require their own database structure. Others have limited ability to integrate with existing database structures. They also store a copy of the data within the electronic form itself. So, a database front end comes with some storage, and thus performance, overhead. Electronic forms products usually have their own development environments and macro languages. This means that converting an existing application to use digitally signed "electronic forms" usually amounts to a complete rewrite.

When it comes to digital signature, the electronic forms products work well as long as one is using the electronic form software to access the database. This is because the digital signature is stored within the electronic form itself. If, for example, a Visual Basic application was written that relied on the data in the database, the digital signature could not be verified. Even if the electronic form product included a programming interface that allowed the digital signature to be verified, the signature would be verified using the copy of the data stored in the electronic form, not the copy stored in the database. This is a very serious problem because the Visual Basic application is making decisions based on the data in the database, not the data stored in the electronic form. The verification of electronic form signature could succeed even if the data in the database was altered.

So, development environments that include digital signature functionality usually come with some serious limitations when applied to relational databases. These limitations stem from the fact that they are not designed to be general-purpose database application development tools. They often do not use the database as their primary storage medium, but offer database support as an optional or auxiliary feature. Their digital signature features are not designed for use in other types of applications. These types of digital signature-enabled tools are "development environment-centric" instead of "data-centric."

A Generic Approach to Digital Signature in Relational Databases

As mentioned, the current approach to securing database applications is to build a virtual "wall" around the database server. This wall is composed of network firewalls, encryption, strong authentication and authorizations, intrusion detection, etc. This works well and is complexly application independent. However, this strategy works at the database server level and falls short of providing verifiable data integrity and non-repudiation at the transaction (or "document") level. Digital signatures are the next step in application security, but digital signature technology is different because it requires a certain amount of application integration. To get to this next step, one needs an application-independent system of digitally signing data stored in relational databases that requires as little application integration as possible.

Basic Requirements for Digital Signature Integration into Database Applications

The following chapter subsections describe basic design goals for a generic database signing system.

No PKI Knowledge Required for Application Developers

Application developers should not have to become digital signature experts. Ideally, they should not even need to understand what a digital signature is, other than that it is an operation that is performed on a certain document at a certain place in the business process. There are five application-specific items that a generic database signature system cannot determine:

1. What type of operation needs to be performed (e.g., signing or verification)
2. What type of document is being signed or verified (e.g., a purchase request, an invoice, a time card, a leave request, a 401k participation form, etc.)
3. Which specific document is being signed or verified (i.e., the "primary key" values that uniquely identify a single document)

4. When in the business process to perform digital signing or verification
5. What to do if an error occurs during signing or verification

All of these items are known by the application developer and are similar to the types of information required by other operations in the application. For example, an application developer must know that "purchase request #123 needs to be signed when the user presses the Submit button." Of course, the actual process is much more complex, but the application developer does not need to know the other details, such as which columns in which tables are signed or where the signature data is stored.

Does Not Require Modification to the Existing Database Structure

If the digital signature system is to be application independent, it should not directly rely on the database structure of a certain application. Adding new tables should not be problem, however.

Allows the Data that Is Signed to Be Specified

Because databases do not store their data as contiguous sets of bytes, the data items that compose a document or transaction must be gathered from the database. The data that is signed must be exactly the same when it is verified as when it was signed. Because one wants this system to be very easy to integrate, one does not want to burden application developer, with this task. And because the digital signature will be performing the data-gathering step, it must allow the data (tables and columns) to be specified. This specification should include information that defines how each data item is to be formatted (e.g., "1:00 PM" or "13:00"). The specification should also be able to represent the "primary keys" of the document and the complex ways that the underlying tables are related to each other.

Scalable and Does Not Introduce a Single Point of Failure

The database server and the application server are all required by the application. The PKI adds a directory server. The digital signature system should not introduce any additional servers that could become a bottleneck or cause application processing to stop.

Signature Storage Overhead Should Be as Small as Possible

Database environments offer great advantages when it comes to the efficient storage of data. The de facto standard format for digital signature storage is PKCS #7, the cryptographic message syntax standard. This standard defines a data structure for cryptographic messages such as signed documents.

Most of the fields are optional, but a typical signed data message includes the signer's certificate, the other CA certificates in the "chain," and a copy of the data that was signed. Essentially, a PKCS #7 signed data message is a large "denormalized" chunk of binary data. Because the database is a central data repository that is shared by the signer and the verifier, the certificates and the data do not need to be stored with each signed document. And because this data is being stored in a database, it can be "normalized." The certificates can be stored only once and linked to the signed document via database relationships. A single certificate is about 600 to 1000 bytes in size. A typical PKCS #7 message contains about three certificates. The data portion, which is of indeterminate length, can be also removed from the PKCS #7 message because the data is already stored in the database and does not need to be stored again. As Exhibit 99.6 shows, the normalization of the signature information greatly reduces the amount of signature storage overhead required by the digital signature system. The "optimized" PKCS #7 is about 300 bytes long versus over 3000 bytes (assuming 1024 bytes of data) for the typical case. Storing less data per document also improves performance because less data has to traverse slow network connections.

Abstracting the Digital Signature Process

Digital signature integration can be viewed as "gluing" digital signature functionality onto an existing application. The actual cryptographic operations and interaction with PKI components are performed by low-level cryptographic toolkits. The "glue" is a program library that knows how to interact with both the database and the cryptographic toolkit.

In Exhibit 99.7, the cryptographic toolkit only knows how to sign raw data. It does not know how to gather it from the database or how to store signature information in the database. The database signing logic knows how to retrieve the purchase request data from the database and how to use the cryptographic toolkit to sign the data. It also handles formatting the signature data in a way that is optimal for storage in the relational database environment.

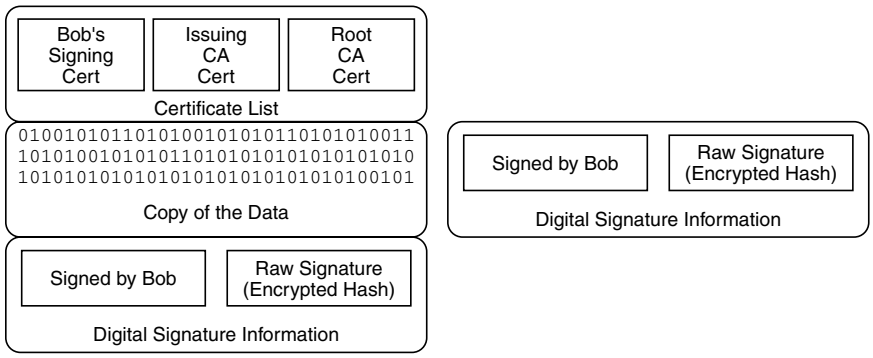


EXHIBIT 99.6 A typical PKCS #7 signed data message vs. one optimized for storage in a database.

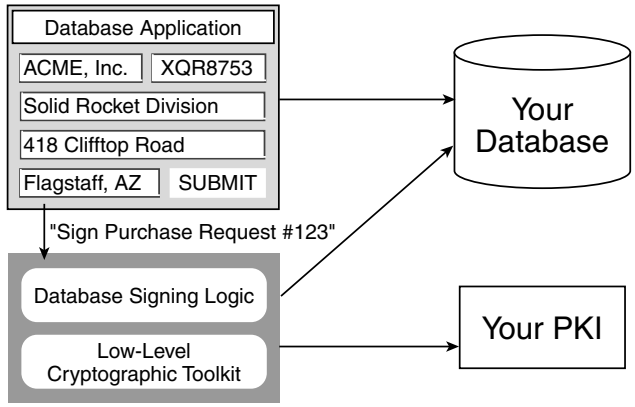


EXHIBIT 99.7 The process of signing a database "document" is standardized and removed from the application logic.

Essentially, the process of digitally signing data in a database is standardized and abstracted from the application so that the application developer does not have to know anything about it. The developer provides just enough information to get the process started. The rest is handled automatically.

Summary

This chapter has discussed some of the unique qualities of both digital signatures and relational databases. Digital signatures are different because they require that data be stored to support signature verification. Relational databases are different because they store data in a very unique way. These two differences work together to make integrating digital signatures into relational database applications a complex and tedious task. The cost and risk of this crucial integration step have hindered the use of digital signatures in many applications. Until recently, there were no digital signature products specifically designed for the database environment. Products such as DBsign from Gradkell Systems, Inc. are now available to vastly simplify the integration of digital signature security into relational database applications. Such products leverage the cryptographic and security expertise of specially trained third-party developers to drastically reduce the cost and risk associated with trying to tackle complex, highly technical integration projects in-house. For more information about DBsign or Gradkell Systems, visit their Web site at www.gradkell.com.

100

Security and Privacy for Data Warehouses: Opportunity or Threat?

David Bonewell, Karen Gibbs, and Adriaan Veldhuisen

How will a company address security and privacy concerns with its customers in an ever-changing environment of increasing public concern for how personal information is collected, used, and distributed by commercial organizations? As consumers become accustomed to defining and deciding how their personal information should be used, they will likely expect their privacy preferences to be respected in *all* forms of interactions.

A growing portion of the concern about privacy invasion surrounds data mining and both its perceived and real threats to personal privacy. Recent events demonstrate how various representatives of the public worldwide are demanding protection against abuse of personal information by organizations using data mining techniques on their warehouse databases. The European Union (EU) has already passed legislation protecting personal privacy. Similar legislative and regulatory privacy protection considerations exist in other countries, including Australia, Canada, New Zealand, Hong Kong, and the Czech Republic, and more have already begun to follow. The U.S. government is encouraging American companies to follow voluntary compliance, reinforced by the Federal Communications Commission (FCC), Federal Trade Commission (FTC), and other regulatory bodies.

A strategy for addressing privacy concerns is to develop and execute sound practices and processes with the highest respect for individual privacy. To effect this, an organization must have the tools and infrastructure that will allow it to comply with regulatory constraints while continuing to gain business advantage with the information it needs to collect and use.

This chapter first describes the business problem concerning privacy laws, rules, and regulations. Realistic business scenarios expose typical privacy-related business requirements from consumer, national, sector, and industry viewpoints that affect system architecture and technology decisions. Business requirements for enabling consumer privacy are illuminated during this discussion. The chapter then illustrates the technical problem through various architectural function perspectives. In summary, this chapter documents how security and privacy requirements impact both business and technical architectural systems across and within a data warehouse.

Problem Description for Enabling Privacy

Data warehousing is a strategic imperative for many companies. Unless adequate measures are taken to protect personal data today, there will be resistance to data mining as a technology in the future. Ignoring security and privacy in a data warehouse will, in particular, undermine an organization's data warehouse strategy if such resistance becomes widespread.

Furthermore, several regulatory activities are occurring worldwide. The European Union (EU) Directives 95/46/EC[1] and 97/66/EC[2] are now in effect and require privacy legislation throughout the EU. The Federal Communications Commission (FCC) interpretations of Section 222 of the Telecommunications Act places legal requirements on telecommunications companies regarding the use of Customer Proprietary Network

EXHIBIT 100.1 Opportunities and Threats as They Affect Business Drivers

	Opportunities	Threats
Use of personal information	Enhanced public trust through appropriate use	Public concern about misuse; potential for costs to an individual resulting from abuses
Legislation, regulation	Potential for customers' compliance useful as competitive weapon for improving company image and eliminating costs associated with litigation; help to stay focused on core business	Fines, suits, and a general inability to do business, potentially causing operational changes or new hardware/software purchases leading to decreased value for shareholders; reduced focus on core business
Economic impact	Data warehouse investments leading to increased value of collected data by removing useless or low-value data, decreasing marketing costs, and improving consumer satisfaction; increased value of information collection	Data warehouse investments in jeopardy, possibly leading to decreased value of collected information and increased costs associated with information removal

Information (CPNI). Movement of citizen, employee, and consumer data between countries is also a significant privacy issue.

A company's response should be to take the necessary actions to be perceived as a leader in privacy protection by adding capabilities that help the company conform to the FTC, FCC, and EU directives, regulations, initiatives, and other emerging legislation.

Privacy protection capabilities will help an organization:

- Determine which data is personally identifiable in a data warehouse
- Identify and modify personally identifiable data
- Utilize data mining techniques that respect consent choices (opt-in and opt-out) of consumers

Privacy: Opportunity or Threat to Business Drivers

Companies manage key business drivers through initiatives that are common to most industries in order to achieve their success. Two of these related business drivers are customer acquisition and customer retention, often accomplished by taking actions to maintain customer loyalty and improve customer service. Another of these business drivers is wallet share, usually achieved through endeavors to grow the customer's share of the market segment addressed. A fourth key driver is total cost of ownership (TCO), generally realized through measures to reduce expenses or improve efficiencies throughout the business' processes.

Exhibit 100.1 captures some of the possible opportunities and potential threats across all industries that arise from privacy-related concerns and issues as they affect these key business drivers.

Enabling consumer privacy imposes both business and technical problems for many companies. Primary concentration on the business problem allows for clarification of key business issues prior to technology and development decisions; however, it is valuable to decompose each perspective of the privacy problem into its constituent parts for further examination. Separating the problem into business and technical discussions focuses attention on the key issues pertinent to each of these two areas and exposes hidden and false assumptions during analysis. Before proceeding to analyze the business and technical perspectives of the privacy problem, it is necessary to discern privacy from security and confidentiality, as well as to clearly understand the different sources for the rules that guide privacy policies. The next two subsections briefly explain these clarifications.

Clarification of Terms

It is important to understand the meaning of the terms "privacy," "security," and "confidentiality" in order to properly understand the business and technical perspectives of the privacy problem.

Privacy defines an individual's freedom from unauthorized intrusion (into matters considered by the individual to be personal).[3] This definition effectively addresses both the U.S. and European notions as well as legal histories, and applies well to data.

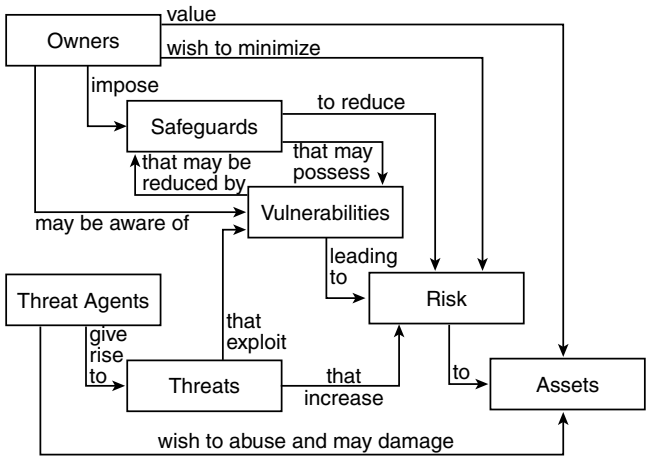


EXHIBIT 100.2 Concepts and relationships (flow of logic) within a security system.

Security defines an attribute of information systems, and includes specific policy-based mechanisms and assurances for protecting the confidentiality and integrity of information, the availability of critical services, and indirectly, privacy.

Confidentiality defines an attribute of information. Confidential information is sensitive or secret information, or information whose unauthorized disclosure could be harmful or prejudicial. Because security is required to ensure privacy and confidentiality of personal information, it must be present throughout business processes in solutions that enable consumer privacy. Exhibit 100.2 diagrams the flow of logic within a security system.

Exhibit 100.2 is taken from Common Criteria ISO 15408 standard specifying the Privacy Class of Common Criteria.[4] It proposes that all security specifications and requirements should come from a general security context. This context states that "security is concerned with the protection of assets from threats, where threats are categorized as the potential for abuse of protected assets." The scope of threat prevention says that all threats should be considered; but in the domain of security, greater attention is given to those threats that are related to malicious or other human activities.

The Common Criteria framework follows a logical progression, wherein first a security environment is described, and then security objectives are determined based on the indicated security environment. More details dealing with security environment characteristics, security objectives, security services requirements and security functional requirements concerned with information protection are briefly discussed in Exhibit 100.3.

The remainder of this chapter assumes that a company has implemented security systems that assure privacy and confidentiality of personal information appropriate for the industry environments in which it does business. Other than identifying security as an ongoing requirement for privacy, no further detail will be explored. It can be further stated that one can have security in a data warehouse and not have privacy; but one cannot have privacy without security in this environment.

Clarification of Rules

Rules for guiding privacy policies are derived from a number of different sources, including national governmental authorities, corporations and market-sector organizations, and consumers.

Government rules are primarily defined and enforced by legislative and regulatory bodies and vary by government entities. An example is the European Directive passed by the European Union.[1,2]

Corporate and sector rules can be defined by businesses that constitute specific market segments or by government agencies covering these markets. An example is the Telecommunications Reform Act of 1995 governing customer proprietary network information.

Consumer rules are defined by private individuals. An example is the preference to receive marketing advertisements via hard-copy mail versus telephone. Another example is the preference to have personal data

EXHIBIT 100.3 Security Requirements (ISO 15408)/Common Evaluation Criteria (CEM)

Security Environment

- **Assumptions:** Descriptions of assumption elements are needed to specify the security aspects of the customer's environment. This should include information about intended usage of applications, potential asset value, possible limitations for use, as well as information about environment use such as physical, personnel, and connectivity aspects.
- **Threats:** These elements are characterized in terms of a threat agent, a presumed attack method, possible vulnerabilities, and protected asset identification.
- **Organizational Security Policies:** These elements are any and all laws, organization security policies, customs, and IT processes determined relevant to the defined environment.

If security objectives are derived from only threats and assumptions, then the description of the organization security polices can be omitted.

Security Objectives

The security objectives address the identified threats, the customer's organizational policies, and environmental assumptions. The intent of determining security objectives is to address all of the security concerns based on a process incorporating engineering judgment, security policy, economic factors, and risk acceptance decisions.

- **Legitimate Use:** Ensuring that information is not used by unauthorized persons or in unauthorized ways.
- **Confidentiality:** Ensuring that information is not disclosed or revealed to unauthorized persons.
- **Data Integrity:** Ensuring consistency, and preventing the unauthorized creation, alteration, and/or deletion of data.
- **Availability:** Ensuring that data and services are accessible when they are needed.

Security Services Requirements

Meeting security objectives requires a set of security services, or mechanisms. Security services fall into six categories:

1. **Authentication:** Services that assure that the user or system is who that person (or system entity) purports to be. Authentication services can be implemented using passwords, tokens, biometrics (e.g., fingerprint readers), and encryption.
2. **Access Control:** Services that assure that people, computer systems, and processes can use only those resources (e.g., files, directories, computers, networks) that they are authorized to use and only for the purposes for which they are authorized. Access control mechanisms can be identity based (e.g., UNIX protection bits, access control lists), label-based (also known as mandatory access controls), or role-based (implemented as a combination of the above, plus system privileges). Access control plays an important role in protecting against illegitimate use and in providing confidentiality and integrity protection.
3. **Confidentiality:** Services that protect sensitive and private information from unauthorized disclosure. Confidentiality services are generally implemented using encryption.
4. **Integrity:** Services that assure that data, computer programs, and system resources are as they are expected to be and that they cannot be modified by unauthorized people, software, or computer equipment. Mechanisms for implementing data integrity include cyclic redundancy checks and checksums, and encryption. Mechanisms for assuring system integrity include physical protection, virus-protection software, secure initialization mechanisms, and configuration control.
5. **Attribution:** Services that assure actions performed on a system are attributable to the entities performing them, and that neither individuals nor systems are able to repudiate their actions. Mechanisms providing attribution include audits, encryption, and digital signatures.
6. **Availability:** Services that assure that systems, applications, and data are available when they are needed. Considerable efforts must be made to safeguard data and critical system services, ensuring that correct and complete information and IT services to deliver and process that information are available to authorized individuals. A critical requirement of any privacy protection schema is to ensure that critical data and services are available at all times. Mechanisms for providing availability include fault-resilient computers, virus protection software, and RAID (Redundant Array of Inexpensive Disks) storage.

Security Functional Requirements

The Common Criteria v2.0 identifies four families of terms that are concerned with the protection against discovery and misuse of information.

1. **Anonymity** ensures that a user may use a resource or service without disclosing the user's identity. The requirements for anonymity provide protection of the user identity. Anonymity is not intended to protect the subject identity.
2. **Pseudonymity** ensures that a user may use a resource or service without disclosing its user identity, but can still be accountable for that use.
3. **Unlinkability** ensures that a user may make multiple uses of resources or services without others being able to link these uses together.
4. **Unobservability** ensures that a user may use a resource or service without others, especially third parties, being able to observe that the resource or service is being used.

not sold to third parties. Allowing individuals to specify personal privacy preferences, or rules, maintains the integrity and credibility of the rules for each consumer.

The Business Problem

The privacy problem described in the previous sections can be summarized into the following, simple business problem statement:

> Companies need to be able to market to their customers while respecting their customers' expectations as well as domestic and international laws regarding how personal information is collected and used.

This section examines the problem of enabling privacy from the business perspective by exploring a business scenario. Business requirements that are discovered during scenario exercises are captured and used to guide system architecture and technology decisions. Additional business requirements for privacy awareness and sensitivity derive from emerging and existing legislation and public pressures. Clarification of the ensuing privacy business requirements will assist in creating an architecture model illustrating the impacts of enabling consumer privacy.

Business Environment for Enabling Privacy

A business scenario includes a short description of the business environment, the actors involved in the scenario, and the business interactions between the actors. For companies, Exhibit 100.4 illustrates the business environment for enabling consumer privacy.

The left side of Exhibit 100.4 displays several choices for how and where a consumer may prefer to conduct interactions with a company. Examples shown include using hard-copy mail, by telephone, in person, through some special-purpose kiosk, or from a PC possibly via the Internet. Not explicitly shown are those interactions that may be conducted by third parties, such as automated applications performing automated decisions or intelligent agents. Interactions may or may not result in one or more transactions (actual exchanges for goods and services) instituting a relationship between a consumer and a company.

The right side of Exhibit 100.4 introduces sources from which a company obtains the business rules that guide company privacy policies. Legislative requirements for ensuring consumer privacy differ among government jurisdictions. Industry sector and corporate rules for consumer privacy likewise differ for various regulated and nonregulated markets. Finally, consumer privacy preferences can be incorporated, depending on company policies.

The center of Exhibit 100.4 focuses on the data warehouse as both the storage site for consumer personal data and the optimal position from which a company can ensure and enforce consumer privacy preferences.

Business Scenario for Enabling Privacy

Exhibit 100.5 reveals a more thorough examination of the business interactions involved in this business scenario. The example assumes that privacy policies have been:

- Established by government, sector, and consumer rules
- Incorporated into database information structure, design, and metadata services
- Presented to the consumer at some point prior to the start of the interaction

Consumer Interactions

It is commonly accepted that an implied contract is established between a consumer and a transaction provider when that consumer voluntarily and knowingly engages in interactions that may ultimately result in transactions with that transaction provider. The contract implies agreement:

- By the consumer to supply personal data required for that transaction
- By the transaction provider to use, maintain, and store this data in some form, for some length of time, for the purpose of fulfilling the contract

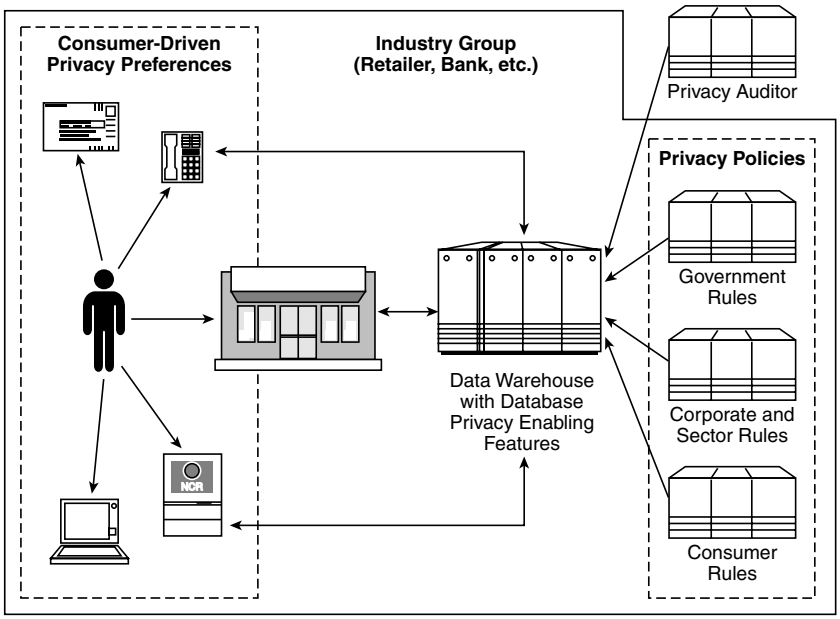


EXHIBIT 100.4 Business environment for enabling consumer privacy.

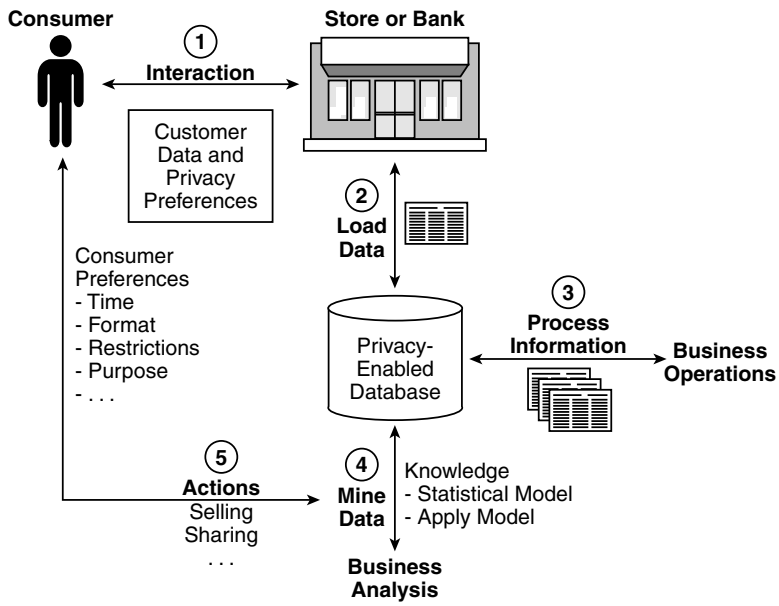


EXHIBIT 100.5 Business interactions involved for enabling consumer privacy.

Consumers are willing to share additional personal data (outside the required purpose) in relationships where the business is *trusted* and where there is an identified need or mutual benefit. The amount and type of data shared reflect explicit and implied consumer preferences, as well as business requirements.

Loading Data

Businesses need to examine the collection of consumer interactions and transactions in order to determine "what happened." This can be done from legal, business, monetary, fiscal, competitive, and other aspects that are necessary for legitimate business functions. Historically, typical storeowners and bankers "remembered" their customers' behaviors and preferences and modified ensuing interactions accordingly. Likewise, larger companies, aided by modern tools such as data warehouses, will be able to "remember" their customers' behaviors and preferences through the history of collected interactions and transactions that have been loaded into their databases.

Processing Information

Once businesses determine "what happened," the next logical step is to learn "why it happened." Numerous tools are available for businesses to use in processing interaction and transaction information. These tools help diagnose and visualize patterns in consumer behaviors and preferences that ultimately guide business operations toward greater efficiencies and optimize corporate behaviors to be consistent with company goals and objectives. Consumers are unlikely to object to such uses for their personal data as long as the insights gained for the business do not automatically lead to actions contrary to their privacy preferences.

Mining Data

After ascertaining "what happened" and "why it happened," businesses employ tools and techniques, such as data mining and analytical modeling, in attempts to predict "what will happen." Such analysis considers a business' memory of interactions and transactions, as well as possible additional information obtained from external sources. Businesses are responsible for ensuring that these external information sources are legal and accurate, and that they have the consent of affected consumers if personally identifiable data is involved. Resulting predictive models are applied to consumer records to forecast future behaviors, typically in the areas of consumer acquisition, retention, and growth. These models can also be used in determining business impact expectations affected by credibility, fraud, affluence, and other business conditions.

Taking Actions

The point at which businesses decide to take "actions" based on predictive modeling results is the final step in the business scenario for enabling consumer privacy. No actions should be taken that are in violation of the law or against the preferences of the consumer. Privacy considerations impact business behaviors and may provide either a threat of increased regulation leading to decreased ability to do business, or an opportunity to better understand and respond to consumer preferences, thereby strengthening the relationship.

In summary, it is crucial to examine the metamorphosis that data undergoes throughout business interactions, and where businesses control, store, and process consumer data. Ultimately, only companies decide how privacy will be executed within their businesses. No implementation will prevent businesses from taking actions contrary to the law or to consumer privacy preferences.

Business Requirements for Enabling Privacy

Legislative developments for protection of personal privacy range between rigorous government involvement and self-regulatory approaches. Voluntary guidelines establishing basic principles for data protection were adopted in 1980 by member nations of the Organization for Economic Cooperation and Development (OECD).[5] These guidelines encourage adoption of legislation and practices recognizing the rights of individual citizens with respect to personally identifiable data gathered about them, and defining parameters for what constitutes personally identifiable data.

A great deal of thought has already gone into consolidating privacy provisions specified in the OECD guidelines with the "key elements" of the Online Privacy Alliance[6] and the Articles of the EU Directive[1,2] in order to generate a comprehensive set of privacy requirements. This chapter briefly summarizes six proposed

privacy requirements and explicitly adds two more related requirements, which, when applied to system architectures, help in determining the impacts of privacy interventions on each system.

1. *Notice* Companies should be able to provide easily understood notice to their customers that personal data will be collected, which data will be collected, and how data will be used and disclosed. Notification should include the identities of the data collector and other intended recipients of the data, as well as information about "logic involved in automated processing."[1,2,7]
2. *Choice/Consent* Companies should be able to provide their customers with suitable choices to opt-in or opt-out[8] of specific personal data items for collection, use, and disclosure, consistent with the jurisdictions and requirements the industry environment in which they do business.
3. *Access* Companies should be able to provide assurance to their customers that the personal data they collect, use, and disclose is accurate and up to date. Accessibility includes the means for individuals to review and correct inaccurate or incomplete personal data, as well as the right to erase or "block" access to data not collected in accordance with the rules of local legislation.
4. *Security* Companies should be able to provide assurance to their customers that the personal data they collect, use, and disclose is secure against loss, and against unauthorized access, destruction, alteration, use, or disclosure.
5. *Limitation* Companies should be able to provide assurance to their customers that the collection and use of personal data will be limited to explicit, specified, and legitimate purposes, and that the data will be kept in identifiable form for no longer than necessary to accomplish original purposes.
6. *Accountability* Companies should be able to establish procedures for their customers to seek resolution or redress for possible violations of stated privacy principles and practices. Accountability includes support for enforcement of existing legal and regulatory remedies (country specific) and notification to privacy authorities in each country of intent to collect personal data relating to their subjects.
7. *Traceability* Companies should be able to provide assurance to regulators that all interactions and processing will be traceable and logged in such a way as to allow for internal assessments, as well as assessments by third parties, that demonstrate customer compliance with privacy policies. This is particularly important for those customers desiring compliance with Safe Harbor[9] proposals.
8. *Anonymity/Pseudonymity* Companies should be able to provide assurance to their customers that personal data can be maintained in a state of either anonymity or pseudonymity, as elected by the individual, such that the data cannot be used later to target the individual.

Mapping Requirements to Architectural Components

The business environment and business scenario, explored previously in Exhibits 100.4 and 100.5, depict the relationship between consumers and companies. When viewed architecturally, three components describe the primary areas impacted by enabling consumer privacy:

1. *Privacy presentation* serves as a "window" into consumer interactions and covers consumer, administrative, and operational devices as well as browsers.
2. *Business logic for enabling privacy* covers business interaction activities, transactions, translations, analysis, and management.
3. *Privacy data* covers query, look-up, and other data management activities for data warehouses, as well as for intermediate data stores, either within applications or stored in smaller databases.

The eight privacy business requirements discussed earlier impact these three architectural components as shown by the chart in Exhibit 100.6. The Xs in the chart indicate which requirements for enabling consumer privacy must be met for each architectural component. For example, the requirement for notice must be implemented for both privacy presentation and business logic components, but not for the privacy data component. As stated previously, security is required for any solution that enables consumer privacy; therefore, security considerations must be implemented for each architectural component.

Architecture Model for Enabling Consumer Privacy

Mapping business requirements to architectural components ensures that implementations are guided primarily by business considerations prior to evaluating technical options for those implementations. The architecture model in Exhibit 100.7 illustrates this mapping graphically.

EXHIBIT 100.6 Mapping Business Requirements for Enabling Consumer Privacy to Architectural Components

	Privacy Presentation	Business Logic for Enabling Privacy	Privacy Data
Notice	X	X	
Choice	X	X	X
Access	X	X	X
Security	X	X	X
Limitation		X	X
Accountability		X	
Traceability	X	X	X
Anonymity		X	X

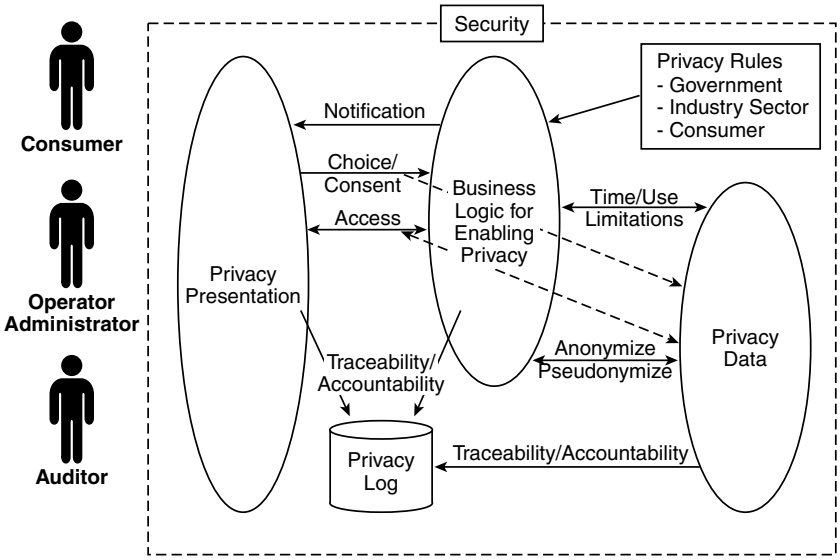


EXHIBIT 100.7 Architecture model for enabling consumer privacy.

The model identifies several different types of users who can interact with a customer's business system, predictably with different types of interfaces, through the privacy presentation component. They include consumers, operators and administrators, and privacy auditors. Users can also be applications and agents operating on behalf of human beings. The model also indicates the various sources for privacy rules impacting the business logic component, that is, government, industry/sector, and consumer. It also illustrates how requirements for security envelop all business processes that are impacted for enabling consumer privacy.

The model shows that both privacy presentation and business logic components will need to contain sub-components that address requirements for notice, choice/consent (which involves data collection), and access (which may or may not involve data correction). It reveals that the requirements for time and use limitations, as well as anonymity/pseudonymity, will need to have sub-components contained in both business logic and privacy data components.

The model further represents that all three architectural components will need to contain sub-components dealing with requirements for traceability, which will likely be required to support requirements for accountability procedures defined by the business.

During interactions, and in addition to sending privacy policy notification, companies should be able to allow consumers to specify:

- Whether or not they can be tracked for purposes beyond the contracted business agreement
- What data they are willing to share beyond that which is required for the contracted business agreement
- Under what circumstances they will share data (loyalty programs) beyond that which is required for the contracted business agreement
- What data, if any, they are willing to have retained or sold

During business operations, companies should be able to allow:

- Consumers to examine their personal data
- Consumers to correct erroneous data
- Consumers to interact anonymously
- Regulators to examine company compliance with protecting personal data

During analysis, companies should be able to comply with:

- Regulations for retention periods
- Regulations for authorized use
- Anonymization rules
- Consumer rules for retaining or selling data

Popular thinking deems that the best place to control privacy is at the point of access; however, the authors maintain that the best place to control privacy is within the data warehouse where the rules for using personally identifiable information can be strictly enforced.

Additional details on the functions required for enabling consumer privacy, and how they map to the architecture model just described, is the focus of the next chapter section.

The Technical Problem

The technical problem of enabling consumer privacy is complicated by customer investments in current technologies, rapid business environmental changes, emerging technologies, and evolving standards. The following technical problem statement captures these concerns:

> Companies need technologies and services that sustain existing and emerging privacy requirements, and that offer flexibility for changes in privacy rules, scalability for growth, and acceptable changes in performance, reliability, availability, and manageability.

This section examines the problem of enabling privacy from various technical perspectives. The business requirements that were revealed during investigation of the business problem are further scrutinized to identify the functions, processes, and technologies necessary to meet the requirements. These business requirements, along with the business environment, influence technology decisions that help formulate the technical requirements impacting the architecture.

Functions Required for Enabling Privacy

Exhibit 100.8 describes functions, along with the types of data, necessary to implement each business requirement for enabling privacy. Current and emerging technologies that apply to these functions are identified, and those that are advocated for this solution are underlined.

Technical Perspectives for Enabling Privacy

Technical perspectives depend on the focus of business objectives and other qualitative attributes, such as function or performance. Different attributes abstract specific details from the business environment with respect to different criteria, thus generating the different system perspectives. Each perspective can independently define the meanings for components, interrelationships, and guidelines, but resulting system perspectives are not independent.

Recognizing the fact that enabling consumer privacy requires changes to existing architectures and not entirely new architectures, each of the technical perspectives discussed below addresses only those specific

aspects that must be considered when applying changes to a system's architecture that enable it for consumer privacy. The next four subsections examine functional, performance, availability/reliability, and OA&M perspectives.

Functional Perspective

The functional perspective exhibits architectural views of processes, data flows, communications, and presentation for each of the components identified in the architecture model. The functions exhibited within the architecture components comprise the architecture building blocks for enabling privacy.

Privacy Presentation Component

Exhibit 100.9 captures the functions necessary within the privacy presentation component to support the business requirements for enabling consumer privacy. Five functional architecture building blocks are defined.

The left-most, vertically oriented building block within the privacy presentation component in Exhibit 100.9 highlights the authentication and authorization functions necessary to fulfill the security requirements. The building block at the bottom of the exhibit highlights functions for tracking activities performed on, or with, personal data and privacy preferences that are necessary to fulfill the traceability and accountability requirements. The three remaining building blocks highlight the functions necessary to fulfill the privacy requirements for privacy policy notification, choice/consent, and access of personal data and privacy preferences.

The following describes the flow of data through the privacy presentation component. An initial communication occurs between some type of "user" (human, agent, or other application) and the appropriate "user" interface to an implementation of the privacy presentation component. The user may or may not have been previously notified regarding the privacy policy through various mechanisms, including hard-copy mail, brochure, electronic mail, HTTP, and others. Once the user is authenticated and authorized to operate within this component, all activities that "get," "move," or "use" personal data (including privacy preferences) are logged and monitored.

The privacy presentation component executes functions that send and receive personal data and privacy preferences between "users" and the component implementing business logic for enabling privacy. It also executes functions that allow these "users" to review and correct personal data and privacy preferences. Such review and correction may occur dynamically in the future; however, it is more likely that, for the present, these functions will be implemented through some type of paper-based, report-and-update mechanism.

For automated systems, privacy preferences can be specified periodically or maintained every time a consumer conducts business. For the latter case, programmable Web agents may be appropriate mechanisms to ease the overhead of specifying and maintaining privacy preferences. The recommended standards for communication among privacy presentation functions are HTTP and P3P (Web-based client position for P3P, personal privacy protection, is the most evolved; however, the types and formats for defined privacy data elements can be extended to other operating environments).

An advocated position for communicating between privacy presentation functions and the functions for implementing business logic enabling privacy are Microsoft's messaging services (i.e., MSMQ), Microsoft's object request broker architecture (i.e., COM/DCOM), or Web-based services (i.e., HTTP, P3P). Industry-specific interfaces will apply on top of COM/DCOM (i.e., DNAfs) for financial.

Business Logic for Enabling Privacy Component

Exhibit 100.10 captures the functions necessary within the business logic component to support the business requirements for enabling consumer privacy. Four functional architecture building blocks are defined. The first three building blocks within the business logic component in Exhibit 100.10 highlight the functions necessary to fulfill the privacy requirements for privacy policy notification, choice/consent, and access of personal data and privacy preferences. Specifically, the functions maintain the privacy policy and enforce privacy rules for the business. The building block at the bottom of the exhibit highlights functions for tracking activities performed on, or with, personal data and privacy preferences necessary to fulfill the traceability and accountability requirements.

The following describes the flow of data through the business logic component for enabling privacy. All activities that "get," "move," or "use" personal data (including privacy preferences) are logged and monitored.

The business logic component executes functions that process requests and responses regarding personal data and privacy preferences between the privacy presentation and privacy data components. As part of processing these requests and responses, the business logic component also executes functions that enforce

EXHIBIT 100.8 Functions Required for Enabling Privacy

	Functions Necessary	Types of Data Needed	Technologies
Notice	•Communicate privacy policy •Include explanations for any "automated processing" •Data usage tracing facility (to track the use of data within the IT system end-to-end)	•Company privacy policy	•Paper-based and Web-based devices and protocols •Specific devices, kiosks •Scripts •Metadata repository (documenting the use of privacy-enabled data)
Choice/consent	•Identify specific data elements that must be displayed, which elements can be changed, and by whom •Present personal preference choice options/current settings •Make and change personal preference settings •Negotiate (option) personal preference settings •Commit/acknowledge personal preference setting changes	•Personal preference choice options •Personal preference current settings •Company privacy policy rules •Privacy metadata •Negotiation rules	•Paper-based and Web-based devices and protocols •Specific devices, kiosks •For interactions involving data warehouse (DW) then metadata standard for privacy is MDIS •For interactions not involving DW, then metadata standard for privacy is P3P •Data collection/update MUI (multimedia user interface) •Scripts •DB access
Access	•Identify specific data elements that must be displayed, which elements can be changed, and by whom •For user-initiated requests: —Authenticate user —Request access to view personal data —Respond to access request •For business-initiated requests: —Present current personal preference settings —Request update to settings •Negotiate (option) or change personal preference settings •Delete all instances of specific and "allowable" elements •Commit and acknowledge personal preference setting changes	•Personal preference current settings •Company privacy policy rules •Negotiation rules	•Web-based devices, protocols, verification mechs (VeriSign) •Specific devices, kiosks •Call centers •Paper reports (OLAP/SQL) •For interactions involving DW, then metadata standard for privacy is MDIS •For interactions not involving DW, then metadata standard for privacy is P3P •Data collection/update MUI (multimedia user interface) •Scripts •DB access (create, delete, update, and delete) •Transaction integrity (to assure accuracy of database updates)

Limitation	•For "use" limitation (what company can do with personal data), enforce use preferences •For "retention" limitation (how long company can use personal data, may not be known), enforce retention preferences	•Company privacy policy rules •Personal preference current settings •Additional collected data	•Application logic assuring "legitimate purposes" are carried out •Business processes handling manual and automated intervention for opting out of automated processing •For interactions involving DW, then "database views" control time/use limits •For interactions not involving DW, then stored procedures control time/use limits •One has potential to develop "privacy state information" to help enforce dynamic temporal changes •Possible application development technology that assures new applications adhere to rules •Possible application execution environment logic to assure legitimate use
Accountability	•For controller or processor of personal data (also requires traceability): —Interrogate systems and make corrections —Non-repudiation capability	•Company privacy policy rules •Personal preference current settings •Controller processor identification •Privacy log repository	•Business procedures •Security technologies (for non-repudiation and logging)
Traceability	•Architecture for managing traceability and verifying requirements •Log event occurrences, alarms, exceptions, etc. •UI to look at logs and reconcile between different data services •Generate reports •"Tracking facility" for privacy adherence/compliance •Enforce logging function and protect logged data •Establish logging of configuration controls	•Company privacy policy rules •Personal preference current settings •Privacy log repository	•Many, depending on chosen architecture for enabling traceability •Application execution environment logging (pre- and post-call logging)
Anonymity/ pseudonymity	•Anonymity (as it applies to usage, takes identifiers away; is NOT reversible) —Block, strip, or screen out personally identifiable data •Pseudonymity (assigns nonidentifiable name to collection of data; is reversible) —Generate pseudonyms with appropriate controls	•Personal preference settings on usage •Personal preference settings on retention	•For interactions involving DW, then "database views" handle anonymity •For interactions not involving DW, then stored procedures handle anonymity •Pseudonym generators

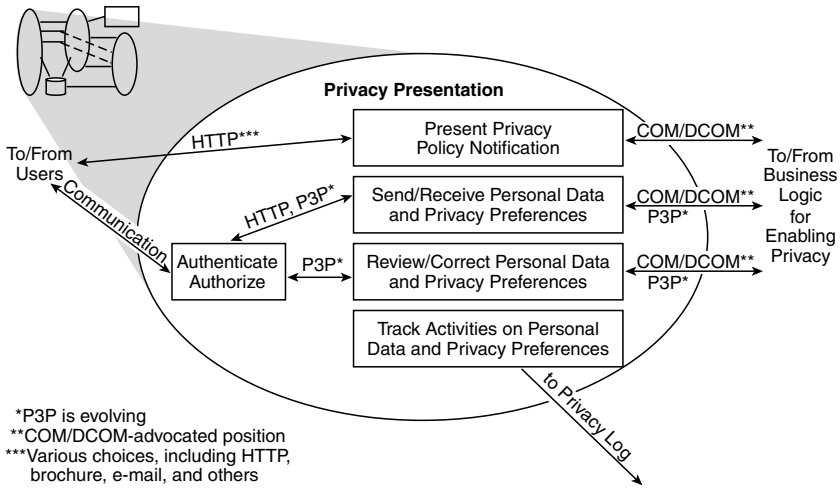


EXHIBIT 100.9 Functions within privacy presentation component for enabling consumer privacy.

privacy rules derived from the business rules and sources for government, industry/sector, and consumer privacy rules.

Where business logic functions are implemented within applications, there are no recommended standards for communication among these business logic functions. Business policies governing operational and analytical applications will likely dictate how information is communicated within these automated systems.

An *advocated* position for communicating between the functions for implementing business logic enabling privacy and privacy data functions are Microsoft's object request broker architecture (i.e., COM/DCOM) or Web-based services (i.e., P3P). The P3P session information passed across these component interfaces is different from that passed across for the privacy presentation component. Those customers with preexisting infrastructures (e.g., proprietary, CORBA, messaging, DB2) for data communication will likely maintain their infrastructures.

Privacy Data Component

Exhibit 100.11 captures the functions necessary within the privacy data component to support the business requirements for enabling consumer privacy. Four functional architecture building blocks are defined.

The left-most, vertically oriented building block within the privacy data component in Exhibit 100.11 highlights the data integrity protection and data access control functions necessary to fulfill security requirements. The building block at the bottom of the exhibit highlights functions for tracking activities performed on, or with, personal data and privacy preferences that are necessary to fulfill the traceability and accountability requirements. The two remaining building blocks highlight the functions necessary to fulfill privacy requirements for choice/consent and access of personal data and privacy preferences, time/use limitations, and anonymity/pseudonymity.

The following describes the flow of data through the privacy data component. All activities that "get," "move," or "use" personal data (including privacy preferences) are logged and monitored. The privacy data component executes functions that verify the integrity and access permissions for data requests received from the business logic component.

The privacy data component also executes functions that filter the data according to previously established privacy preferences prior to accessing personal data or responding back to the business logic component. Furthermore, the privacy data component executes functions providing privacy metadata services for personal data stored either in databases or within specific applications.

Where privacy data functions are implemented within nondatabase applications, there are no recommended standards for communication among these privacy data functions. Business policies governing operational and analytical applications will likely dictate how information is communicated within these automated systems. Where privacy data functions are implemented within database system applications, the recommended standards for communication among functions are SQL, XML, MDIS and ODBC, as well as OLE/DB and OLE/DBO.

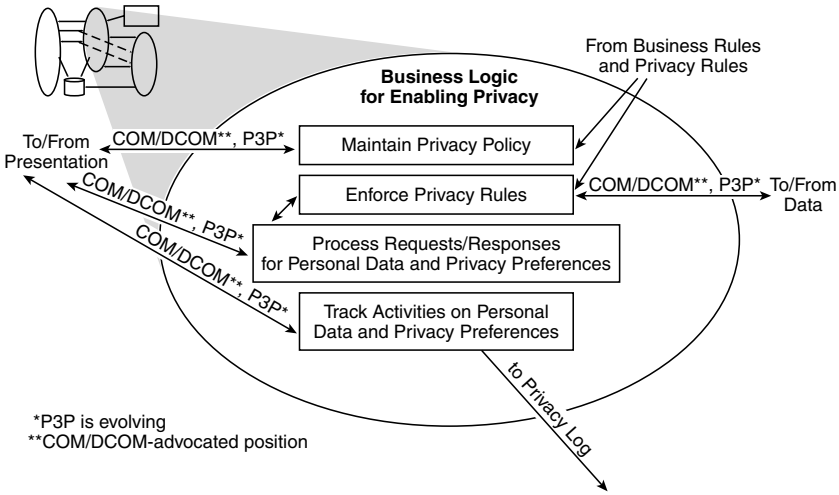


EXHIBIT 100.10 Functions within business logic for enabling consumer privacy component.

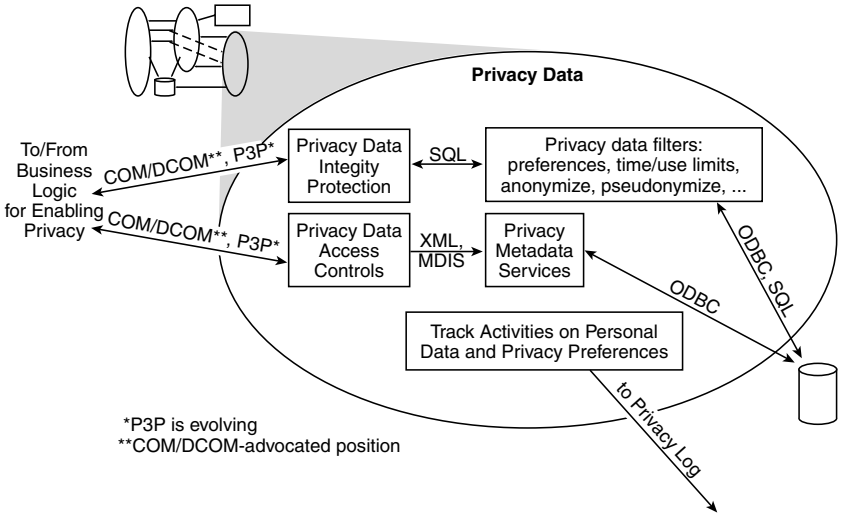


EXHIBIT 100.11 Functions within the privacy data component for enabling consumer privacy.

Performance Perspective

The performance perspective addresses performance implications to the architecture as a result of enabling consumer privacy. As with any system, performance is balanced against features and functions. A trade-off is established between required features and functions, and acceptable performance.

Within the privacy presentation component depicted in Exhibit 100.10, the functions most likely to affect performance are those implementing requirements for choice/consent and access (whether real-time or delayed), traceability (depending on the level of logging), and security. The functions implementing notice are expected to affect performance to a lesser degree.

Within the business logic component depicted in Exhibit 100.11, the functions most likely to affect performance are those implementing requirements for choice/consent and access (related to enforcement of the privacy rules), and traceability. Functions implementing maintenance of the privacy rules are expected to affect performance to a lesser degree.

Within the privacy data component depicted in Exhibit 100.11, the functions most likely to affect performance are those implementing requirements for access, time/use limitations, traceability, and security. Performance thus depends on where and how personal data is stored and maintained. For implementations using teradata data warehouses, performance is minimally affected because requirements for enabling consumer privacy are accommodated by the existing data warehouse design. Other data warehouses, intermediate data stores, and types of databases, as well as other types of applications maintaining personal data, will likely have performance degradations due to the additional functions imposed by privacy requirements.

There are also likely to be performance implications based on implementation choices for communications between the three main architectural components. The emerging World Wide Web Consortium (W3C) standard for P3P may have performance implications on server interactions; however, despite its current popularity and because this standard is evolving, these implications are unknown.

Availability/Reliability Perspective

The availability/reliability perspective is concerned with impacts to the availability and reliability of solutions based on the architecture resulting from enabling for consumer privacy. Availability focuses on the time between system failures. Reliability focuses on the frequency with which a system fails. As with any system, acceptable levels of availability and reliability are determined by the requirements for the industry's operating environment.

The question for each industry to ask itself is whether or not privacy is such an integral part of the system that the whole system is down when privacy-related elements, such as privacy log connections, are unavailable. Trade-offs will be made by each business' policies, based on the risk imposed by doing business when these privacy elements are unavailable. Given the current state of emerging personal privacy legislation worldwide, it is likely that most industries will need to specify high availability and reliability of all privacy-related elements. Obviously, the more complicated the rules are, the more complicated enforcement will be.

OA&M Perspective

The OA&M perspective addresses impacts to the operation, administration, and management of solutions based on the architecture as a result of enabling for consumer privacy. As with any system, OA&M requirements are determined by the business' policies and operating environment. Only those aspects of OA&M systems impacted by privacy are of concern to the architecture.

OA&M systems are comprised of components implementing instrumentation, infrastructure, and management applications. Because management infrastructure exists wholly to support management functions, there are no expected impacts to this component arising from privacy requirements. Primary impact derives from any additional instrumentation required as a result of enabling privacy, as well as new management applications that may be created to handle the new instrumentation data.

Some of the events that can be instrumented for privacy include access to personal/sensitive data, frequency of access to personal/sensitive data elements, logging of critical events, backup and recovery of personal/sensitive data, and performance monitoring. Threshold values will need to be established for the number of hits on personal data items, the number of violations, and the number and types of alerts. Alerts can be instituted for attempts to access personal data, as well as for unexpected and unauthorized accesses.

For implementations using some form of database system to store and maintain personal data, existing data management system rules will need to be augmented with privacy-related utilities and management applications for monitoring privacy-related events. Authorized system and database administrators must be aware of, and apply, legal issues and rules to the creation of additional rules and views required for enabling privacy. These authorized users must also have exclusive access to the privacy log for security reasons.

Summary

This chapter is intended as a guide as companies begin to launch activities that migrate their products and services toward including capabilities enabling consumer security and privacy within data warehouse environments. The expectation is that companies will examine their industry environments and leverage the content of this chapter addressing security and privacy concerns as they evolve in the industry architectures. Recommendations to modify this chapter are anticipated as a matter of course as better and more accurate information is gathered.

Notes

1. Directive 95/46/EC of the European Parliament and of the Council, 24 October 1995. See also "European Union Directive on Data Protection, Articles" at http://www.odpr.org/restofit/Legislation…les/Directive_Articles.html#anchor3080.
2. Directive 97/66/EC of the European Parliament and of the Council, 15 December 1997.
3. Merriam Webster Collegiate Edition, 1998.
4. Privacy Class of Common Criteria v2.0 (CC2.0 part 2) Security Functional Requirements (ISO/ IEC 15408).
5. "OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data," 23 September, 1980. http://www.oecd.org/dsti/sti/secur/prod/PRIV-EN.htm.
6. "FTC Releases Report on Consumers' Online Privacy," Report to Congress on Privacy Online, June 4, 1998, http://www.ftc.gov/opa/9806/privacy2.htm.
7. See Ken O'Flaherty's White Paper.
8. Opt-in: choosing to participate. Opt-out: choosing not to participate.
9. U.S. Safe Harbor proposals are designed to balance the privacy concerns of EU countries with the capabilities of U.S. companies to meet privacy requirements for doing business with citizens of EU countries.

RELATIONAL DATABASE SECURITY: AVAILABILITY, INTEGRITY, AND CONFIDENTIALITY

Ravi S. Sandhu and Sushil Jajodia

INSIDE

Access Controls, Multilevel Security, Inference and Aggregation, Integrity Mechanisms

INTRODUCTION

Data security has three separate, but interrelated objectives:

- *Confidentiality.* This objective concerns the prevention of improper disclosure of information.
- *Integrity.* This objective concerns prevention of improper modification of information or processes.
- *Availability.* This objective concerns improper denial of access to information.

These three objectives arise in practically every information system. There are differences, however, regarding the relative importance of these objectives in a given system. The commercial and military sectors have similar needs for high-integrity systems; however, the confidentiality and availability requirements of the military are often more stringent than those for typical commercial applications.

In addition, the objectives differ with respect to the level of understanding of the objectives themselves and the technology to achieve them. For example, availability is technically the least understood objective, and currently, no products address it directly. Therefore, availability is discussed only in passing in this article.

PAYOFF IDEA

Data security is an ongoing concern for database managers. This article explains the basic principles and mechanisms for enforcing security in relational databases. With a focus on prevention, it also covers common threats and the levels of security provided by relational database products.

The security policy defines the three security objectives in the context of the organization's needs and requirements system. In general, the policy defines what is improper for a particular system. This may be required by law (e.g., for confidentiality in the classified military and government sectors). However, the security policy is largely determined by the organization rather than by external mandates, particularly in the areas of integrity and availability.

Two distinct, mutually supportive mechanisms are used to meet the security objectives: prevention (i.e., attempts to ensure that security breaches cannot occur) and detection (i.e., provision of an adequate audit trail so that security breaches can be identified after they have occurred). Every system employs a mix of these techniques, though sometimes the distinction between them gets blurred. This article focuses on prevention, which is the more fundamental technique. To be effective, a detection mechanism first requires a mechanism for preventing improper modification of the audit trail.

A third technique for meeting security objectives is referred to as tolerance. Every practical system tolerates some degree of risk with respect to potential security breaches; however, it is important to understand which risks are being tolerated and which are covered by preventive and detective mechanisms.

Security mechanisms can be implemented with various degrees of assurance, which is directly related to the effort required to subvert the mechanism. Low-assurance mechanisms are easy to implement but relatively easy to subvert. High-assurance mechanisms are notoriously difficult to implement, and they often suffer from degraded performance. Fortunately, rapid advances in hardware performance are alleviating these constraints on performance.

ACCESS CONTROLS IN CURRENT SYSTEMS

This section discusses the access controls provided in the current generation of commercially available database management systems, with a focus on relational systems. The access controls described are often referred to as discretionary access controls as opposed to the mandatory access controls of multilevel security. This distinction is examined in the next section.

The purpose of access controls is to ensure that a user is permitted to perform only those operations on the database for which that user is authorized. Access controls are based on the premise that the user has been correctly identified to the system by some authentication procedure. Authentication typically requires the user to supply his or her claimed identity (e.g., user name or operator number) along with a password or some other authentication token. Authentication may be performed by the operating system, the database management system, a special authentication server, or some combination thereof.

Granularity and Modes of Access Control

Access controls can be imposed at various degrees of granularity in a system. For example, they can be implemented through the entire database, over one or more data relations, or in columns or rows of relations. Access controls are differentiated with respect to the operation to which they apply. These distinctions are important — for example, each employee may be authorized to read his own salary but not to write it. In relational databases, access control modes are expressed in terms of the basic SQL operations (i.e., SELECT, UPDATE, INSERT, and DELETE), as follows:

- The ability to insert and delete data is specified on a relation-by-relation basis.
- SELECT is usually specified on a relation-by-relation basis. Finer granularity of authorization for SELECT can be provided by views.
- UPDATE can be restricted to certain columns of a relation.

In addition to these access control modes, which apply to individual relations or parts thereof, there are privileges, which confer special authority on users. A common example is the DBA privilege for database administrators.

Data-Dependent Access Controls

Database access controls are often data dependent. For example, some users may be limited to viewing salaries less than $30,000. Similarly, a manager may be restricted to seeing salaries for employees in his or her department. There are two basic techniques for implementing data-dependent access controls in relational databases: view-based access controls and query modification.

View-based access control. A base relation is a relation actually stored in the database. A view is a virtual relation derived from base relations and other views. The database stores the view definitions and materializes the view as needed.

To illustrate the concept of a view and its security application, the following table shows the base relations of EMPLOYEE (the value NULL indicates that Harding has no manager):

NAME	DEPT	SALARY	MANAGER
Smith	Toy	10,000	Jones
Jones	Toy	15,000	Baker
Baker	Admin	40,000	Harding
Adams	Candy	20,000	Harding
Harding	Admin	50,000	NULL

The following SQL statement defines a view of these relations called TOY-DEPT:

```
CREATE   VIEW     TOY-DEPT
AS       SELECT   NAME, SALARY, MANAGER
         FROM     EMPLOYEE
         WHERE    DEPT = 'Toy'
```

This statement generates the view shown in the following table:

NAME	SALARY	MANAGER
Smith	10,000	Jones
Jones	15,000	Baker

To illustrate the dynamic aspects of views, a new employee, Brown, is inserted in base relation EMPLOYEE, as shown in the following table:

NAME	DEPT	SALARY	MANAGER
Smith	Toy	10,000	Jones
Jones	Toy	15,000	Baker
Baker	Admin	40,000	Harding
Adams	Candy	20,000	Harding
Harding	Admin	50,000	NULL
Brown	Toy	22,000	Harding

The view TOY-DEPT is automatically modified to include Brown, as shown in the following table:

NAME	SALARY	MANAGER
Smith	10,000	Jones
Jones	15,000	Baker
Brown	22,000	Harding

Views can be used to provide access to statistical information. For example, the following view gives the average salary for each department:

```
CREATE   VIEW AVSAL   (DEPT, AVG)
AS       SELECT       DEPT, AVG(SALARY)
         FROM         EMPLOYEE
         GROUP BY     DEPT
```

For retrieval purposes, users need not distinguish between views and base relations. A view is simply another relation in the database, which happens to be automatically modified by the DBMS whenever its base relations are modified. Thus, views provide a powerful mechanism for

specifying data-dependent authorization for data retrieval. However, there are significant problems if views are modified by users directly (rather than indirectly through modification of base relations). This is a result of the theoretical inability to translate updates of views into updates of base relations (discussed in a later section). This limits the usefulness of views for data-dependent authorization of update operations.

Query modification. Query modification is another technique for enforcing data-dependent access controls for retrieval. (Query modification is not supported in SQL but is discussed here for the sake of completeness.) In this technique, a query submitted by a user is modified to include further restrictions as determined by the user's authorization.

For example, the database administrator has granted Thomas the ability to query the EMPLOYEE base relation for employees in the toy department as follows:

```
GRANT    SELECT
ON       EMPLOYEE
TO       Thomas
WHERE    DEPT = 'Toy'
```

Thomas then executes the following query:

```
SELECT   NAME, DEPT, SALARY, MANAGER
FROM     EMPLOYEE
```

In the absence of access controls, this query would obtain the entire EMPLOYEE relation. Because of the GRANT command, however, the DBMS automatically modifies this query to the following:

```
SELECT   NAME, DEPT, SALARY, MANAGER
FROM     EMPLOYEE
WHERE    DEPT = 'Toy'
```

This limits Thomas to retrieving that portion of the EMPLOYEE relation for which he was granted SELECT access.

Granting and Revoking Access

GRANT and REVOKE statements allow users to selectively and dynamically grant privileges to other users and subsequently revoke them if so desired. In SQL, access is granted by means of the GRANT statement, which applies to base relations as well as views. For example, the following GRANT statement allows Chris to execute SELECT queries on the EMPLOYEE relation:

```
GRANT SELECT ON EMPLOYEE TO CHRIS
```

The GRANT statement may also be used to allow a user to act as database administrator, which carries with it many privileges. Because the database administrator DBA privilege confers systemwide authority, no relation need be specified in the command. For example, the following statement allows Pat to act as database administrator, and furthermore, to grant this privilege to others:

```
GRANT DBA TO PAT WITH GRANT OPTION
```

In SQL, it is not possible to give a user the GRANT OPTION on a privilege without further allowing the GRANT OPTION to be given to other users.

Accesses are revoked in SQL by means of the REVOKE statement. The REVOKE statement can remove only those privileges that the user also granted. For example, if Thomas has already granted Chris the SELECT privilege, he may execute the following command to revoke that privilege:

```
REVOKE SELECT ON EMPLOYEE FROM CHRIS
```

However, if Pat had also granted Chris the SELECT privilege, Chris would continue to retain this privilege after Thomas revokes it.

Because the WITH GRANT OPTION statement allows users to grant their privileges to other users, the REVOKE statements can have a cascading effect. For example, if Pat grants Chris the SELECT privilege, and Chris subsequently grants this privilege to Kelly, the privilege would be revoked from both Chris and Kelly if Pat later revokes it from Chris.

These access controls are said to be discretionary because the granting of access is at the user's discretion — that is, users who possess a privilege with the GRANT OPTION are free to grant that privilege to whomever they choose. This approach has serious limitations with respect to confidentiality requirements, as discussed in the following section.

Limitations of Discretionary Access Controls

If a privilege is granted without the GRANT OPTION, that user should not be able to grant the privilege to other users. However, this intention can be subverted by simply making a copy of the relation. For example, the first example of a GRANT statement allows Chris to execute SELECT queries on the EMPLOYEE relation, but it does not allow Chris to grant this privilege to others. Chris can get around this limitation by creating a copy of the EMPLOYEE relation, into which all the rows of EMPLOYEE are copied.

As the creator of COPY-OF-EMPLOYEE, Chris has the authority to grant any privileges for it to any user. For example, with the following state-

ment, Chris could grant Pat the ability to execute SELECT queries on the COPY-OF-EMPLOYEE relation:

```
GRANT SELECT ON COPY-OF-EMPLOYEE TO PAT
```

In essence, this gives Pat access to all the information in the original EMPLOYEE relation, as long as Chris keeps COPY-OF-EMPLOYEE reasonably up-to-date with respect to EMPLOYEE.

Even if users are trusted not to deliberately violate security in this way, Trojan horses can be programmed to do so. The solution is to impose mandatory access controls that cannot be violated, even by Trojan horses. Mandatory access controls are discussed in the following section.

MULTILEVEL SECURITY

This selection introduces the issue of multilevel security, which focuses on confidentiality. Discretionary access controls pose a serious threat to confidentiality; mandatory access controls help eliminate these problems. Multilevel secure database systems enforce mandatory access controls in addition to the discretionary controls commonly found in most current products.

The use of multilevel security, however, can create potential conflicts between data confidentiality and integrity. Specifically, the enforcement of integrity rules can create covert channels for discovering confidential information, which even mandatory access controls cannot prevent.

This section concludes with a brief discussion of the evaluation criteria for secure computer systems developed by the U.S. Department of Defense. It should be noted that although multilevel security systems were developed primarily for the military sector, they are relevant to the commercial sector as well.

Mandatory Access Controls

With mandatory access controls, the granting of access is constrained by the system security policy. These controls are based on security labels associated with each data item and each user. A label on a data item is called a security classification, and a label on a user is called a security clearance. In a computer system, every program run by a user inherits the user's security clearance — that is, the user's clearance applies not only to the user but to every program executed by that user. Once assigned, the classifications and clearances cannot be changed, except by the security officer.

Security labels in the military and government sectors have two components: a hierarchical component and a set of categories. The hierarchical component consists of the following classes, listed in decreasing

order of sensitivity: top secret, secret, confidential, and unclassified. The set of categories may be empty, or it may consist of such items as nuclear, conventional, navy, army, or NATO.

Commercial organizations use similar labels for protecting sensitive information. The main difference is that procedures for assigning clearances to users are much less formal than in the military or government sectors.

It is possible for security labels to dominate each other. For example, label *X* is said to dominate label *Y* if the hierarchical component of *X* is greater than or equal to the hierarchical component of *Y* and if the categories of *X* contain all the categories of *Y*. That is, if label *X* is (TOP-SECRET, {NUCLEAR, ARMY}) and label *Y* is (SECRET, {ARMY}), then label *X* dominates label *Y*. Likewise, if label *X* is (SECRET, {NUCLEAR, ARMY}), it would dominate label *Y*. If two labels are exactly identical, they are said to dominate each other.

If two labels are not comparable, however, neither one dominates the other. For example, if label *X* is (TOP-SECRET, {NUCLEAR}) and label *Y* is (SECRET, {ARMY}), they are not comparable.

The following discussion is limited to hierarchical labels without any categories. Although many subtle issues arise as a result of incomparable labels with categories, the basic concepts can be demonstrated with hierarchical labels alone. For simplicity, the labels denoting secret and unclassified classes are primarily used in this discussion.

When a user signs on to the system, that user's security clearance specifies the security level of that session. That is, a particular program (e.g., a text editor) is run as a secret process when executed by a secret user, but is run as an unclassified process when executed by an unclassified user. It is possible for a user to sign on at a security level lower than the one assigned to that user, but not at one higher. For example, a secret user can sign on as an unclassified user, but an unclassified user may not sign on as a secret user. Once a user is signed on at a specific level, all programs executed by that user will be run at that level.

Covert Channels

Although a program running at the secret level is prevented from writing directly to unclassified data items, there are other ways of communicating information to unclassified programs. For example, a program labeled secret can acquire large amounts of memory in the system. This can be detected by an unclassified program that is able to observe how much memory is available. If the unclassified program is prevented from directly observing the amount of free memory, it can do so indirectly by making a request for a large amount of memory itself. Such indirect methods of communication are called covert channels. Covert channels present a formidable problem for ensuring multilevel security. They are

difficult to detect, and once detected, they are difficult to close without incurring significant performance penalties.

Evaluation Criteria

The *Orange Book* established a metric against which computers systems can be evaluated for security. The metric consists of several levels: A1, B3, B2, B1, C2, C1, and D, listed here in decreasing order of how secure the system is.

For each level, the *Orange Book* lists a set of requirements that a system must have to achieve that level of security. Briefly, the D level consists of all systems that are not secure enough to qualify for any of A, B, or C levels. Systems at levels C1 and C2 provide discretionary protection of data; systems at level B1 provide mandatory access controls, and systems at levels B2 or higher provide increasing assurance, particularly against covert channels. Level A1, which is most rigorous, requires verified protection of data.

INFERENCE AND AGGREGATION

Even in multilevel secure DBMSs, it is possible for users to draw inferences from the information they obtain from the database. The inference could be derived purely from the data obtained from the database system, or it could additionally depend on some prior knowledge obtained by users from outside the database system. An inference presents a security breach if higher-classified information can be inferred from lower-classified information.

There is a significant difference between the inference and covert channel problems. Inference is a unilateral activity in which an unclassified user legitimately accesses unclassified information, from which that user is able to deduce secret information. Covert channels, on the other hand, require cooperation of a secret process that deliberately or unwittingly transmits information to an unclassified user by means of indirect communication. The inference problem exists even in an ideal system that is completely free of covert channels.

There are many difficulties associated with determining when more highly classified information can be inferred from lower-classified information. The biggest problem is that it is impossible to determine precisely what a user knows. The inference problem is somewhat manageable if the closed-world assumption is adopted; this is the assumption that if information Y can be derived using information X, both X and Y are contained in the database. In reality, however, the outside knowledge that users bring plays a significant role in inference.

There are two important cases of the inference problem that often arise in database systems. First, an aggregate problem occurs whenever

there is a collection of data items that is classified at a higher level than the levels of the individual data items by themselves. A classic example from a military context occurs when the location of individual ships in a fleet is unclassified, but the aggregate information concerning the location of all ships in the fleet is secret. Similarly, in the commercial sector, the individual sales figures for branch offices might be considered less sensitive than the aggregate sales figures for the entire company.

Second, a data association problem occurs whenever two values seen together are classified at a higher level than the classification of either value individually. For example, although the list consisting of the names of all employees and the list containing all employee salaries are unclassified, a combined list giving employee names with their salaries is classified. The data association problem is different from the aggregate problem because what is really sensitive is not the aggregate of the two lists, but the exact association giving an employee name and his salary.

The following sections describe some techniques for solving the inference problem. Although these methods can be extremely useful, a complete and generally applicable solution to the inference problem remains elusive.

Appropriate Labeling

One way to prevent unclassified information X from permitting disclosure of secret information Y is to reclassify all or part of information X such that it is no longer possible to derive Y from the disclosed subset of X. For example, attribute A is unclassified, and attribute B is secret. The database enforces the constraint $A + B \leq 20$, and that constraint is known to unclassified users. The value of B does not affect the value of A directly; however, it does constrain the set of possible values A can take. This is an inference problem, which can be prevented by reclassifying A as secret.

Query Restriction

Many inference violations arise as a result of a query that obtains data at the user's level; evaluation of this query requires accessing data above the user's level. For example, data is classified at the relations level, and there are two relations: (1) an unclassified relation, called EP, with attributes EMPLOYEE-NAME and PROJECT-NAME; and (2) a secret relation called PT, with attributes PROJECT-NAME and PROJECT-TYPE.EMPLOYEE-NAME as the key of the first relation and PROJECT-NAME as the key of the sec-

ond. (The existence of the relation scheme PT is unclassified.) An unclassified user makes the following SQL query:

```
SELECT   EP. PROJECT-NAME
FROM     EP,PT
WHERE    EP. PROJECT - NAME = PT. PROJECT-NAME AND
         EP.PROJECT - TYPE = 'NUCLEAR'
```

The data obtained by this query (i.e., the project names) is extracted from the unclassified relation EP. As such, the output of this query contains unclassified data, yet it reveals secret information by virtue of being selected on the basis of secret data in the PT relation.

Query restriction ensures that all data used in the process of evaluating the query is dominated by the level of the user and therefore prevents such inferences. To this end, the system can either simply abort the query or modify the user query so that the query involves only the authorized data.

Polyinstantiation

The technique of polyinstantiation is used to prevent inference violations. Essentially it allows different versions of the same information item to exist at different classification levels. For example, an unclassified user wants to enter a row in a relation in which each row is labeled either S (secret) or U (unclassified). If the same key is already occurring in an S row, the unclassified user can insert the U row, gaining access to any information by inference. The classification of the row must therefore be treated as part of the relation key. Thus, U rows and S rows always have different keys because the keys have different security classes.

The following table, which has the key STARSHIP-CLASS, helps illustrate this:

STARSHIP	DESTINATION	CLASS
Enterprise	Jupiter	S
Enterprise	Mars	U

A secret user inserted the first row in this relation. Later, an unclassified user inserted the second row. The second insertion must be allowed because it cannot be rejected without revealing to the unclassified user that a secret row for the enterprise already exists. Unclassified users see only one row for the Enterprise — namely, the U row. Secret users see both rows. These two rows might be interpreted in two ways:

- There are two distinct Starships named Enterprise going to two distinct destinations. Unclassified users know of the existence of only one of them (i.e., the one going to Mars). Secret users know about both of them.
- There is a single Starship named Enterprise. Its real destination is Jupiter, which is known only to secret users. However, unclassified users have been told that the destination is Mars.

Presumably, secret users know which interpretation is intended.

Auditing

Auditing can be used to control inferences. For example, a history can be kept of all queries made by a user. Whenever the user makes a query, the history is analyzed to determine whether the response to this query, when compared with responses to earlier queries, might suggest an inference violation. If so, the system can take appropriate action (i.e., abort the query).

The advantage of this approach is that it may deter many inference attacks by threatening discovery of violations. There are two disadvantages to this approach. First, it may be too cumbersome to be useful in practical situations. Second, it can detect only very limited types of inferences — it assumes that a violation can always be detected by analyzing the audit record for abnormal behavior.

Tolerating Limited Inferences

Tolerance methods are useful when the inference bandwidth is so small that these violations do not pose any threat. For example, the data may be classified at the column level, with two relations — one called PD with the unclassified attribute PLANE and the secret attribute DESTINATION, and another called DF with the unclassified attribute DESTINATION and the unclassified attribute FUEL-NEEDED. Although knowledge of the fuel needed for a particular plane can provide clues to the destination of the plane, there are too many destinations requiring the same amount of fuel for this to be a serious inference threat. Moreover, it would be too time-consuming to clear everybody responsible for fueling the plane to the secret level. Therefore, it is preferred that the derived relation with attributes PLANE and FUEL-NEEDED be made available to unclassified users.

Although it has been determined that this information does not provide a serious inference threat, unclassified users cannot be allowed to extract the required information from PD and DF, by, for example, executing the following query:

```
SELECT    PLANE,FUEL-NEEDED
FROM      PD,DF
WHERE     PD.DESTINATION = DF.DESTINATION
```

This query would open up a covert channel for leaking secret information to unclassified users.

One solution is to use the snapshot approach, by which a trusted user creates a derived secret relation with attributes PLANE and FUEL-NEEDED and then downgrades it to unclassified. Although this snapshot cannot be updated automatically without opening a covert channel, it can be kept more or less up-to-date by having the trusted user recreate it from time to time. A snapshot or a sanitized file is an important technique for controlling inferences, especially in offline, static databases. It has been used quite effectively by the U.S. Census Bureau.

INTEGRITY PRINCIPLES AND MECHANISMS

Integrity is a much less tangible objective than secrecy. For the purposes of this chapter, integrity is defined as being concerned with the improper modification of information. Modification includes insertion of new information, deletion of existing information, and changes to existing information. Such modifications may be made accidentally or intentionally.

Data may be accidentally modified when users simultaneously update a field or file, get deadlocked, or inadvertently change relationships. Therefore, controls must be in place to prevent such situations. Controls over nonmalicious errors and day-to-day business routines are needed as well as controls to prevent malicious errors.

Some definitions of integrity use the term unauthorized instead of improper. Integrity breaches can and do occur without authorization violations; however, authorization is only part of the solution. The solution must also account for users who exercise their authority improperly.

The threat posed by a corrupt authorized user is quite different in the context of integrity from what it is in the context of confidentiality. A corrupt user can leak secrets by using the computer to legitimately access confidential information and then passing on this information to an improper destination by another means of communication (e.g., a telephone call). It is impossible for the computer to know whether or not the first step was followed by the second step. Therefore, organizations have no choice but to trust their employees to be honest and alert.

Although the military and government sectors have established elaborate procedures for this purpose, the commercial sector is much more informal in this respect. Security research focusing on confidentiality

considers the principal threat to be Trojan horses embedded in programs; that is, the focus is on corrupt programs rather than on corrupt users.

Similarly, a corrupt user can compromise integrity by manipulating stored data or falsifying source or output documents. Integrity must therefore focus on the corrupt user as the principal problem. In fact, the Trojan horse problem can itself be viewed as a problem of corrupt system or application programmers who improperly modify the software under their control. In addition, the problem of the corrupt user remains even if all of the organization's software is free of Trojan horses.

Integrity Principles and Mechanisms

This section identifies basic principles for achieving data integrity. Principles lay down broad goals without specifying how to achieve them. The following section maps these principles to DBMS mechanisms, which establish how the principles are to be achieved.

There are seven integrity principles:

- *Well-formed transactions.* The concept of the well-formed transaction is that users should not manipulate data arbitrarily, only in restricted ways that preserve integrity of the database.
- *Least privilege.* Programs and users should be given the least privilege necessary to accomplish their jobs.
- *Separation of duties.* Separation of duties is a time-honored principle for prevention of fraud and errors by ensuring that no single individual is in a position to misappropriate assets on his own. Operationally, this means that a chain of events that affects the balance of assets must be divided into separate tasks performed by different individuals.
- *Reconstruction of events.* This principle seeks to deter improper behavior by threatening its discovery. The ability to reconstruct what happened in a system requires that users be accountable for their actions (i.e., that it is possible to determine what they did).
- *Delegation of authority.* This principle concerns the critical issue of how privileges are acquired and distributed in an organization. The procedures to do so must reflect the structure of the organization and allow for effective delegation of authority.
- *Reality checks.* Cross-checks with external reality are an essential part of integrity control. For example, if an internal inventory record does not correctly reflect the number of items in the warehouse, it makes little difference if the internal record is correctly recorded in the balance sheet.
- *Continuity of operation.* This principle states that system operations should be maintained at an appropriate level during potentially dev-

astating events that are beyond the organization's control, including natural disasters, power outages, and disk crashes.

These integrity principles can be divided into two groups, on the basis of how well existing DBMS mechanisms support them. The first group consists of well-formed transactions, continuity of operation, and reality checks. The second group comprises least privilege, separation of duties, reconstruction of events, and delegation of authority. The principles in the first group are adequately supported in existing products (to the extent that a DBMS can address these issues), whereas the principles in the second group are not so well understood and require improvement. The following sections discuss various DBMS mechanisms for facilitating application of these principles.

Well-formed transactions. The concept of a well-formed transaction corresponds well to the standard DBMS concept of a transaction. A transaction is defined as a sequence of primitive actions that satisfies the following properties:

- *Correct-state transform.* If run by itself in isolation and given a consistent state to begin with, each transaction will leave the database in a consistent state.
- *Serializability.* The net effect of executing a set of transactions is equivalent to executing them in a sequential order, even though they may actually be executed concurrently (i.e., their actions are interleaved or simultaneous).
- *Failure atomicity.* Either all or none of the updates of a transaction take effect. (In this context, update means modification, including insertion of new data, deletion of existing data, and changes to existing data.)
- *Progress.* Every transaction is eventually completed. That is, there is no indefinite blocking owing to deadlocks and no indefinite restarts owing to live locks (i.e., the process is repeatedly aborted and restarted because of other processes).

The basic requirement is that the DBMS must ensure that updates are restricted to transactions. If users are allowed to bypass transactions and directly manipulate relations in a database, there is no foundation to build on. In other words, updates should be encapsulated within transactions. This restriction may seem too strong because, in practice, there will always be a need to perform ad hoc updates. However, ad hoc updates can themselves be carried out by means of special transactions. The authorization for these special ad hoc transactions should be carefully controlled and their use properly audited.

DBMS mechanisms can help ensure the correctness of a state by enforcing consistency constraints on the data. (Consistency constraints are also often called integrity constraints or integrity rules.) The relational data model primarily imposes two consistency constraints:

- *Entity integrity* stipulates that attributes in the primary key of a relation cannot have null values. This amounts to requiring that each entity represented in the database must be uniquely identifiable.
- *Referential integrity* is concerned with references from one entity to another. A foreign key is a set of attributes in one relation whose values are required to match those of the primary key of some specific relation. Referential integrity requires that a foreign key either be null or that a matching tuple exist in the relation being referenced. This essentially rules out references to nonexistent entities.

Entity integrity is easily enforced. Referential integrity, on the other hand, requires more effort and has seen limited support in commercial products. In addition, the precise method for achieving it is highly dependent on the semantics of the application, particularly when the referenced tuple is deleted. There are three options: prohibiting the delete operation, deleting the referencing tuple (with a possibility of further cascading deletes), or setting the foreign key attributes in the referencing tuple to NULL.

In addition, the relational model encourages the use of domain constraints that require the values in a particular attribute (column) to come from a given set. These constraints are particularly easy to state and enforce as long as the domains are defined in terms of primitive types (e.g., integers, decimal numbers, and character strings). A variety of dependence constraints, which constrain the tuples in a given relation, have been extensively studied.

A consistency constraint can be viewed as an arbitrary predicate that all correct states of the database must satisfy. The predicate may involve any number of relations. Although this concept is theoretically appealing and flexible in its expressive power, in practice the overhead in checking the predicates for every transaction is prohibitive. As a result, relational DBMSs typically confine their enforcement of consistency constraints to domain constraints and entity integrity.

Least privilege. The principle of least privilege translates into a requirement for fine-grained access control. For the purpose of controlling read access, DBMSs have employed mechanisms based on views or query modification. These mechanisms are extremely flexible and can be as fine-grained as desired. However, neither one of the mechanisms provides the same flexibility for highly granular control of updates. The fundamental reason for this is the theoretical inability to translate updates on

views into updates of base relations. As a result, authorization to control updates is often less sophisticated than authorization for read access.

Fine-grained control of updates by means of views does not work well in practice. However, views are extremely useful for controlling retrieval. For example, the following table shows two base relations: EMP-DEPT and DEPT-MANAGER:

EMP	DEPT
Smith	Toy
Jones	Toy
Adams	Candy

DEPT	MANAGER
Toy	Brown
Candy	Baker

The following statement provides the EMP-MANAGER view of the base relations:

```
CREATE   VIEW EMP-MANAGER
AS       SELECT      EMP, MANAGER
         FROM        EMP-DEPT, DEPT-MANAGER
         WHERE       EMP-DEPT.DEPT = DEPT-MANAGER.DEPT
```

This statement results in the following table:

EMP	MANAGER
Smith	Brown
Jones	Brown
Adams	Baker

This view can be updated with the following statement:

```
UPDATE    EMP-MANAGER
SET       MANAGER = 'Green'
WHERE     EMP = 'Smith'
```

If EMP-MANAGER is a base relation, this statement would create the following table:

EMP	MANAGER
Smith	Green
Jones	Brown
Adams	Baker

This effect cannot be attained, however, by updating existing tuples in the two base relations in the first table. For example, the manager of the toy department can be changed as follows:

```
UPDATE    DEPT-MANAGER
SET       MANAGER = 'Green'
WHERE     DEPT = 'Toy'
```

This statement results in the following view:

EMP	MANAGER
Smith	Green
Jones	Green
Adams	Baker

The first updated view of EMP-MANAGER can be realized by modifying the base relations in the first table as follows:

EMP	DEPT	DEPT	MANAGER
Smith	X	X	Green
Jones	Toy	Toy	Brown
Adams	Candy	Candy	Baker

In this case, Smith is assigned to an arbitrary department whose manager is Green. It is difficult, however, to determine whether this is the intended result of the original update. Moreover, the UPDATE statement does not explain what X is.

Separation of duties. Separation of duties is not well supported in existing products. Although it is possible to use existing mechanisms for separating duties, these mechanisms were not designed for this purpose. As a result, their use is awkward at best.

Separation of duties is inherently concerned with sequences of transactions rather than individual transactions in isolation. For example, payment in the form of a check is prepared and issued by the following sequence of events:

- A clerk prepares a voucher and assigns an account.
- The voucher and account are approved by a supervisor.
- The check is issued by a clerk, who must be different from the clerk in the first item. Issuing the check also debits the assigned account.

This sequence embodies separation of duties because the three steps must be executed by different people. The policy has a dynamic flavor in that a particular clerk can prepare vouchers on one occasion and issue checks on another. However, the same clerk cannot prepare a voucher and issue a check for that voucher.

Reconstruction of events. The ability to reconstruct events in a system serves as a deterrent to improper behavior. In the DBMS context, the mechanism for recording the history of a system is traditionally called an audit trail. As with the principle of least privilege, a high-end DBMS should be capable of reconstructing events to the finest detail. In practice, this ability must be tempered with the reality that gathering audit data indiscriminately can generate an overwhelming volume of data. Therefore, a DBMS must also allow fine-grained selectivity regarding what is audited.

In addition, it should structure the audit trail logically so that it is easy to query. For example, logging every keystroke provides the ability to reconstruct the system history accurately. However, with this primitive logical structure, a substantial effort is required to reconstruct a particular transaction. In addition to the actual recording of all events that take place in the database, an audit trail must provide support for true auditing (i.e., an audit trail must have the capability for an auditor to examine it in a systematic manner). In this respect, DBMSs have a significant advantage because their powerful querying abilities can be used for this purpose.

Delegation of authority. The need to delegate authority and responsibility within an organization is essential to its smooth functioning. This need appears in its most developed form with respect to monetary budgets. However, the concept applies equally well to the control of other assets and resources of the organization.

In most organizations, the ability to grant authorization is never completely unconstrained. For example, a department manager may be able to delegate substantial authority over departmental resources to project managers within his department and yet be prohibited from delegating this authority to project managers outside the department. Traditional delegation mechanisms based on the concept of ownership (e.g., as embodied in the SQL `GRANT` and `REVOKE` statements) are not adequate in this context. Further work remains to be done in this area.

Reality checks. This principle inherently requires activity outside the DBMS. The DBMS has an obligation to provide an internally consistent view of that portion of the database that is being externally verified. This is particularly important if the external inspection is conducted on an ad hoc, on-demand basis.

Continuity of operation. The basic technique for maintaining continuity of operation in the face of natural disasters, hardware failures, and other disruptive events is redundancy in various forms. Recovery mechanisms in DBMSs must also ensure that the data is left in a consistent state.

CONCLUSION

Data security has three objectives: confidentiality, integrity, and availability. A complete solution to the confidentiality problem requires high-assurance, multilevel systems that impose mandatory controls and are known to be free of covert channels. Such systems are currently at the research and development stage and are not available.

Until these products become available, security administrators must be aware of the limitations of discretionary access controls for achieving secrecy. Discretionary access controls cannot cope with Trojan horse attacks. It is therefore important to ensure that only high-quality software of known origin is used in the system. Moreover, database administrators must appreciate that even the mandatory controls of high-assurance, multilevel systems do not directly prevent inference of secret information.

The integrity problem, somewhat paradoxically, is less well understood than confidentiality but is better supported in existing products. The basic foundation of integrity is the assurance that all updates are carried out by well-informed transactions. This is reasonably well supported by currently available DBMS products (e.g., DB2 and Oracle). Other integrity principles — such as least privilege, separation of duties, and delegation of authority — are not well supported. Products that satisfy these requirements are still in development. The availability objective is poorly understood. Therefore, existing products do not address it to any significant degree.

Ravi S. Sandhu and Sushil Jajodia are professors in the Information and Software Systems Engineering Department at George Mason University, Fairfax, VA.

101

Enterprise Security Architecture

William Hugh Murray, CISSP

Introduction

Sometime during the 1980s we crossed a line from a world in which the majority of computer users were users of multi-user systems to one in which the majority were users of single-user systems. We are now in the process of connecting all computers in the world into the most complex mechanism that humans have ever built. Although for many purposes we may be able to do this on an *ad hoc* basis, for purposes of security, audit, and control it is essential that we have a rigorous and timely design. We will not achieve effective, much less efficient, security without an enterprisewide design and a coherent management system.

If you look in the dictionary for the definitions of enterprise, you will find that an enterprise is a project, a task, or an undertaking; or, the readiness for such, the motivation, or the moving forward of that undertaking. The dictionary does not contain the definition of the enterprise as we are using it here. For our purposes here, the enterprise is defined as the largest unit of business organization, that unit of business organization that is associated with ownership. If the institution is a government institution, then it is the smallest unit headed by an elected official. What we need to understand is that it is a large, coordinated, and independent organization.

Enterprise Security in the 1990s

Because the scale of the computer has changed from one scaled to the enterprise to one scaled to the application or the individual, the computer security requirements of the enterprise have changed. The new requirement can best be met by an architecture or a design.

We do not do design merely for the fun of it or even because it is the "right" thing to do. Rather, we do it in response to a problem or a set of requirements. While the requirements for a particular design will be those for a specific enterprise, there are some requirements that are so pervasive as to be typical of many, if not most, enterprises. This section describes a set of observations by the author to which current designs should respond.

- *Inadequate expression of management intent.* One of these is that there is an inadequate expression of management's intent. Many enterprises have no written policy at all. Of those that do, many offer inadequate guidance for the decisions that must be made. Many say little more than "do good things." They fail to tell managers and staff how much risk general management is prepared or intends to accept. Many fail to adequately assign responsibility or duties or fix the discretion to say who can use what resources. This results in inconsistent risk and inefficient security, i.e., some resources are overprotected and others are underprotected.
- *Multiple sign-ons, IDs, and passwords.* Users are spending tens of minutes per day logging on and logging off. They may have to log on to several processes in tandem in order to access an application. They may have to log off of one application in order to log on to another. They may be required to remember multiple user identifiers and coordinate many passwords. Users are often forced into insecure or

inefficient behavior in futile attempts to compensate for these security measures. For example, they may write down or otherwise record identifiers and passwords. They may even automate their use in macros. They may postpone or even forget tasks so as not to have to quit one application in order to open another. This situation is often not obvious to system managers. They tend to view the user only in the context of the systems that they manage rather than in the context of the systems the user uses. Managers may also see this cost as "soft money," not easily reclaimed by him. On the other hand, it is very real money to the enterprise, which may have thousands of such users and which might be able to get by with fewer if they were not engaged in such activity. Said another way, information technology management overlooks what general management sees as an opportunity.

- *Multiple points of control.* Contrary to what we had hoped and worked for in the 1980s, data is proliferating and spreading throughout the enterprise. We did not succeed in bringing all enterprise data under a single access control system. Management is forced to rely on multiple processes to control access to data. This often results in inconsistent and incomplete control. Inconsistent control is usually inefficient. It means that management is spending too much or too little for protection. Incomplete control is ineffective. It means that some data is completely unprotected and unreliable.
- *Unsafe defaults.* In order to provide for ease of installation and avoid deadlocks, systems are frequently shipped with security mechanisms set to unsafe conditions by default. The designers are concerned that even before the system is completely installed, management may lose control. The administrator might accidentally lock himself out of his own system with no remedy but to start from scratch. Therefore, the system may be shipped with controls defaulted to their most open settings. The intent is that after the systems are configured and otherwise stable, the administrator will reset the controls to a safe condition. However, in practice and so as not to interfere with running systems, administrators are often reluctant to alter these settings. This may be complicated by the fact that systems that are not securely configured are, by definition, unstable. The manager has learned that changes to an already-unstable system tend to aggravate the instability.
- *Complex administration.* The number of controls, relations between them, and the amount of special knowledge required to use them may overwhelm the training of the administrator. For example, to properly configure the password controls for a Novell server, the administrator may have to set four different controls. The setting of one requires not only knowledge of how the others are set but also how they relate to each other. The administrator's training is often focused on the functionality of the systems rather than on security and control. The documentation tends to focus on the function of the controls while remaining silent on their use to achieve a particular objective or their relationship to other controls.
- *Late recognition of problems.* In part because of the absence of systematic measurement and monitoring systems, many problems are being detected and corrected late. Errors that are not detected or corrected may be repeated. Attacks are permitted to go on long enough to succeed. If permitted to continue for a sufficient length of time without corrective action, any attack will succeed. The cost of these problems is greater than it would be if they were detected on a more timely basis.
- *Increasing use, users, uses, and importance.* Most important for our purposes here, security requirements arise in the enterprise as the result of increasing use of computers, increasing numbers of users, increasing numbers of uses and applications, and increasing importance of those applications and uses to the enterprise. All of these things can be seen to be growing at a rate that dwarfs our poor efforts to improve security. The result is that relative security is diminishing to the point that we are approaching chaos.

Architecture Defined

In response to these things we must increase not only the effectiveness of our efforts but also their efficiency. Because we are working on the scale of the enterprise, *ad hoc* and individual efforts are not likely to be successful. Success will require that we coordinate the collective efforts of the enterprise according to a plan, design, or architecture.

Architecture can be defined as that part of design that deals with what things look like, what they do, where they are, and what they are made of. That is, it deals with appearance, function, location, and materials. It is

used to agree on what is to be done and what results are to be produced so that multiple people can work on the project in a collaborative and cooperative manner and so that we can agree when we are through and the results are as expected.

The design is usually reflected in a picture, model, or prototype; in a list of specified materials; and possibly in procedures to be followed in achieving the intended result. When dealing in common materials, the design usually references standard specifications. When using novel materials, the design must describe these materials in detail.

In information technology we borrow the term *architecture* from the building and construction industry. However, unlike this industry, we do not have 10,000 years of tradition, conventions, and standards behind us. Neither do we share the rigor and discipline that characterize them.

Traditional IT Environment

Computing environments can be characterized as traditional and modern. Each has its own security requirements but, in general and all other things being equal, the traditional environment is easier to secure than its modern equivalent.

- *Closed.* Traditional IT systems and networks are closed. Only named parties can send messages. The nodes and links are known in advance. The insertion of new ones requires the anticipation and cooperation of others. They are closed in the sense that their uses or applications are determined in advance by their design, and late changes are resisted.
- *Hierarchical.* Traditional IT can be described as hierarchical. Systems are organized and controlled top down, usually in a hierarchical or tree structure. Messages and controls flow vertically better than they do horizontally. Such horizontal traffic as exists is mediated by the node at the top of the tree, for example, a mainframe.
- *Point-to-point.* Traffic tends to flow directly from point to point along nodes and links that, at least temporarily, are dedicated to the traffic. Traffic flows directly from one point to another; what goes in at node A will come out only at node B.
- *Connection switched.* The resources that make up the connection between two nodes are dedicated to that connection for the life of the communication. When either is to talk to another, the connection is torn down and a new one is created. The advantage is in speed of communication and security, but capacity may not be used efficiently.
- *Host-dependent workstations.* In traditional computing, workstations are incapable of performing independent applications. They are dependent on cooperation with a host or master in order to be able to perform any useful work.
- *Homogeneous components.* In traditional networks and architectures, there is a limited number of different component types from a limited number of vendors. Components are designed to work together in a limited number of ways. That is to say, part of the design may be dictated by the components chosen.

Modern IT Environment

- *Open*. By contrast, modern computing environments are open. Like the postal system, for the price of a stamp anyone may send a message. For the price of an accommodation address, anyone can get an answer back. For not much more, anyone can open his own post office. Modern networks are open in the sense that nodes can be added late and without the permission or cooperation of others. They are open in the sense that their applications are not predetermined.
- *Flat.* The modern network is flat. Traffic flows with equal ease between any two points in the network. It flows horizontally as well as it does vertically. Traffic flows directly and without any mediation. If one were to measure the bandwidth between any two points in the network, chosen arbitrarily, it would be approximately equal to that between any other two points chosen the same way. While traffic may flow faster between two points that are close to each other, taken across the collection of all pairs, it flows with the same speed.

- *Broadcast*. Modern networks are broadcast. While orderly nodes accept only that traffic which is intended for them, traffic will be seen by multiple nodes in addition to the one for which it is intended. Thus, confidentiality may depend in part upon the fact that a large number of otherwise unreliable devices all behave in an orderly manner.
- *Packet-switched.* Modern networks are packet-switched rather than circuit-switched. In part this means that the messages are broken into packets and each packet is sent independent of the others. Two packets sent from the same origin to the same destination may not follow the same path and may not arrive at the destination in the same order that they were sent. The sender cannot rely on the safety of the path or the arrival of the message at the destination, and the receiver cannot rely on the return address. In part, it means that a packet may be broadcast to multiple nodes, even to all nodes, in an attempt to speed it to its destination. By design it will be heard by many nodes other than the ones for which it is intended.
- *Intelligent work stations.* In modern environments, the workstations are intelligent, independently programmable, and capable of performing independent work or applications. They are also vulnerable both to the leakage of sensitive information and to the insertion of malicious programs. These malicious programs may be untargeted viruses or they may be password grabbers that are aimed at specific workstations, perhaps those used by privileged users.
- *Heterogeneousness.* The modern network is composed of a variety of nodes and links from many different vendors. There may be dozens of different workstations, servers, and operating systems. The links may be of many speeds and employ many different kinds of signaling. This makes it difficult to employ an architecture that relies on the control or behavior of the components.

Other Security Architecture Requirements

- *IT architecture.* The information security architecture is derivative of and subordinate to the information technology architecture. It is not independent. One cannot build a security architecture except in the context of and in response to an IT architecture. An information technology architecture describes the appearance, function, location, and materials for the use of information technology. Often one finds that the IT architecture is not sufficiently well thought out or documented to support the development of the security architecture. That is to say, it describes fewer than all four of the things that an architecture must describe. Where it is documented at all, one can expect to find that it describes the materials but not appearance, location, or function.
- *Policy or management intent*. The security architecture must document and respond to a policy or an expression of the level of risk that management is prepared to take. This will influence materials chosen, the roles assigned, the number of people involved in sensitive duties, etc.
- *Industry and institutional culture*. The architecture must document and respond to the industry and institutional culture. The design that is appropriate to a bank will not work for a hospital, university, or auto plant.
- *Other*. Likewise, it must respond to the management style — authoritarian or permissive, prescriptive or reactive — of the institution, to law and regulation, to duties owed to constituents, and to good practice.

Security Architecture

The security architecture describes the appearance of the security functions, what is to be done with them; where they will be located within the organization, its systems, and its networks; and what materials will be used to craft them. Among other things, it will describe the following:

- *Duties, roles, and responsibilities*. It will describe who is to do what. It specifies who management relies on and for what. For every choice or degree of freedom within the system, the architecture will identify who will exercise it.
- *How objects will be named*. It will describe how objects are named. Specifically, it will describe how users are named, identified, or referred to. Likewise it will describe how information resources are to be named within the enterprise.

- *What authentication will look like.* It must describe how management gains sufficient confidence in these names or identifiers. How does it know that a user is who he says he is and that the data returned for a name is the expected data? Specifically, the architecture describes what evidence the user will present to demonstrate identity. For example, if authentication is based on something that the user knows, what are the properties (length and character set) of that knowledge?
- *Where it will be done.* Similarly, the architecture will describe where the instant data is to be collected, where the reference data will be stored, and what process will reconcile the two.
- *What the object of control will be.* The architecture must describe what it is that will be controlled. In the traditional IT architecture, this was usually a file or a dataset, or sometimes a procedure such as a program or a transaction type. In modern systems, it is more likely to be a database object such as a table or a view.
- *Where access will be controlled.* The architecture will describe where, i.e., what processes, will exercise control over the objects. In the traditional IT architecture, we tried to centralize all access control in a single process, scaled to the enterprise. In more modern systems, access will be controlled in a large number of places. These places will be scaled to departments, applications, and other ways of organizing resources. They may be exclusive or they may overlap. How they are related and where they are located is the subject of the design.
- *Generation and distribution of warnings and alarms.* Finally, the design must specify what events or combinations of events require corrective action, what process will detect them, who is responsible for the action, and how the warning will be communicated from the detecting process to the party responsible for the correction.

Policy

A Statement of Management's Intent

Among other things, a policy is a statement of management's intent. Among other things, a security policy describes how much risk management intends to take. This statement must be adequate for managers to be able to figure out what to do in a given set of circumstances. It should be sufficiently complete that two managers will read it the same way, reach similar conclusions, and behave in similar ways.

It should speak to how much risk management is prepared to take. For example, management expects to take normal business risk, or acceptable and accepted risk. Alternately or in addition, management can specify the intended level of control. For example, management can say that controls must be such that multiple people must be involved in sensitive duties or material fraud.

The policy should state what management intends to achieve, for example, data integrity, availability, and confidentiality, and how it intends to do it. It should clearly state who is to be responsible for what. It should state who is to have access to what information. Where such access is to be restricted or discretionary, then the policy should state who will exercise the discretion.

The policy should be such that it can be translated into an access control policy. For example, it might say that read access to confidential data must be restricted to those authorized by the owner of the data. The architecture will describe how a given platform or a network of platforms will be used to implement that policy.

Important Security Services

The architecture will describe the security mechanisms and services that will be used to implement the access control policy. These will include but not be limited to the following:

- *User name service.* The user name service is used for assigning unique names to users and for resolving aliases where necessary. It can be thought of as a database, database application, or database service. The server can encode and decode user names into user identifiers. For the distinguished user name, it returns a system user identifier or identifiers. For the system user identifier, it returns a distinguished user name. It can be used to store information about the user. It is often used to store other descriptive

data about the user. It may store office location, telephone number, department name, and manager's name.

- *Group name service*. The group name service is used for assigning unique group names and for associating users with those groups. It permits the naming of any arbitrary but useful group such as member of department m, employees, vendors, consultants, users of system 1, users of application A, etc. It can also be used to name groups of one, such as the payroll manager. For the group name, it returns the names, identifiers, or aliases of members of the group. For a user name, it returns a list of the groups of which that user is a member. A complete list of the groups of which a user is a member is a description of his role or relationship to the enterprise. Administrative activity can be minimized by assigning authority, capabilities, and privileges to groups and assigning users to the groups. While this is indirect it is also usually efficient.
- *Authentication server*. The authentication server reconciles evidence of identity. Users are enrolled along with the expectation, i.e., the reference data, for authenticating their identity. For a user identifier and an instance of authenticating data, the server returns *true* if the data meets its expectation, i.e., matches the reference data, and *false* if it does not. If *true*, the server will vouch to its clients for the identity of the user. The authentication server must be trusted by its client, and the architecture must provide the basis for that trust. The server may be attached to its client by a trusted path or it may give its client a counterfeit-resistant voucher (ticket or encryption-based logical token).
- *Authentication service products*. A number of authentication services are available off the shelf. These include Kerberos, SESAME, NetSP, and Open Software Foundation Distributed Computing Environment (OSF/DCE). These products can meet some architectural requirements in whole or in part.
- *Single point of administration*. One implication of multiple points of control is that there may be multiple controls that must be administered. The more such controls there are, the more desirable it becomes to minimize the points of administration. Such points of administration may simply provide for a common interface to the controls or may provide for a single database of its own. There are a number of standard architectures that are useful here. These include SESAME and the Open Software Foundation Distributed Computing Environment.

Recommended Enterprise Security Architecture

This section makes some recommendations about enterprise security architecture. It describes those choices which, all other things being equal, are to be preferred over others.

- *Single-user name space for the enterprise*. Prefer a single-user name space across all systems. Alternatively, have an enterprise name server that relates all of a user's aliases to his distinguished name. This server should be the single point of name assignment. In other words, it is a database application or server for assigning names.
- *Prefer strong authentication*. Strong authentication should be preferred by all enterprises of interest. Strong authentication is characterized by two kinds of evidence, at least one of which is resistant to replay. Users should be authenticated using two kinds of evidence. Evidence can be something that only one person knows, has, is, or can do. The most common form of strong authentication is something that the user knows, such as a password, passphrase, or personal identification number (PIN), plus something that the user carries, such as a token. The token generates a one-time password that is a function of time or a challenge. Other forms in use include a token plus palm geometry or a PIN plus the way the user speaks.
- *Prefer single sign-on*. A user should have to log on only once per workstation per enterprise per day. A user should not be surprised that if he changes workstations, crosses an enterprise boundary, or leaves for the day, he should have to log on again. However, he should not have to log off one application to log on to another or log on to multiple processes to use one application.
- *Application or service as point of control*. Prefer the application or service as the point of control. The first applicable principle is that the closer to the data the control is, the fewer instances of it there will be, the less subject it will be to user interference, the more difficult it will be to bypass, and consequently, the more reliable it will be. This principle can be easily understood by contrasting it to the worst case —

the one where the control is on the desktop. Multiple copies must be controlled, they are very vulnerable to user interference, not to mention complete abrogation, and the more people there are who are already behind the control. The second principle is that application objects are specific, i.e., their behavior is intuitive, predictable from their name, and obvious as to their intended use. Contrast "update name and address of customer" to "write to customer database." One implication of the application as the point of control is that there will be more than one point of control. However, there will be fewer than if the control were even closer to the user.

- *Multiple points of control.* Each server or service should be responsible for control of access to all of its dynamically allocated resources. Prefer that all such resources be of the same resource type. To make its access decision, the server may use local knowledge or data or it may use a common service that is sufficiently abstract to include its rules. One implication of the server or service as the point of control is that there will be multiple points of control. That is to say, there are multiple repositories of data and multiple mechanisms that management must manipulate to exercise control. This may increase the requirement for special knowledge, communication, and coordination.
- *Limited points of administration.* Therefore, prefer a limited number of points of administration that operate across a number of points of control. These may be relatively centralized to respond to a requirement for a great deal of special knowledge about the control mechanism. Alternatively, it can be relatively decentralized to meet a requirement for special knowledge about the users, their duties, and responsibilities.
- *Single resource name space for enterprise data.* Prefer a single name space for all enterprise data. Limit this naming scheme to enterprise data; i.e., data that is used and meaningful across business functions or that is related to the business strategy. It is not necessary to include all business functional data, project data, departmental data, or personal data.
- *Object, table, or view as unit of control.* Prefer capabilities, objects, tables, views, rows, columns, and files, in that order, as objects of control. This is the order in which the data are most obvious as to meaning and intended use.
- *Arbitrary group names with group-name service.* It is useful to be able to organize people into affinity groups. These may include functions, departments, projects, and other units of organization. They may also include such arbitrary groups as employees, nonemployees, vendors, consultants, contractors, etc. The architecture should deal only with enterprisewide groups. It should permit the creation of groups that are strictly local to a single organizational unit or system. Enterprise group names should be assigned and group affinities should be managed by a single service across the enterprise and across all applications and systems. This service may run as part of the user name service. Within reasonable bounds, any user should be able to define a group for which he is prepared to assume ownership and responsibility. Group owners should be able to manage group membership or delegate it. For example, the human resources manager might wish to restrict the ability to add members to the group *payroll department* while permitting any manager to add users to the group *employee* or the group *nonemployee.*
- *Rules-based (as opposed to list-based) access control.* Prefer rules-based to list-based access control. For example, "access to data labelled confidential is limited to employees" should be preferred to "user A can access dataset 1." While the latter is more granular and specific, the former covers more data in a single rule. The latter will require much more administrative activity to accomplish the same result as the former. Similarly, it can be expressed in far less data. While the latter may permit only a few good things to happen, the former forbids a large number of bad things. This recommendation is counter-intuitive to those of us who are part of the tradition of "least-possible privilege." This rule implies that a user should be given access to only those resources required to do his job and that all access should be explicit. The rule of least privilege worked well in a world in which the number of users, data objects, and relations between them was small. It begins to break down rapidly in the modern world of tens of millions of users and billions of resources.
- *Data-based rules.* Access control rules should be expressed in terms of the name and other labels of the data rather than in terms of the procedure to be performed. They should be independent of the procedures used to access the data or the environment in which they are stored. That is, it is better to say that a user has *read* access to *filename* than to say that he has *execute* access to *word.exe*. It makes little sense to say that a user is restricted to a procedure that can perform arbitrary operations on an

unbounded set of objects. This is an accommodation to the increase in the number of data objects and the decreasing granularity of the procedures.

- *Prefer single authentication service.* Evidence of user identity should be authenticated by a single central process for the entire enterprise and across all systems and applications. These systems and applications can be clients of the authentication server, or the server can issue trusted credentials to the user that can be recognized and honored by the using systems and applications.
- *Prefer a single standard interface for invoking security services.* All applications, services, and systems should invoke authentication, access control, monitoring, and logging services via the same programming interface. The generalized system security application programming interface (GSSAPI) is preferred in the absence of any other overriding considerations. Using a single interface permits the replacement or enhancement of the security services with a minimum of disruption.
- *Encryption services.* Standard encryption services should be available on every platform. These will include encryption, decryption, key management, and certificate management services. The Data Encryption Standard algorithm should be preferred for all applications, save key management, where RSA is preferred. A public key server should be available in the network. This service will permit a user or an application to find the public key of any other.
- *Automate and hide all key management functions.* All key management should be automated and hidden from users. No keys should ever appear in the clear or be transcribed by a user. Users should reference keys only by name. Prefer dedicated hardware for the storage of keys. Prefer smart cards, tokens, PCMCIA cards, other removable media, laptops, or access-controlled single-user desktops, in that order. Only keys belonging to the system manager should be stored on a multi-user system.
- *Use firewalls to localize and raise the cost of attacks.* The network should be compartmented with firewalls. These will localize attacks, prevent them from spreading, increase their cost, and reduce the value of success. Firewalls should resist attack traffic in both directions. That is, each subnetwork should use a firewall to connect to any other. A subnet manager should be responsible for protecting both his own net and connecting nets from any attack traffic. A conservative firewall policy is indicated. That is, firewalls should permit only that traffic that is necessary for the intended applications and should hide all information about one net from the other.
- *Access control begins on the desktop.* Access control should begin on the desktop and be composed up rather than begin on the mainframe and spread down. The issue here is to prevent the insertion of malicious programs more than to prevent the leakage of sensitive data.

Appendix I

Principles of Good Design

- *Prefer broad solutions to point solutions.* Prefer broad security solutions, which work across the enterprise, multiple applications, multiple resources, and against multiple hazards, to those that are limited to or specific to one of these. Such practices are almost always more efficient than a collection of mechanisms that are specific to applications, resources, or hazards.
- *Prefer end-to-end solutions to point-by-point solutions.* Similarly, prefer encryption-based end-to-end security solutions that are independent of the network. The more sensitive the application and the more hostile the network, the greater this preference. Such solutions are more robust and more efficient than those that attempt to identify and fix all of the vulnerabilities between the ends of the path.
- *Design top down, implement bottom up.* Design by functional decomposition and successive refinement. Implement by composition from the bottom. Prefer early deployment of those services and servers that will be required over the long haul.
- *Do it right the first time.* When building infrastructure, build for the ages. Do it right the first time. This strategy is more effective and more efficient than the "assess and patch" strategy that has been the approach to security in the past.

- *Prefer planning to fixing*. Similarly, work by plan and design rather than by experimentation. Necessary experimentation should be carefully identified, contained, and controlled.
- *Prefer long term to short*. Applications are becoming more sensitive and the environment more hostile. While one may consent to a plan that permits an early deployment of an application with a plan to deploy the agreed-upon security function by a certain date , do not take a "wait and see" approach.
- *Justify across the enterprise and time*. Security measures must be justified across the entire enterprise and across the life of the application or the mechanism. By definition, security prefers predictable, regular, prevention costs to unpredictable, irregular, remedial costs. They should be justified across a time frame that is consistent with the normal frequency of the events that it addresses. Security measures are relatively easy to justify in this manner and difficult to justify locally or in the short term. In justifying security measures, weight should be given to the fact that applications are becoming more sensitive, more interoperable, and more important, and that the environment in which they operate is becoming less reliable and more hostile.
- *Provide economy of safe use*, Using the system safely should require as little user effort as possible. For example, a user should have to log on only once per enterprise, per workstation, per day.
- *Provide consistent presentation and appearance*. Security should look the same across the enterprise, i.e., applications, systems, and platforms.
- *Make control predictable and intuitive*. Systems should be supportive. They should encapsulate the special knowledge required by the manager and user to operate them. They should make this information available to the manager and user at the time of use.
- *Provide ease of safe use.* Design in such a way that it is easy to do the right thing. Penalties should be associated with doing the wrong thing (e.g., economy of log on, user should have to log on only once per workstation, per enterprise, per day.)
- *Prefer mechanisms that are obvious as to their intent*. Avoid mechanisms that are complex or obscure, that might cause error, or be used to conceal malice. For example, prefer online transactions, EDI, secure formatted e-mail, formatted e-mail, e-mail, and file transfer in that order. The online transaction is always obvious and predictable; for a given set of inputs one can predict the outputs. Although the intent of a file transfer may be obvious, it is not necessarily so.
- *Encapsulate necessary special knowledge*. Necessary special knowledge should be included in documentation or programs.
- *Prefer simplicity; hide complexity.* For example, all other things being equal, simple mechanisms should be preferred to complex ones. Prefer a single mechanism to two, a single instance of a mechanism should be preferred to multiple ones. For example, prefer a single appearance of administration, such as CA Unicenter Star, to the appearance of all the systems that may be hidden by it. Similarly, prefer a single point of administration such as SAM or RAS to Unicenter Star.
- *Place controls close to the resource*. As a rule and all other things being equal, controls should be as close to the resource as possible. The closer to the resource, the more reliable the control, the more resistant to interference, and the more resistant to bypass. Controls should be server-based, rather than client-based.
- *Place operation of the control as close as possible to where the knowledge is and where the effect can be observed*. For example, prefer controls operated by the owner of the resource, the manager of the group, the manager of the system, and the manager of the user rather than by a surrogate such as a security administrator. Although a surrogate has the necessary special knowledge to operate the control, he knows less about the intent and the effect of the control. He cannot observe the effect and take corrective action. Surrogates are often compensation for a missing, complex, or poorly designed control.
- *Prefer localized control and data*. As a general rule and all other things being equal, prefer solutions that place reliance on as few controls in as few places as possible. Not only are such solutions more effective and efficient, but they are also more easily apprehended, comprehended, and demonstrated. Distribute function and data as required or indicated for performance, reliability, availability, and use or control.

Appendix II

References

IBM Security Architecture [SC28-8135-01]
ECMA 138 (SESAME) (see http://www.esat.kuleuven.ac.be/cosic/sesame3_2.html)
Open Systems Foundation Distributed Computing Architectures
(see http://www.osf.org/tech_foc.htm)

Appendix III

Glossary

Architecture — That part of design that deals with appearance, function, location, and materials.
Authentication — The testing or reconciliation of evidence; reconciliation of evidence of user identity.
Cryptography — The art of secret writing; the translation of information from a public code to a secret one and back again for the purpose of limiting access to it to a select few.
Distinguished User Name — User's full name so qualified as to be unique within a population. Qualifiers may include such things as enterprise name, organization unit, date of birth, etc.
Enterprise — The largest unit of organization; usually associated with ownership. (In government, it is associated with sovereignty or democratic election.)
Enterprise Data — Data that is defined, meaningful, and used across business functions or for the strategic purposes of the enterprise.
Name Space — All of the possible names in a domain, whether used or not.
PIN — Personal Identification Number; evidence of personal identity when used with another form.

Appendix IV

Products of Interest

- *Secure authentication products.* A number of clients and servers share a protocol for secure authentication. These include Novell Netware, Windows NT, and Oracle Secure Network Services. A choice of these may meet some of the architectural requirements.
- *Single sign-on products.* Likewise, there are a number of products on the market that meet some or all of the requirements for limited or single sign-on:
  - SSO DACS (Mergent International) (see http://www.pilgrim.umass.edu/pub/security/mergent.html)
  - NetView Access Services (IBM) (see http://www.can.ibm.com/mainframe/software/sysman/p32.html)
  - SuperSession (see http://www.candle.com/product_info/solutions/SOLCL.HTM)
  - NetSP (IBM) (see http://www.raleigh.ibm.com/dce/dcesso.html)
- *Authentication services.* A number of standard services are available for authenticating evidence of user identity:
  - Ace Server (see http://www.securid.com/ID188.100543212874/Security/ACEdata.html)
  - TACACS (see http://sunsite.auc.dk/RFC/rfc/rfc1492.html)
  - Radius (see http://www.tribe.com/support/TribeLink/RADIUS/RADIUSpaper.html)

- *Administrative services.* There are a number of products that are intended for creating and maintaining access control data across a distributed computing environment:
  - Security Administration Manager (SAM) (Schumann, AG) (see http://www.schumann-ag.de/deutsch/sam/sam.html)
  - RAS (Technologic) (see http://www.technologic.com/RAS/rashome.html)
  - Omniguard Enterprise Security Manager (Axent) (http://www.axent.com:80/axent/products/products.html)
  - Mergent Domain DACS (http://www.mergent.com/html/products.html)
  - RYO ("Roll yer own")

102

Certification and Accreditation Methodology

Mollie E. Krehnke, CISSP, IAM and
David C. Krehnke, CISSP, CISM, IAM

The implementation of a certification and accreditation (C&A) process within industry for information technology systems will support cost-effective, risk-based management of those systems and provide a level of security assurance that can be known (proven). The C&A process addresses both technical and nontechnical security safeguards of a system to establish the extent to which a particular system meets the security requirements for its business function (mission) and operational environment.

Definitions

Certification involves all appropriate security disciplines that contribute to the security of a system, including administrative, communications, computer, operations, physical, personnel, and technical security. Certification is implemented through involvement of key players, conduct of threat and vulnerability analyses, establishment of appropriate security mechanisms and processes, performance of security testing and analyses, and documentation of established security mechanisms and procedures.

Accreditation is the official management authorization to operate a system in a particular mode, with a prescribed set of countermeasures, against a defined threat with stated vulnerabilities and countermeasures, within a given operational concept and environment, with stated interconnections to other systems, at an acceptable level of risk for which the accrediting authority has formally assumed responsibility, and for a specified period of time.

C&A Target

The subject of the C&A, the information technology system or application (system), is the hardware, firmware, and software used as part of the system to perform organizational information processing functions. This includes computers, telecommunications, automated information systems, and automatic data processing equipment. It includes any assembly of computer hardware, software, and firmware configured to collect, create, communicate, compute, disseminate, process, store, and control data or information.

Repeatable Process

The C&A is a repeatable process that can ensure an organization (with a higher degree of confidence) that an appropriate combination of security measures is correctly implemented to address the system's threats and

vulnerabilities. This assurance is sustained with the conduct of periodic reviews and monitoring of the system's configuration throughout its life cycle, as well as recertification and reaccreditation on a routine, established basis.

References for Creating a C&A Process

The performance of certification and accreditation is well established within the federal government sector, its civil agencies, and the Department of Defense. There are numerous processes that have been established, published, and implemented. Any of these documents could serve as an appropriate starting point for a business organization. Several are noted below:

- *Guideline for Computer Security Certification and Accreditation* (Federal Information Processing Standard Publication 102)[1]
- *Introduction to Certification and Accreditation* (NCSC-TG-029, National Computer Security Center)[2]
- *National Information Assurance Certification and Accreditation Process* (NIACAP) (NTISSI No. 1000, National Security Agency)[3]
- *Sample Generic Policy and High-Level Procedures Certification and Accreditation* (National Institute of Standards and Technology)[4]
- *DoD Information Technology Security Certification and Accreditation Process* (DITSCAP) (Department of Defense Instruction Number 5200.40)[5]
- *How to Perform Systems Security Certification and Accreditation (C&A) within the Defense Logistics Agency (DLA) Using Metrics and Controls for Defense-in-Depth*[6]
- *Certification and Accreditation Process Handbook for Certifiers* (Defense Information Systems Agency [DISA])[7]

The FIPS guideline, although almost 20 years old, presents standards and processes that are applicable to government and industry. The NIACAP standards expand upon those presented in the NCSC documentation. The NIST standards are generic in nature and are applicable to any organization. The DLA documentation is an example of a best practice that was submitted to NIST and made available to the general public for consideration and use.

Take Up the Tools and Take a Step

This chapter presents an overview of the C&A process, including key personnel, components, and activities within the process that contribute to its success in implementation. The conduct of the C&A process within an industrial organization can also identify areas of security practices and policies that are presently not addressed, but need to be addressed to ensure information resources are adequately protected. The C&A task may appear to be daunting, but even the longest journey begins with a single step. Take that step and begin.

C&A Components

The timely, accurate, and effective implementation of a C&A initiative for a system is a choreography of people, activities, documentation, and schedules. To assist in the understanding of what is involved in a C&A, the usual resources and activities are grouped into the following tables and then described:

- Identification of key personnel to support the C&A effort
- Analysis and documentation of minimum security controls and acceptance
- Other processes that support C&A effectiveness
- Assessment and recertification timelines
- Associated implementation factors

The tables reflect the elements under discussion and indicate whether the element was cited by a reference used to create the composite C&A presented in this chapter. The content is very similar across references, with minor changes in terms used to represent a C&A role or phase of implementation.

Identification of Key Personnel to Support C&A Effort

The C&A process cannot be implemented without two key resources: people and funding. The costs associated with a C&A will be dependent on the type of C&A conducted and the associated activities. For example, the NIACAP identifies four general certification levels (discussed later in the chapter). In contrast, the types of personnel, and their associated functions, required to implement the C&A remain constant. However, the number of persons involved and the time on task will vary with the number and complexity of C&As to be conducted and the level of testing to be performed. These personnel are listed in Exhibit 102.1. It is vital to the completeness and effectiveness of the C&A that these individuals work together as a team, and they all understand their roles and associated responsibilities.

Authorizing Official/Designated Approving Authority

The authorizing official/designated approving authority (DAA) has the authority to formally assume responsibility for operating a system at an acceptable level of risk. In a business organization, a vice president or chief information officer would assume this role. This individual would not be involved in the day-to-day operations of the information systems and would be supported in the C&A initiatives by designated representatives.

Certifier

This individual is responsible for making a technical judgment of the system's compliance with stated requirements, identifying and assessing the risks associated with operating the system, coordinating the certification activities, and consolidating the final certification and accreditation packages. The certifier is the technical expert that documents trade-offs between security requirements, cost, availability, and schedule to manage the security risk.

Information Systems Security Officer

The information systems security officer (ISSO) is responsible to the DAA for ensuring the security of an IT system throughout its life cycle, from design through disposal, and may also function as a certifier. The ISSO provides guidance on potential threats and vulnerabilities to the IT system, provides guidance regarding security requirements and controls necessary to protect the system based on its sensitivity and criticality to the organization, and provides advice on the appropriate choice of countermeasures and controls.

Program Manager/DAA Representative

The program manager is ultimately responsible for the overall procurement, development, integration, modification, operation, maintenance, and security of the system. This individual would ensure that adequate resources (e.g., funding and personnel) are available to conduct the C&A in a timely and accurate manner.

EXHIBIT 102.1 Key Personnel

Title	FIPS	NCSC	NIACAP	NIST	DITSCAP
Authorizing Official/Designated Approving Authority	X	X	X	X	X
Certifier	X	X	X	X	X
Information Systems Security Officer	X	X	X	X	X
Program Manager/DAA Representative	X	X	X		X
System Supervisor/Manager	X	X	X	X	X
User/User Representative	X	X	X	X	X

System Supervisor or Manager

The supervisor or manager of a system is responsible for ensuring the security controls agreed upon during the C&A process are consistently and correctly implemented for the system throughout its life cycle. If changes are required, this individual has the responsibility for alerting the ISSO as the DAA representative about the changes; and then a determination can be made about the need for a new C&A, because the changes could impact the security of the system.

User and User Representative

The user is a person or process that accesses the system. The user plays a key role in the security of the system by protecting the assigned passwords, following established rules to protect the system in its operating environment, being alert to anomalies that could indicate a security problem, and not sharing information with others who do not have a need to know that information. A user representative supports the C&A process by ensuring that system availability, access, integrity, functionality, performance, and confidentiality as they relate to the users, their business functions, and the operational environment are appropriately addressed in the C&A process.

Analysis and Documentation of Security Controls and Acceptance

A system certification is a comprehensive analysis of technical and nontechnical security features of a system. Security features are also referred to as controls, safeguards, protection mechanisms, and countermeasures. Operational factors that must be addressed in the certification are system environment, proposed security mode of operation, specific users, applications, data sensitivity, system configuration, site/facility location, and interconnections with other systems. Documentation that reflects analyses of those factors and associated planning to address specified security requirements is given in Exhibit 102.2. This exhibit represents a composite of the documentation that is suggested by the various C&A references.

Threats, Vulnerabilities, and Safeguards Analysis

A determination must be made that proposed security safeguards will effectively address the system's threats and vulnerabilities in the operating environment at an acceptable level of risk. This activity could be a technical assessment that is performed by a certifier or contained in the risk management process (also noted in Exhibit 102.2). The level of analysis will vary with the level of certification that is performed.

EXHIBIT 102.2 Analysis and Documentation of Security Controls and Acceptance

Documentation	FIPS	NCSC	NIACAP	NIST	DITSCAP
Threats, Vulnerabilities, and Safeguards Analysis	X	X	X	X	X
Contingency/Continuity of Operations Plan	X	X	X	X	X
Contingency/Continuity of Operations Plan Test Results	X	X	X	X	X
Letter of Acceptance/Authorization Agreement	X	X	X	X	X
Letter of Deferral/List of System Deficiencies	X	X	X	X	X
Project Management Plan for C&A	X		X		X
Risk Management	X	X	X	X	X
Security Plan/Security Concept of Operations	X	X	X	X	X
Security Specifications	X	X	X	X	X
Security/Technical Evaluation and Test Results	X	X	X	X	X
System Security Architecture	X	X	X		X
User Security Rules	X	X	X	X	X
Verification and Validation of Security Controls	X	X	X	X	X

Contingency/Continuity of Operations Plan

The resources allocated to continuity of operations will be dependent upon the system business functions, criticality, and interdependency with other systems. The plan for the system should be incorporated into the plan for the facility in which the system resides and should address procedures that will be implemented at varying levels of business function disruption and recovery.

Contingency/Continuity of Operations Plan Test Results

Testing of the continuity of operations plan should be conducted on an established schedule that is based on system factors cited above and any associated regulatory or organizational requirements. There are various levels of testing that can be performed, depending on the system criticality and available resources, including checklists, table-top testing, drills, walk-throughs, selected functions testing, and full testing.

Letter of Acceptance/Authorization Agreement

The decision to accredit a system is based upon many factors that are encompassed in the certification results and recommendations: threats and vulnerabilities, system criticality, availability and costs of alternative countermeasures, residual risks, and nonsecurity factors such as program and schedule risks.

The DAA has several options available:

- Full accreditation for the originally intended operational environment and acceptance of the associated recertification/reaccreditation timeline
- Accreditation for operation outside of the originally intended environment (e.g., change in mission, crisis situation, more restrictive operations)
- Interim (temporary) accreditation approval with a listing of activities to be performed in order to obtain full accreditation
- Accreditation disapproval (see letter of deferral below)

Letter of Deferral/List of System Deficiencies

This letter indicates the accreditation is disapproved, and it includes recommendations and timelines for correcting specified deficiencies.

Project Management Plan for C&A

Many individuals (and organizations) provide support in the accurate and timely completion of a system C&A. A project management plan reflects the activities, timelines, and resources that have been allocated to the C&A effort; and it must be managed as any other tasking is managed.

Risk Management

The identification of system threats, vulnerabilities, and compensating controls that enable the system to function at an acceptable level of risk is key to the C&A process. Risk analysis should be conducted throughout the system life cycle to ensure the system is adequately protected, and it should be conducted as early as possible in the development process. The DAA must accept responsibility for system operation at the stated level of risk. A change in the threats, vulnerabilities, or acceptable level of risk may trigger a system recertification prior to the planned date as defined in the DAA acceptance letter.

Security Plan/Concept of Operations

The security plan/concept of operations (CONOPS) documents the security measures that have been established and are in place to address a system security requirement. Some organizations combine the security plan and CONOPS into one document, and other organizations include the technical controls in the security plan and the day-to-day administrative controls in the CONOPS. The security plan/CONOPS is a living

document that must be updated when security controls, procedures, or policies are changed. NIST has provided a generic security plan template for both applications and major systems that is recognized as appropriate for government and industry.

Security Specifications

The level to which a security measure must perform a designated function must be specified during the C&A process. Security functions will include authentication, authorization, monitoring, security management, and security labeling. These specifications will be utilized during the testing of the security controls prior to acceptance and periodically thereafter, particularly during the annual self-assessment process.

Security/Technical Evaluation and Test Results

The evaluation and testing of controls is performed to assess the performance of the security controls in the implementation of the security requirements. The controls must function as intended on a consistent basis over time. Each control must be tested to ensure conformance with the associated requirements. In addition, the testing must validate the functionality of all security controls in an integrated, operational setting. The level of evaluation and testing will depend upon the level of assurance required for a control. The testing should be performed at the time of installation and at repeated intervals throughout the life cycle of the control to ensure it is still functioning as expected. Evaluation and testing should include such areas as identification and authentication, audit capabilities, access controls, object reuse, trusted recovery, and network connection rule compliance.

System Security Architecture

A determination must be made that the system architecture planned for operation complies with the architecture description provided for the C&A documentation. The analysis of the system architecture and interconnections with other systems is conducted to assess how effectively the architecture implements the security policy and identified security requirements. The hardware, software, and firmware are also evaluated to determine their implementations of security requirements. Critical security features, such as identification, authentication, access controls, and auditing, are reviewed to ensure they are correctly and completely implemented.

User Security Rules

All authorized users will have certain security responsibilities associated with their job functions and with a system. These responsibilities and the rules associated with system use must be clearly defined and understood by the user. General user rules and responsibilities may be covered during security awareness and training. Other rules and responsibilities associated with a particular system may be covered during specific system operational and security training.

Verification and Validation of Security Controls

The identification, evaluation, and tracking of the status of security safeguards is an ongoing process throughout the life cycle of a system. The evaluation of the security posture of a control can also be used to evaluate the security posture of the organization. The following evaluations should be considered:

- *Requirements evaluation.* Are the security requirements acceptable? Certification is only meaningful if security requirements are well defined.
- *Function evaluation.* Does the design or description of security functions satisfy the security requirements? Basic evaluations should address all applicable control features down through the logical specification level as defined in the functional requirements document, and they should include internal computer controls and external physical and administrative controls.
- *Control implementation determination.* Are the security functions implemented? Functions that are described in a document or discussed in an interview do not prove that they have been implemented. Visual inspection and testing will be necessary.

- *Methodology review.* Does the implementation method provide assurance that security functions are acceptably implemented? This review may be used if extensive testing is not deemed necessary or cannot be implemented. The review contributes to a confidence judgment on the extent to which controls are reliably implemented and on the susceptibility of the system to flaws. If the implementation cannot be relied upon, then a detailed evaluation may be required.
- *Detailed evaluation.* What is the quality of the security safeguards? First decide what safeguards require a detailed analysis, and then ask the following questions: Do the controls function properly? Do controls satisfy performance criteria? How readily can the controls be broken or circumvented?

Other Processes Supporting C&A Effectiveness

See Exhibit 102.3 for information on other processes supporting C&A effectiveness.

Applicable Laws, Regulations, Policies, Guidelines, and Standards — Federal and State

Federal and state regulations and policies provide a valuable and worthwhile starting point for the formulation and evaluation of security requirements — the cornerstone of the C&A process. Compliance may be mandatory or discretionary, but implementing information security at a generally accepted level of due diligence can facilitate partnerships with government and industry.

Applicable Policies, Guidelines, and Standards — Organizational

Organizational policies reflect the business missions, organizational and environmental configurations, and resources available for information security. Some requirements will be derived from organizational policies and practices.

Configuration and Change Management

Changes in the configuration of a system, its immediate environment, or a wider organizational environment may impact the security posture of that system. Any changes must have approval prior to implementation so that the security stance of the system is not impacted. All changes to the established baseline must be documented. Significant changes may initiate a new C&A (discussed later in this chapter). Accurate system configuration documentation can also reduce the likelihood of implementing unnecessary security mechanisms. Extraneous mechanisms add unnecessary complexity to the system and are possible sources of additional vulnerabilities.

EXHIBIT 102.3 Other Processes Supporting C&A Effectiveness

Topic/Activity	FIPS	NCSC	NIACAP	NIST	DITSCAP
Applicable laws, regulations, policies, guidelines, and standards — federal and state	X	X	X	X	X
Applicable policies, guidelines, and standards — organizational	X	X	X	X	X
Configuration and change management	X	X	X		X
Incident response		X	X		X
Incorporation of security into system life cycle	X	X	X		X
Personnel background screening	X	X	X	X	X
Security awareness training	X	X	X	X	X
Security management organization	X	X	X		X
Security safeguards and metrics	X	X	X	X	X

Incident Response

Incidents are going to happen. An organization's response to an incident — that is, identification, containment, isolation, resolution, and prevention of future occurrences — will definitely affect the security posture of the organization. The ability to respond to an incident in a timely and effective manner is necessary to maintaining an organization's business functions and its perceived value to customers.

Incorporation of Security into System Life Cycle

The determination of applicable security functionality early in system design and development will reduce the security costs and increase the effectiveness and functionality of the designated security controls. Adding on security functions later in the development or production phase will reduce the security options and add to the development costs. The establishment of system boundaries will ensure that security for the system environment is adequately addressed, including physical, technical, and administrative security areas.

Personnel Background Screening

Managers are responsible for requesting suitability screening for the staff in their respective organizations. The actual background investigations are conducted by other authorized organizations. The determination of what positions will require screening is generally based upon the type of data to which an individual will have access and the ability to bypass, modify, or disable technical or operating system security controls. These requirements are reviewed by an organization's human resources and legal departments, and are implemented in accordance with applicable federal and state laws and organizational policy.

Security Awareness Training

The consistent and appropriate performance of information security measures by general users, privileged users, and management cannot occur without training. Training should encompass awareness training and operational training, including basic principles and state-of-the-art technology. Management should also be briefed on the information technology security principles so that the managers can set appropriate security requirements in organizational security policy in line with the organization's mission, goals, and objectives.

Security Management Organization

The security management organization supports the development and implementation of information security policy and procedures for the organization, security and awareness training, operational security and rules of behavior, incident response plans and procedures, virus detection procedures, and configuration management.

Security Safeguards and Metrics

A master list of safeguards or security controls and an assessment of the effectiveness of each control supports the establishment of an appropriate level of assurance for an organization. The master list should contain a list of uniquely identified controls, a title that describes the subject area or focus of the control, a paragraph that describes the security condition or state that the control is intended to achieve, and the rating of compliance based on established metrics for the control.

The levels of rating are:

1. No awareness of the control or progress toward compliance
2. Awareness of the control and planning for compliance
3. Implementation of the security control is in progress
4. Security control has been fully implemented, and the security profile achieved by the control is actively maintained

The metrics can be based on federal policy, audit findings, commercial best practices, agency system network connection agreements, local security policy, local configuration management practices, information sensitivity and criticality, and DAA-specified requirements.

EXHIBIT 102.4 Assessment and Recertification Timelines

Topic/Activity	FIPS	NCSC	NIACAP	NIST	DITSCAP
Annual assessment between C&As			X	X	X
Recertification required every three to five years	X	X	X	X	X
Significant change or event	X	X	X	X	X
Security safeguards operating as intended	X	X	X	X	X

Assessment and Recertification Timelines

Certification and accreditation should be viewed as continuing and dynamic processes. The security posture of a system must be monitored, tracked, and assessed against the security controls and processes established at the time of the approval and acceptance of the certification documentation (see Exhibit 102.4).

Annual Assessment between C&As

The annual assessment of a system should include a review of the system configuration, connections, location, authorized users, and information sensitivity and criticality. The assessment should also determine if the level of threat has changed for the system, making the established controls less effective and thereby necessitating the need for a new C&A.

Recertification Required Every Three to Five Years

Recertification is required in the federal government on a three- to five-year basis, or sooner if there has been a significant change to the system or a significant event that alters the security stance (or effectiveness of the posture) of a system. The frequency with which recertification is conducted in a private organization or business will depend on the sensitivity and criticality of the system and the impact if the system security controls are not adequate for the organizational environment or its user population.

Significant Change or Event

The C&A process may be reinitiated prior to the date established for recertification. Examples of a significant change or event are:

- *Upgrades to existing systems*: upgrade/change in operating system, change in database management system, upgrade to central processing unit (CPU), or an upgrade to device drivers.
- *Changes to policy or system status*: change to the trusted computing base (TCB) as specified in the security policy, a change to the application's software as specified in the security policy, a change in criticality or sensitivity level that causes a change in the countermeasures required, a change in the security policy (e.g., access control policy), a change in activity that requires a different security mode of operation, or a change in the threat or system risk.
- *Configuration changes to the system or its connectivity*: additions or changes to the hardware that require a change in the approved security countermeasures, a change to the configuration of the system that may affect the security posture (e.g., a workstation is connected to the system outside of the approved configuration), connection to a network, and introduction of new countermeasures technology.
- *Security breach or incident*: if a security breach or significant incident occurs for a system.
- *Results of an audit or external analysis*: if an audit or external analysis determines that the system was unable to adequately respond to a higher level of threat force than that originally determined, or a change to the system created new vulnerabilities, then a new C&A would be initiated to ensure that the system operates at the acceptable level of risk.

EXHIBIT 102.5 Associated Implementation Factors

Topic/Activity	FIPS	NCSC	NIACAP	NIST	DITSCAP
Documentation available in hard copy and online					X
Grouping of systems for C&A			X		X
Presentation of C&A process to management					X
Standardization of procedures, templates, worksheets, and reports	X		X		X
Standardization of responses to report sections for enterprise use	X		X		X

Security Safeguards Operating as Intended

An evaluation of the system security controls should be performed to ensure that the controls are functioning as intended. This activity should be performed on a routine basis throughout the year and is a component of the annual self-assessment conducted in support of the C&A process.

Associated Implementation Factors

Associated implementation factors are listed in Exhibit 102.5.

Documentation Available in Hard Copy and Online

If a number of systems are undergoing the C&A process, it is beneficial to have the C&A documentation available in hard copy and online so that individuals responsible for its completion can have ready access to the forms. This process can save time and ensure a higher level of accuracy in the C&A results because all individuals have the appropriate forms.

Grouping of Systems for C&A

It is acceptable to prepare one C&A for like systems that have the same configuration, controls, location, function, and user groups. The grouping of systems does not reduce the effectiveness of the C&A process, as long as it can be assured that all of the systems are implementing the established controls in the appropriate manner and that the controls are appropriate for each system.

Presentation of C&A Process to Management

Management at all levels of an organization must understand the need for and importance of the C&A process and the role that each plays in its successful implementation. Management must also understand that the C&A process is an ongoing activity that is going to require resources (at a predesignated level) over the system life cycle to preserve its security posture and reduce risk to an acceptable level.

Standardization of C&A Procedures, Templates, Worksheets, and Reports

Standardization within an organization supports accuracy and completeness in the forms that are completed and the processes that are performed. Standardized forms enhance the analysis and preparation of summary C&A reports and enable a reviewer to readily locate needed information. Standardization also facilitates the identification of gaps in the information provided and in the organization's security posture.

Standardization of Responses to Report Sections for Enterprise Use

The results of the C&A process will be provided to management. The level of detail provided may depend on the responsibilities of the audience, but consistency across systems will allow the organization to establish an enterprisewide response to a given threat or vulnerability, if required.

C&A Phases

The C&A process is a method for ensuring that an appropriate combination of security measures are implemented to counter relevant threats and vulnerabilities. Activities conducted for the C&A process can be grouped into phases, and a composite of suggested activities (from the various references) is described below. The number of activities or steps varies slightly among references.

Phase 1: Precertification

Activity 1: Preparation of the C&A Agreement

Analyze pertinent regulations that impact the content and scope of the C&A. Determine usage requirements (e.g., operational requirements and security procedures). Analyze risk-related considerations. Determine the certification type. Identify the C&A team. Prepare the C&A agreement.

Aspects to be considered in this activity include mission criticality, functional requirements, system security boundary, security policies, security concept of operations, system components and their characteristics, external interfaces and connection requirements, security mode of operation or overall risk index, system and data ownership, threat information, and identification of the DAAs.

Activity 2: Plan for C&A

Plan the C&A effort, obtain agreement on the approach and level of effort, and identify and obtain the necessary resources (including funding and staff).

Aspects to be considered in this activity include reusability of previous evidence, life-cycle phase, and system milestones (time constraints).

Phase 2: Certification

Activity 3: Perform the Information Security Analysis of Detailed System Information

Conduct analyses of the system documentation, testing performed, and architecture diagrams. Conduct threat and vulnerability assessments, including impacts on confidentiality, integrity, availability, and accountability.

Aspects to be considered in this activity include the certification team becoming more familiar with the security requirements and security aspects of individual system components, specialized training on the specific system (depending on the scope of this activity and the experience of the certification team), determining whether system security controls adequately satisfy security requirements, identification of system vulnerabilities, and determination of residual risks.

Activity 4: Document the Certification Results in a Certification Package

Document all analyses, testing results, and findings. The certification package is the consolidation of all the certification activity results. This documentation will be used as supporting documentation for the accreditation decision and will also support recertification/reaccreditation activities.

Aspects to be considered in this documentation package include system need/mission overview, security policy, security CONOPS or security plan, contingency plan/continuity of operations, system architectural description and configuration, reports of evaluated products, statements from other responsible agencies indicating specified security requirements have been met, risk analysis report and associated countermeasures, test plans, test procedures, test results, analytic results, configuration management plan, and previous C&A information.

Phase 3: Accreditation

Activity 5: Perform Risk Assessment and Final Testing

Review the analysis, documentation, vulnerabilities, and residual risks. Final testing is conducted at this time to ensure the DAAs are satisfied that the residual risk identified meets an acceptable level of risk.

Aspects to be considered in this activity include assessment of system information via the certification package review, the conduct of a site accreditation survey to verify that the residual risks are at an acceptable level, and verification of the contents of the C&A package.

Activity 6: Report Findings and Recommendations

The recommendations are derived from documentation gathered by the certification team, testing conducted, and business functions/mission considerations, and include a statement of residual risk and supporting documentation.

Aspects to be considered in this activity include executive summary of mission overview; architectural description; system configuration, including interconnections; memoranda of agreement (MOA); waivers signed by the DAA that specific security requirements do not need to be met or are met by other means (e.g., procedures); residual risk statement, including rationale for why residual risks should be accepted or rejected; recommendation for accreditation decision.

Activity 7: Make the Accreditation Decision

The decision will be based on the recommendation from the certifier or certification authority. Is the operation of the system, under certain conditions, in a specified environment, functioning at an acceptable level of risk?

Accreditation decision options include full accreditation approval, accreditation for operations outside the originally intended environment, interim (temporary) accreditation approval, or accreditation disapproval.

Phase 4: Post-Accreditation

Activity 8: Maintain the Security Posture and Accreditation of the System

Periodic compliance inspections of the system and recertification at established time frames will help to ensure that the system continues to operate within the stated parameters as specified in the accreditation letter. A configuration management or change management system must be implemented and procedures established for baselining, controlling, and monitoring changes to the system. Substantive changes may require the system to be recertified and reaccredited prior to the established time frame. However, maximum reuse of previous evaluations or certifications will expedite this activity.

Aspects to be considered in this activity include significant changes that may impact the security of the system.

Types of Certification

NIACAP identifies four general certification levels: Level 1 — Basic Security Review, Level 2 — Minimum Analysis, Level 3 — Detailed Analysis, and Level 4 — Comprehensive Analysis. FIPS PUB 102 presents three levels of evaluation: basic, detailed, and detailed focusing. DISA identified the following types of C&A.

Type 1: Checklist

This type of certification completes a checklist with yes or no responses to the following content areas: administrative, personnel authorization, risk management, personnel security, network security, configuration management, training, media handling, and physical security. This type of certification also includes verification that procedures for proper operation are established, documented, approved, and followed.

Type 2: Abbreviated Certification

This type of certification is more extensive than Type 1 certification but also includes the completion of the Type 1 checklist. The amount of documentation required and resources devoted to the Type 2 C&A is minimal. The focus on this type of certification is information security functionality (e.g., identification and authentication, access control, auditing).

FIPS Pub. 102's first level of evaluation, the basic evaluation, is similar to the Type 2 category; it is concerned with the overall functional security posture, not with the specific quality of individual controls. The basic evaluation has four tasks:

1. *Security requirements evaluation*. Are applicable security requirements acceptable?
   - — *Assets*. What should be protected?
   - — *Threats*. What are assets protected against?
   - — *Exposures*. What might happen to assets if a threat is realized?

— *Controls*. How effective are safeguards in reducing exposures?

2. *Security function evaluation.* Do application security functions satisfy the requirements?
   — *Defined requirements/security functions*. Authentication, authorization, monitoring, security management, security labeling.
   — *Undefined requirements/specific threats*. Analysis of key controls; that is, how effectively do controls counter specific threats?
   — *Completed to the functional level*. Logical level represented by functions as defined in the functional requirements document.
3. *Control existence determination*. Do the security functions exist?
   — *Assurance* that controls exist via visual inspection or testing of internal controls.
4. *Methodology review*. Does the implementation method provide assurance that security functions are acceptably implemented?
   — *Documentation*. Is it current, complete, and of acceptable quality?
   — *Objectives*. Is security explicitly stated and treated as an objective?
   — *Project control*. Was development well controlled? Were independent reviews and testing performed, and did they consider security? Was an effective change control program used?
   — *Tools and techniques*. Were structured design techniques used? Were established programming practices and standards used?
   — *Resources*. How experienced in security were the people who developed the application? What were the sensitivity levels or clearances associated with their positions?

Type 3: Moderate Certification

This type of certification is more detailed and complex and requires more resources. It is generally used for systems that require higher degrees of assurance, have a greater level of risk, or are more complex. The focus of this type of certification is also information security functionality (e.g., identification and authentication, access control, auditing); however, more extensive evidence is required to show that the system meets the security requirements.

FIPS Pub. 102's second level of evaluation, the detailed evaluation, is similar to the Type 3 category; and it provides further analysis to obtain additional evidence and increased confidence in evaluation judgments. The detailed evaluation may be initiated because (1) the basic evaluation revealed problems that require further analysis, (2) the application has a high degree of sensitivity, or (3) primary security safeguards are embodied in detailed internal functions that are not visible or suitable for examination at the basic evaluation level.

Detailed evaluations involve analysis of the quality of security safeguards. The tasks include:

- *Functional operation.* Do controls function properly?
  — *Control operation*. Do controls work?
  — *Parameter checking*. Are invalid or improbable parameters detected and properly handled?
  — *Common error conditions*. Are invalid or out-of-sequence commands detected and properly handled?
  — *Control monitoring*. Are security events properly recorded? Are performance measurements properly recorded?
  — *Control management*. Do procedures for changing security tables work?
- *Performance.* Do controls satisfy performance criteria?
  — *Availability*. What proportion of time is the application or control available to perform critical or full services?
  — *Survivability*. How well does the application or control withstand major failures or natural disasters?
  — *Accuracy*. How accurate is the application or control, including the number, frequency, and significance of errors?
  — *Response time*. Are response times acceptable? Will the user bypass the control because of the time required?
  — *Throughput*. Does the application or control support required usage capabilities?
- *Penetration resistance.* How readily can controls be broken or circumvented?

Resistance testing is the extent to which the application and controls must block or delay attacks. The focus of the evaluation activities will depend on whether the penetrators are users, operators, application programmers, system programmers, managers, or external personnel. Resistance testing should also be conducted against physical assets and performance functions. This type of testing can be the most complex of detailed evaluation categories, and it is often used to establish a level of confidence in security safeguards.

Areas to be considered for detailed testing are:

- Complex interfaces
- Change control process
- Limits and prohibitions
- Error handling
- Side effects
- Dependencies
- Design modifications/extensions
- Control of security descriptors
- Execution chain of security services
- Access to residual information

Additional methods of testing are flaw identification or hypothesizing generic flaws and then determining if they exist. These methods can be applied to software, hardware, and physical and administrative controls.

Type 4: Extensive Certification

This type of certification is the most detailed and complex type of certification and generally requires a great deal of resources. It is used for systems that require the highest degrees of assurance and may have a high level of threats or vulnerabilities. The focus of this type of certification is also information security functionality (e.g., identification and authentication, access control, auditing) and assurance. Extensive evidence, generally found in the system design documentation, is required for this type of certification.

FIPS Pub. 102's third level of evaluation, the detailed focusing evaluation, is similar to the Type 4 category. Two strategies for focusing on a small portion of the security safeguards for a system are: (1) security-relevant components and (2) situational analysis.

The security-relevant components strategy addresses previous evaluation components in a more detailed analysis:

- *Assets.* Which assets are most likely at risk? Examine assets in detail in conjunction with their attributes to identify the most likely targets.
- *Threats.* Which threats are most likely to occur? Distinguish between accidental, intentional, and natural threats and identify perpetrator classes based on knowledge, skills, and access privileges. Also consider threat frequency and its components: magnitude, asset loss level, exposures, existing controls, and expected gain by the perpetrator.
- *Exposures.* What will happen if the threat is realized, for example, internal failure, human error, errors in decisions, fraud? The focus can be the identification of areas of greatest potential loss or harm.
- *Controls.* How effective are the safeguards in reducing exposures? Evaluations may include control analysis (identifying vulnerabilities and their severity), work-factor analysis (difficulty in exploiting control weaknesses), or countermeasure trade-off analysis (alternative ways to implement a control).

Situational analysis may involve an analysis of attack scenarios or an analysis of transaction flows. Both of these analyses are complementary to the high-level basic evaluation, providing a detailed study of a particular area of concern. An attack scenario is a synopsis of a projected course of events associated with the realization of a threat. A manageable set of individual situations is carefully examined and fully understood. A transaction flow is a sequence of events involved in the processing of a transaction, where a transaction is an event or task of significance and visible to the user. This form of analysis is often conducted in information systems auditing and should be combined with a basic evaluation.

Conclusion

Summary

There are a significant number of components associated with a certification and accreditation effort. Some of the key factors may appear to be insignificant, but they will greatly impact the success of the efforts and the quality of the information obtained.

- All appropriate security disciplines must be included in the scope of the certification. Although a system may have very strong controls in one area, weak controls in another area may undermine the system's overall security posture.
- Management's political and financial support is vital to the acceptance and implementation of the C&A process. Management should be briefed on the C&A program, its objectives, and its processes.
- Information systems to undertake a C&A must be identified and put in a priority order to ensure that the most important systems are addressed first.
- Security requirements must be established (if not already available); and the requirements must be accurate, complete, and understandable.
- Technical evaluators must be capable of performing their assigned tasks and be able to remain objective in their evaluation. They should have no vested interest in the outcome of the evaluation.
- Access to the personnel and documentation associated with an information system is vital to the completion of required documentation and analyses.
- A comprehensive basic evaluation should be performed. A detailed evaluation should be completed where necessary.

Industry Implementation

Where do you stand?

- If your organization's security department is not sufficiently staffed, what type of individuals (and who) can be tasked to support C&As on a part-time basis?
- C&A process steps and associated documentation will be necessary. Use the references presented in this chapter as a starting point for creating the applicable documentation for your organization.
- Systems for which a C&A will be conducted must be identified. Consider sensitivity and criticality when you are creating your list. Identify those systems with the highest risks and most impact if threats are realized. Your organization has more to lose if those systems are not adequately protected.
- The level of C&A to be conducted will depend on the available resources. You may suggest that your organization starts with minimal C&A levels and move up as time and funding permit. The level of effort required will help you determine the associated costs and the perceived benefits (and return on investment) for conducting the C&As.

Take that Step and Keep Stepping

You may have to start at a lower level of C&A than you would like to conduct for your organization, but you are taking a step. Check with your colleagues in other organizations on their experiences. Small, successful C&As will serve as a marketing tool for future efforts. Although the completion of a C&A is no guarantee that there will not be a loss of information confidentiality, integrity, or availability, the acceptance of risk is based on increased performance of security controls, user awareness, and increased management understanding and control. Remember: take that step. A false sense of security is worse than no security at all.

References

1. Guideline for Computer Security Certification and Accreditation, Federal Information Processing Standards Publication 102, U.S. Department of Commerce, National Bureau of Standards, September 27, 1983.
2. Introduction to Certification and Accreditation, NCSC-TG-029, National Computer Security Center, U.S. Government Printing Office, January 1994.
3. National Information Assurance Certification and Accreditation Process (NIACAP), National Security Telecommunications and Information Systems Security Committee, NSTISSC 1000, National Security Agency, April 2000.
4. Sample Generic Policy and High Level Procedures, Federal Agency Security Practices, National Institute of Standards and Technology, www.csrc.nist.gov/fasp.
5. Department of Defense (DoD) Information Technology Security Certification and Accreditation Process (DITSCAP), DoD Instruction 5200.40, December 30, 1997.
6. How to Perform Systems Security Certification and Accreditation (C&A) within the Defense Logistics Agency (DLA) Using Metrics and Controls for Defense-in-Depth (McDid), Federal Agency Security Practices, National Institute of Standards and Technology, www.csrc.nist.gov/fasp.
7. *The Certification and Accreditation Process Handbook for Certifiers,* Defense Information Systems Agency, INFOSEC Awareness Division, National Security Agency.

A Framework for Certification Testing

Kevin J. Davidson, CISSP

The words have often been heard, "We have a firewall" in response to the question, "What are you doing to protect your information?" Security professionals recognize the fact that the mere existence of a firewall does not in and of itself constitute good information security practices. Information system owners and managers generally are not aware of a need to verify that the security policies and procedures they have established are followed, if in fact they have established policies or procedures.

In this chapter, the focus is on system security certification as an integral part of the system accreditation process. Accreditation may also be called *authorization* or *approval*. The fact is that each and every information system that is operating in the world today has been through some type of accreditation or approval process, either through some formal or informal process or, in many cases, by default because the process does not exist. System owners and managers along with information owners and managers have approved the system to operate, either by some identified and documented process or by default. It is incumbent upon information security professionals and practitioners to subscribe to a method of ensuring those systems operate as safely and securely as possible in the interconnected and open environment that exists in the world today.

The approaches and methods outlined in this chapter are intended as guidelines and a framework from which to build an Information System Security Certification Test. They are not intended to be a set of rules; rather, they are intended to be a process that can be tailored to meet the needs of each unique environment.

INTRODUCTION

To provide a common frame of reference, it is necessary to define the terms that are used in this chapter. The following definitions apply to the discussion herein.

What Is Accreditation?

Accreditation refers to the approval by a cognitive authority to operate a computer system within a set of parameters. As previously mentioned, the process for approving the operation of the information system may be formal, informal, or nonexistent.

Take the case of a consumer who purchases a personal computer (PC) from a vendor as an example. The proud new owner of that PC takes it home, connects all the wires in the right places, and turns it on. Probably one of the next actions that new PC owner will take is to connect the PC to an Internet service provider (ISP) by means of some type of communication device. In this scenario, the owner of that PC has unwittingly assumed the risk and responsibility for the operation of that computer within the environment the owner has selected. There is no formal approval process in place, yet the owner assumes the responsibility for the operation of that computer. This responsibility extends to any potential activity that may be initiated from that computer — even illegal activity. The owner also assumes the responsibility for the operation of that computer even if it becomes a zombie used for a distributed denial-of-service (DDoS) attack. No formal policies have been established, and no formal procedures are in place. Dependent upon the skill and experience of the owner, the computer may be correctly configured to defend against hostile actions. Additionally, if other persons, such as family members, use this computer, there may be little control over how this computer is used, what software is installed, what hostile code may be introduced, or what information is stored.

At the other end of the scale, a government entity may acquire a large-scale computer system. Many governments have taken action to introduce a formal accreditation process. The governments of Canada, Australia, and the United States, among others, have developed formal accreditation or approval processes. Where these processes are developed, information security professionals should follow those processes. They identify specific steps that must be followed in order to approve a computer system to operate. In some cases, specific civil and criminal liabilities are established to encourage the responsible authorities within those government entities to follow the process.

A huge middle ground exists between the new PC owner and the large computer system in the government entity. This middle ground encompasses small business owners, medium-sized business entities, and large corporations. The same principle applies to these entities. Somewhere within the management of the organization, someone has made a decision to operate one or more computer systems. These systems may be interconnected and may have access to the global communications network. Business owners, whether sole proprietors, partnerships, or corporations, have assumed the risk and responsibility associated with operating those

computer systems. It would be advisable for those business owners to implement a formal accreditation process, as many have. By so doing, business owners can achieve a higher level of assurance that their computer systems are part of the solution to the information security problem instead of being potential victims or contributors to the information security problem. In addition, implementing and practicing a formal accreditation process will help to show that the owners have exercised due diligence if a problem or incident should arise.

Elements of Accreditation

What are the elements of an accreditation process? One of the major advantages of having a formal accreditation process is the documentation generated by the process itself. By following a process, the necessary rules and procedures are laid down. Conscious thought is given to the risks associated with operating the identified computer system. Assets are identified and relative values are assigned to those assets, including information assets. In following the process, protection measures are weighed against the benefit to the information or asset protected, and a determination is made regarding the cost effectiveness of that protection measure. Methods to maintain the security posture of the system are identified and planned. Also, evidence is generated to help protect the business unit against potential future litigations.

Some of the documents that may be generated include Security Policy, Security Plan, Security Procedures, Vulnerability Assessment, Risk Assessment, Contingency Plan, Configuration Management Plan, Physical Security Plan, Certification Plan, and Certification Report. This is neither an inclusive nor exhaustive list. The contents of these documents may be combined or separated in a manner that best suits the environment accredited. A brief explanation of each document follows.

Security Policy. The Security Policy for the information system contains the rules under which the system must operate. The Security Policy will be one of the major sources of the system security requirements, which are discussed later in this chapter. Care should be exercised to see that statements in the Security Policy are not too restrictive. Using less restrictive rules avoids the pitfall of having to change policy every time technology changes.

An example of a policy statement is shown in Exhibit 31-1. This clearly states the purpose of the statement without dictating the method by which the policy will be enforced. A policy statement such as this one could be fulfilled by traditional user ID and password mechanisms, smart card systems, or biometric authentication systems. As technology changes, the policy does not need to be changed to reflect advances in the technology.

Exhibit 31-1. Sample security policy statement.

Users of the XYZ Information System will be required to identify themselves and authenticate their identification prior to being granted access to the information system.

Exhibit 31-2. Sample security plan statement.

A thumbprint reader will be used to identify users of the XYZ Information System. Users who are positively identified by a thumbprint will then be required to enter a personal identification number (PIN) to authenticate their identity.

Exhibit 31-3. Sample security procedure.

Log-On Procedure for the XYZ Information System

1. Place your right thumb on the thumbprint reader window so that your thumbprint is visible to the window.
2. When your name is displayed on the display monitor, remove your thumb from the thumbprint reader.
3. From the keyboard, enter your personal identification number (PIN).
4. Press **Enter** (or **Return**).
5. Wait for your personal desktop to be displayed on the display monitor.

Security Plan. The Security Plan for the information system is a fluid document. It identifies the methods employed to meet the policy. This document will change with technology. As new mechanisms are developed that satisfy Security Policy statements, they can be incorporated into the Security Plan and implemented when it is appropriate to do so within the environment.

To satisfy the Security Policy statement given in Exhibit 31-1, the Security Plan may contain a statement such as the one given in Exhibit 31-2. This Security Plan statement identifies the mechanism that will be used to satisfy the statement in the policy.

Security Procedures. Security Procedures for the information system are usually written in language intended for a less technical audience. Security Procedures may cover a wide variety of topics, from physical security to firewall configuration guidelines. They generally provide step-by-step instructions for completing a specific task. One such procedure may include a series of statements similar to those given in Exhibit 31-3. By following this procedure, the system user would successfully gain access to the computer system, while satisfying the Security Policy statement given

in Exhibit 31-1, using the mechanism identified in Exhibit 31-2. The user need not be familiar with either the Security Policy or the Security Plan when the procedure identifies the steps necessary to accomplish the task within the parameters laid down in the policy and the plan.

Vulnerability Assessment. Vulnerability Assessment is often confused with Risk Assessment. They are not the same thing. While the results of a Vulnerability Assessment and a Risk Assessment are often reported in the same document, it is important to note the differences.

A Vulnerability Assessment is that part of the accreditation process that identifies weaknesses in the security of the information system. Vulnerabilities are not limited to technical vulnerabilities such as those reported by Carnegie Mellon's Computer Emergency Response Team (CERT). Vulnerabilities could also include physical security weaknesses, natural disaster susceptibilities, or resource shortages. Any of these contingencies could introduce risk to an information system. For example, the most technically secure operating system offers little protection if the system console is positioned in the parking lot with the administrator's password taped to the monitor. Vulnerability Assessments attempt to identify those weaknesses and document them in order.

Risk Assessment. The Risk Assessment attempts to quantify the likelihood that hostile persons will exploit the vulnerabilities identified in the Vulnerability Assessment. The Risk Assessment will serve as a major source for system security requirements. There are two basic schools of thought when it comes to assessing risk. One school of thought attempts to quantify risk in terms of absolute monetary value or annual loss expectancy (ALE). The other school of thought attempts to quantify risk in subjective terms such as high, medium, or low. It is not the purpose of this chapter to justify either approach. Insight is given into these approaches so that the information security professional is apprised that risk assessment methodologies may take a variety of forms and approaches. It is left to the discretion of the information security professional and the accrediting authority — who, after all, is the one who will have to approve the results of the process to determine the best risk assessment method for the environment. The Risk Assessment will qualify the risk associated with the vulnerabilities identified in the Vulnerability Assessment so that they may be mitigated through security countermeasures or accepted by the Approving Authority.

Contingency Plan. There may be a Contingency Plan or Business Continuity Plan for the information system. This plan will identify the plans for maintaining critical business operations of the information system in the event one or more occurrences cause the information system to be inoperable or marginally operable for a specified period of time. Contingency

planning is probably of more value to businesses such as E-commerce sites or ISPs, and one is more likely to expect this type of documentation for these types of organizations. The plan should identify critical assets, operations, and functions. These are noteworthy for the information security professional in that this information identifies critical assets — both physical assets and information assets — that should be the focus of the certification effort.

Configuration Management. Configuration Management is that discipline by which changes to the system are made using a defined process that incorporates management approval. Larger installations will usually have a Configuration Management Plan. It is important to systematically consider changes to the information system in order to avoid introducing undesirable results and potential vulnerabilities into the environment. Good configuration management discipline will be reflected favorably in the certification process, as is discussed later in this chapter.

Physical Security. Again, good information security is dependent upon good physical security. Banks usually build vaults to protect their monetary assets. In like manner, physical security of information assets is a necessity. Organizations may have physical security plans to address their physical security needs. Regardless of the existence of a plan, the certification effort will encompass the physical security needs of the information system certified.

Training. No system security program can be considered complete without some form of security awareness and training provisions. The training program will address those principles and practices specific to the security environment. Training should be both formal and informal. It should include classroom training and awareness reminders such as newsletters, e-mails, posters, or signs.

Certification

Certification means many different things to many different people. The context in which one discusses certification has much to do with the meaning derived from the word. The following are some examples of how this word may be used.

Professional organizations provide certifications of individuals. A person may carry the designation of Certified Public Accountant (CPA), Certified Information Systems Security Professional (CISSP), or perhaps Certified Protection Professional (CPP). These designations, along with a multitude of others, state that the individual holding the designation has met a defined standard for the designation held.

Vendors may provide certifications of individuals on their products. The vendor offers this certification to say that an individual has met the minimum standards or level of expertise on the products for which they are certified. Examples of this type of certification include the Cisco Certified Network Associate (CCNA) or Check Point Certified Security Administrator (CCSA), among many others.

Vendors also provide certifications for products. Many vendors offer certifications of interoperability or compatibility, stating that the standards for interoperability or compatibility have been met. For example, Microsoft offers a certification for computer manufacturers that the operating system and the hardware are compatible.

Governments offer certifications for a wide variety of persons, products, processes, facilities, utilities, and many other things too numerous to list in this chapter. These government certifications state that the person, object, or process certified has met the standard as defined by that government.

Standards organizations may offer certifications. For example, a corporate entity may be certified by the standards organization to perform testing under the Common Criteria for Information Technology Security Evaluation (ISO/IEC 15408). A certified laboratory has met the standards defined by the standards organization. These certified laboratories might in turn offer certification for vendor products to given evaluation assurance levels (EALs), which range from 1 through 6. By giving a certification to a product, these certified labs are stating that the product has met the standard defined for the product.

For the purpose of the discussion within this chapter, certification refers to that part of the accreditation process in which a computer system is evaluated against a defined standard. The results of that evaluation are documented, repeatable, defendable, and reportable. The results are presented to the Approving Authority as evidence for approval or disapproval of the information system.

The common theme that runs through the world of certification is that there is a defined standard and that the standard has been met. Certification does not attempt to quantify or qualify the degree to which the standard may have been met or exceeded. Certification states that the minimum standard has been achieved.

What Is It? Simply put, system security certification is the process by which a system is measured against a defined standard. In a formal certification process, the results of that measurement are recorded, documented, and reported.

Cost versus Benefits. The direct monetary benefits to conducting a certification of the information system may not be obvious to management. The

question then becomes: Why spend the time, effort, and money if a monetary benefit is not readily obvious? Further, how does the information security professional convince management of the need for certification? To answer these questions, one needs to identify the assets protected.

- *Financial information.* Financial information assets are deserving of protection. The system may process information such as bank accounts, including their transaction balances. It may store the necessary information, such as log-on identification and passwords that would allow a would-be thief to transfer funds to points unknown. Adequate protection mechanisms may be in place to protect financial information, and conducting a certification is one of the best ways to know for sure that the security mechanisms are functioning as advertised and as expected.
- *Personal information.* Many governments have taken steps to provide their citizens with legal protection of personal and private information. In addition to legal requirements that may be imposed by a local authority, civil liabilities may be incurred if personal information is released by the information system. In the event of a civil or criminal proceeding, it would be advantageous to be able to document due diligence. Conducting a certification is a good way to show that due diligence has been exercised.
- *Corporate information.* Information that is considered proprietary in nature or company confidential needs to be protected for reasons determined by managers and owners. This information has value to the business interests of the corporation, agency, or entity. For this reason, certification should be considered part of the approval process in order to verify that the installed security mechanisms are functioning in such a manner as to provide adequate protection to that information. Serious damage to the business interests of the corporation, agency, or entity may be incurred if corporate information were to fall into the wrong hands.
- *Legal requirements.* Laws are constantly changing. Regulatory bodies may change the rules. Conducting a certification of the information system help to keep managers one step ahead of the changing environment and perhaps avoid fines and penalties resulting from a failure to meet legal requirements.

Why Certify? It is left to the reader to determine the best justification for proceeding with the certification part of the accreditation process. Remember the earlier discussion regarding the approval to operate an information system? In that discussion, it was discovered that approval and certification are done either through a conscious effort, be it formal or informal, or by default. Choosing to do nothing is not a wise course of action. The fact that you have a firewall, a "secure" operating system, or

other security measures installed does not ensure that those features and functions are operating correctly. Many times, the certification process has discovered that these security measures have provided only a false sense of security and that they did not provide any real protection to the information system.

ROLES AND RESPONSIBILITIES

Once the decision has been made to proceed with a certification, it is necessary to assemble a team of qualified individuals to perform the certification. It can be performed in-house or may be outsourced. In the paragraphs that follow, a suggested list of Roles and Responsibilities for the Certification Test Team are presented. The roles and responsibilities do not necessarily require one person for each role. Roles may be combined or modified to meet the requirements of the environment. Resource availability as well as the size and complexity of the system evaluated will drive the decision on the number of personnel needed.

- *Approving Authority.* The Approving Authority is the person legally responsible for approving the operation of the information system. This person will give the final approval or accreditation for the information system to go into production. The authority of this individual may be derived from law or from business directive. This person will have not only the legal authority to assume the residual risk associated with the operation of the information system, but will also assume the civil and criminal liabilities associated with the operation of the information system.
- *Certifying Authority.* The Certifying Authority or *Certifier* is the individual responsible for approving, certifying, and reporting the results of the certification. This person is sometimes appointed by the Approving Authority but most certainly has the full faith and support of those in authority to make such an appointment within the agency, business, or corporation. This person must possess a sufficient level of technical expertise to understand the results presented. This individual will function on behalf of the Approving Authority, or those having authority to make the appointment, in all matters pertaining to certification as it relates to the accreditation process. This individual may also be called upon to contribute to a recommendation to the Approving Authority regarding approval or disapproval of the information system to operate.
- *Test Director.* The Test Director operates under the direction of the Certifying Authority. This individual is responsible for the day-to-day conduct of the certification test. The Test Director ensures that the tests are conducted as prescribed and that the results are recorded, collected, preserved, and reported. Depending on the size and complexity of the information system certified, the Test Director may be

required to provide periodic updates to the Certifying Authority. Periods may be weekly, daily, or perhaps hourly, if needed. The Test Director will ensure that all tests are performed in accordance with the test plan.

- *System Manager.* The System Manager must be an integral part of the certification process. It is impossible for anyone to know everything about a given information system, even if the system is well documented. The System Manager will usually have the most intimate and current knowledge of the information system. This individual will make significant contributions to preparing test scenarios and test scripts necessary to document the test plan. The System Manager, or a designee of the System Manager, will actually perform many of the tests prescribed in the test plan.
- *Test Observer.* Test Observers may be required if the information system is of sufficient size and complexity. At a minimum, it is recommended that there be at least one test observer to capture and record the results of the test as they are performed. Test Observers operate under the direction of the Test Director.
- *Test Recorder.* The Test Recorder is responsible to the Test Director for logging and preserving the test results, evidence, and artifacts generated during the test. In the case of smaller installations, the Test Recorder may be the same person as the Test Director. In larger installations, the Test Recorder may be more than one person. The size and complexity of the information system, as well as resource availability, will dictate the number of Test Recorders needed.
- *IV&V.* Independent Verification and Validation (IV&V) is recommended as a part of all certification tests. IV&V is a separate task not directly associated with the tasks of the Certifying Authority or the Certification Test Team. The IV&V is outside the management structure of the Certifying Authority, the Test Director, and their teams. Under ideal conditions, IV&V will provide a report directly to the Approving Authority. In this manner, the Approving Authority will have a second opinion regarding the security of the information system certified. IV&V will have access to all the information generated by the Certification Test Team and will have the authority to direct deviations from the test plan. At the discretion of the Approving Authority, the Certification Test Team may not necessarily have access to information generated by the IV&V. The IV&V task may be outsourced if inadequate resources are not available in-house.

DOCUMENTATION

With the Certification Test Team in place and the proper authorities, appointments, and reporting structure established, it is now time to begin the task of generating a Certification Test Plan. The Certification Test Plan

covers preparation and execution of the certification; delineates schedules and resources for the certification; identifies how results are captured, stored, and preserved; and describes how the Certifying Authority reports the results of the certification to the Approving Authority.

Policy

Security requirements are derived from a variety of sources. There was a discussion of Security Policies and Security Plans earlier in this chapter. Policy statements are usually found in the Security Policy; however, information security professionals should be watchful for policy statements that appear in Security Plans. Often, these are not separate documents, and the Security Plans for the information system are combined with the policy into a single document.

Policy statements are also derived from public law, regulations, and policies. Information security professionals need to be versed in the local laws, regulations, and policies that affect the operations of information systems within the jurisdiction in which they operate. Failing to recognize the legal requirements of local governments could lead to providing false certification results by certifying a system that is operating illegally under local law. For example, some countries require information systems connecting to the Internet to be routed through a national firewall, making it illegal to connect directly to an ISP.

Plans

Security Plans may contain policy statements, as mentioned previously. Security Plans may also address future implementations of security measures. Information security professionals need to carefully read Security Plans and test only those features that are supposed to be installed in the current configuration.

The Certification Test Plan will also ensure that Physical Security, Configuration Management, and Contingency or Emergency Plans are being followed. The absence of these plans must be noted in the Certification Test Report, as the lack of such planning may affect the decision of the Approving authority.

Procedures

Any Security Procedures that were generated as a part of the overall security program for the information system must be tested. The goal of testing these procedures is to ensure that user and operator personnel are aware of the procedures, know where the procedures are kept, and that the procedures are followed. Occasionally it is discovered that the procedures are not followed and, if not followed, the procedures are worthless. The

Approving Authority must be made aware of this fact if discovered during the test.

Risk Assessment

The Risk Assessment is also a major source for security requirements. The Risk Assessment should identify the security countermeasures and mechanisms chosen to mitigate the risk associated with identified vulnerabilities. The Risk Assessment may also prioritize the implementation of countermeasures, although this is normally done in the Security Plan.

DETERMINING REQUIREMENTS

Here is where the hard work begins. Up to this point in the process, available and appropriate documentation has been collected, a Certification Test Team has been appointed and assembled, and the beginnings of a Certification Test Plan have been initiated.

So what is covered by the Certification Test? It tests security requirements. For certification purposes, testing is not limited to technical security requirements of the information system. Later in this chapter, there is a discussion of categorization of requirements; however, before requirements can be categorized, they must be identified, derived, and decomposed. This phase of the certification process may be called the Requirements Analysis Phase. During this phase, direct and derived requirements are identified. The result of this phase is a Requirements Matrix that traces the decomposed requirements to their source.

Direct requirements are those clearly identified and clearly stated in a policy document. Going back to Exhibit 31-1, a clear requirement is given for user identification and subsequent authentication.

Derived requirements are those requirements that cannot be directly identified in a policy statement; rather, they must be inferred or derived from a higher-level requirement. Using Exhibit 31-2 as an example, the need for a thumbprint reader to be installed on the information system must be derived because it is not stated directly in the plan.

Requirements are discussed in the following paragraphs in general order of precedence. The order of precedence given here is not intended to be inflexible; rather, it can be used as a guideline that should be tailored to fit the environment in which it is used.

Legal

Legal requirements are those requirements promulgated by the law of the land. If, in the case of Exhibit 31-2, the law required the use of smart cards instead of biometrics to identify users, then the policy statement given in Exhibit 31-2 could be considered an illegal requirement. It is the

responsibility of the information security professional to be aware of the local laws, and it would be the responsibility of the information security professional to report this inconsistency. The Approving Authority would decide whether to accept the legal implication of approving the information system to operate in the current configuration.

Regulatory

The banking industry is among the most regulated industries in the world. The banking industry is an example of how government regulations can affect how an information system will function. The types of industries regulated and the severity of regulation within those industries vary widely. Information security professionals need to be familiar with the regulatory requirements associated with the industry in which they operate.

Local

Local requirements are the policies and requirements implemented by the entity, agency, business, or corporation. These requirements are usually written in manuals, policies, guidance documents, plans, and procedures specific to the entity, agency, business, or corporation.

Functional

Sometimes security requirements stand in the way of functional or mission requirements, and vice versa. Information security professionals need to temper the need to protect information with the need to get the job done. For this reason, it is recommended that security requirements be tested using functional and operational scenarios. By so doing, a higher level of assurance is given that security features and mechanisms will not disrupt the functional requirements for the information system. It allows the information security professional to evaluate how the security features and mechanisms imposed on the information system may affect the functional mission.

Operational

Operational considerations are also an important part of the requirements analysis. Operational requirements can sometimes be found in the various plans and procedures. It is necessary to capture these requirements in the Requirements Matrix also, so that they can be tested as part of the overall information security program. Operational requirements may include system backup, contingencies, emergencies, maintenance, etc.

Requirements Decomposition

Decomposing a requirement refers to the process by which a requirement is broken into smaller requirements that are quantifiable and testable. Each

Exhibit 31-4. Sample decomposed policy requirements.

1.1 Users of the XYZ Information System will be required to identify themselves prior to being granted access to the information system.
2.2 Users of the XYZ Information System will be required to authenticate their identity prior to being granted access to the information system.
2.1 A thumbprint reader will be used to identify users of the XYZ Information System.
2.1.a Thumbprint readers are installed on the target configuration.
2.2 Users who are positively identified by a thumbprint will then be required to enter a personal identification number (PIN) to authenticate their identification.
2.2.a Keyboards are installed on the target configuration.

decomposed requirement should be testable on a pass-or-fail basis. As an example, Exhibit 31-1 contains at least two individual testable requirements. Likewise, Exhibit 31-2 contains at least two individual testable requirements. Exhibit 31-4 shows the individual decomposed requirements.

Requirements Matrix

A Requirements Matrix is an easy way to display and trace a requirement to its source. It provides a column for categorization of each requirement. The Matrix also provides a space for noting the evaluation method that will be used to test that requirement and a space for recording the results of the test. The following paragraphs identify column heading for the Requirements Matrix and provide an explanation of the contents of that column. Exhibit 31-5 is an example of how the Requirements Matrix may appear.

Category. Categories may vary, depending upon the environment of the information system certified. The categories listed in the following paragraphs are suggested as a starting point. The list can be tailored to meet the needs of the environment. Further information on security services and mechanisms listed in the subsequent paragraphs can be found in ISO 7498-2, *Information Processing Systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture (1989).* The following definitions are attributed to ISO 7498-2. Note that a requirement may fit in more than one category.

- *Security services.* Security services include authentication, access control, data confidentiality, data integrity, and nonrepudiation.
  - — *Authentication.* Authentication is the corroboration that a peer entity is the one claimed.
  - — *Access control*. Access control is the prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner.

Exhibit 31-5. Example requirements matrix.

Req. No.	Category	Source Reference	Stated Requirement	Evaluation Method	Test Procedure	Pass	Fail
1	I&A	XYZ Security Policy	Users of the XYZ Information System will be required to identify themselves prior to being granted access to the information system	Test	IA002S		
2	I&A	XYZ Security Policy	Users of the XYZ Information System will be required to authenticate their identify prior to being granted access to the information system	Test	IA002S		
3	I&A	XYZ Security Plan	A thumbprint reader will be used to identify users of the XYZ Information System	Demonstrate	IA003S		
4	Architecture	Derived	Thumbprint readers are installed on the target configuration	Observation	AR001A		
5	I&A	XYZ Security Plan	Users who are positively identified by a thumbprint will then be required to enter a personal identification number (PIN) to authenticate their identification	Demonstrate	IA003S		
6	Architecture	Derived	Keyboards are installed on the target configuration	Observation	AR001A		

— *Data confidentiality.* Data confidentiality is the property that information is not made available or disclosed to unauthorized individuals, entities, or processes.
— *Data integrity.* Data integrity is the property that data has not been altered or destroyed in an unauthorized manner.
— *Non-repudiation.* Non-repudiation is proof of origin or receipt such that one of the entities involved in a communication cannot deny having participated in all or part of the communication.

- *Additional Categories.* The following categories are not defined in ISO 7498. These categories, however, should be considered as part of the system security Certification Test.
  — *Physical security.* Physical security of the information system is integral to the overall information security program. At a minimum, the Certification Test should look for obvious ways to gain physical access to the information system.
  — *Operational security.* Operational security considerations include items such as backup schedules and their impact on the operational environment. For example, if a system backup is performed every day at noon, the Certification Test should attempt to determine if this schedule has an operational impact on the mission of the system, remembering that availability of information is one of the tenets of sound information security practice.
  — *Configuration management.* At a minimum, the Certification Test should select one change at random to determine that the process for managing changes was followed.
  — *Security awareness and training.* At a minimum, the Certification Test should randomly select user and operator personnel to determine that there is an active Security Awareness and Training Program.
  — *System security procedures.* At a minimum, the Certification Test should select an individual at random to determine if the System Security Procedures are being followed.
  — *Contingency planning.* The Certification Test should look for evidence that the Contingency Plan is routinely tested and updated.
  — *Emergency Planning.* The Certification Test should determine if adequate and appropriate emergency plans are in place.
- *Technical.* Technical controls are those features designed into or added onto the computer system that are intended to satisfy requirements through the use of technology.
  — *Access controls.* The technical access control mechanisms are those that permit or deny access to systems or information based on rules that are defined by system administration and management personnel. This is the technical implementation of the access control security service.

— *Architecture.* Technical architecture is of great importance to the Certification Test process. Verifying the existence of a well-developed system architecture will provide assurance that backdoors into the system do not exist unless there is a strong business case to support the backdoor, and then only if it is properly secured.
— *Identification and authentication.* Identification and authentication is the cornerstone of information security. The Certification Test Plan must thoroughly detail the mechanisms and features associated with the process of identifying a user or process, as well as the mechanisms and features associated with authenticating the identity of the user or process.
— *Object reuse.* In most information systems, shared objects, such as memory and storage, are allocated to subjects (users, processes, etc.) and subsequently released by those subjects. As subjects release objects back to the system to be allocated to other subjects, residual information is normally left behind in the object. Unless the object is cleared of its residual content, it is available to a subject that is granted an allocation to that object. This situation creates insecurity, particularly when the information may be passed outside the organization, thereby unintentionally releasing sensitive information to the public that resides in the file slack space. Clearing the object, either upon release of the object or prior to its allocation to a subject, is the technique used to prevent this insecurity.

The test facilities necessary to test shared resources for residual data may not be available to the information security professional. To test this feature, the Certification Test Team may be required to seek the services of a certified testing facility. At a minimum, the Certification Test Plan should determine if this feature is available on the system under test and also determine if this feature is enabled. If these features have been formally tested by a reputable testing facility, their test results may be leveraged into the local test process.

On a related subject, data remanence may be left on magnetic storage media. That is, the electrical charges on given magnetic media may not be completely discharged by overwriting the information on the media. Sophisticated techniques can be employed to recover information from media, even after it has been rewritten several times. This fact becomes of particular concern when assets are either discarded or transferred out of the organization. Testing this feature requires specialized equipment and expertise that may not be available within the Certification Test Team. At a minimum, the Certification Test Plan should determine if procedures and policies are in place to securely erase all data remanence from media upon destruction or transfer, through a process known as degaussing.

Audit

Auditing is the technical security mechanism that records selected actions on the information system. Audit logs must be protected from tampering, destruction, or unauthorized access. The Certification Test Plan should include a test of the audit features of the system to determine their effectiveness.

System Integrity

Technical and nontechnical features and mechanisms should be implemented to protect the integrity of the information system. Where these features are implemented, the Certification Test Plan should examine them to determine their adequacy to meet their intended results.

Security Practices and Objectives

Test categories that address security practices and objectives may be found in the International Organization for Standards (ISO) and the International Electrotechnical Commission (IEC) from their adaptation of British Standard (BS) 7799, which was published as ISO/IEC International Standard (IS) 17799, *Information Technology — Code of Practice for Information Security Management,* dated December 2000. ISO/IEC IS 17799 recommends standards for and identifies several objectives that are elements of information security management. In keeping with the spirit of the IS, the elements herein identified are recommendations and not requirements. These elements can be tailored to adapt to the environment in which the test is executed. For a further explanation and detailed definition of each of these categories, the reader is referred to ISO/ISE IS 17799.

Exhibit 31-6 lists the various security services and mechanisms from ISO 7498-2 and the various security management practices and objectives from ISO 17799.

Source

Each requirement must be traceable to its source. The source may be any one or more of the documents identified above.

Specific Requirement. Each decomposed requirement will be listed separately. This allows for easy reference to the individual requirement.

Evaluation Method. This column identifies the method that will be used to evaluate the requirement. Possible evaluation methods include *Test, Demonstration, Inspection, Not Evaluated,* or *Too General*.

- *Test.* This evaluation method calls for the requirement to be tested on a system of the same configuration as the live system. Testing on a live system is not recommended; however, if resource constraints necessitate

Exhibit 31-6. Security services, practices, and objectives.

SECURITY SERVICES (ISO 7498-2)
Authentication
- Peer entity authentication
- Data origin authentication

Access control
Data confidentiality
- Connection confidentiality
- Connectionless confidentiality
- Selective field confidentiality
- Traffic flow confidentiality

Data integrity
- Connection integrity with recovery
- Connection integrity without recovery
- Selective field connection integrity
- Connectionless integrity
- Selective field connectionless integrity

Non-repudiation
- Non-repudiation with proof of origin
- Non-repudiation with proof of delivery

SPECIFIC SECURITY MECHANISMS (ISO 7498-2)
Encipherment
Digital signature
Access control
Data integrity
Authentication exchange
Traffic padding
Routing control
Notarization

PERVASIVE SECURITY MECHANISMS (ISO 7498-2)
Trusted functionality
Security labels
Event detection
Security audit trail
Security recovery
Security policy
- Information security policy document
- Review and evaluation

Organizational Security
- Information security infrastructure
  - Management information security forum
  - Information security coordination
  - Allocation of information security responsibilities
  - Authorization process for information processing facilities
  - Specialist information security advice
  - Cooperation between organizations
  - Independent review of information security

Exhibit 31-6. Security services, practices, and objectives (Continued).

- Security of third-party access
  - Identification of risks from third-party access
  - Security requirements in third-party contracts
- Outsourcing
  - Security requirements in outsourcing contracts

Asset Classification and Control

- Accountability for assets
  - Inventory of assets
- Information classification
  - Classification guidelines
  - Information labeling and handling

Personnel Security

- Security in job definition and resourcing
  - Including security in job responsibilities
  - Personnel screening and policy
  - Confidentiality agreements
  - Terms and conditions of employment
- User training
  - Information security education and training
- Responding to security incidents and malfunctions
  - Reporting security incidents
  - Reporting security weaknesses
  - Reporting security malfunctions
  - Learning from incidents
  - Disciplinary process

Physical and Environmental Security

- Secure areas
  - Physical security perimeter
  - Physical entry controls
  - Security offices, rooms and facilities
  - Working in secure areas
  - Isolated delivery and loading areas
- Equipment security
  - Equipment sitting and protection
  - Power supplies
  - Cabling security
  - Equipment maintenance
  - Security of equipment off-premises
  - Secure disposal or reuse of equipment
- General controls
  - Clear desk and clear screen policy
  - Removal of property

Communications and Operations Management

- Operational procedures and responsibilities
  - Documented operating procedures
  - Operational change control
  - Incident management procedures
  - Segregation of duties

Exhibit 31-6. Security services, practices, and objectives (Continued).

- Separation of development and operational facilities
- External facilities management
- System planning and acceptance
  - Capacity planning
  - System acceptance
- Protection against malicious software
  - Controls against malicious software
- Housekeeping
  - Information backup
  - Operator logs
  - Fault logging
- Network management
  - Network controls
- Media handling and security
  - Management of removable computer media
  - Disposal of media
  - Information handling procedures
  - Security of system documentation
- Exchanges of information and software
  - Information and software exchange agreements
  - Security of media in transit
  - Electronic commerce security
  - Security of electronic mail
  - Security of electronic office systems
  - Publicly available systems
  - Other forms of information exchange

Access control

- Business requirements for access control
  - Access control policy
- User access management
  - User registration
  - Privilege management
  - User password management
  - Review of user access rights
- User responsibilities
  - Password use
  - Unattended user equipment
- Network access control
  - Policy on use of network services
  - Enforced path
  - User authentication for external connections
  - Node authentication
  - Remote diagnostic port protection
  - Segregation in networks
  - Network connection control
  - Network routing control
  - Security of network services
- Operating system access control
  - Automatic terminal identification

Exhibit 31-6. Security services, practices, and objectives (Continued).

- Terminal log-on procedures
- User identification and authentication
- Password management system
- Use of system utilities
- Duress alarm to safeguard users
- Terminal timeout
- Limitation of connection time

Application access control
- Information access restriction
- Sensitive system isolation

Monitoring system access and use
- Event logging
- Monitoring system use
- Clock synchronization

Mobile computing and teleworking
- Mobile computing
- Teleworking

Systems Development and Maintenance

Security requirements of systems
- Security requirements analysis and specification

Security in application systems
- Input data validation
- Control of internal processing
- Message authentication
- Output data validation

Cryptographic controls
- Policy on the use of cryptographic controls
- Encryption
- Digital signatures
- Non-repudiation services
- Key management

Security of system files
- Control of operational software
- Protection of system test data
- Access control to program source library

Security in development and support processes
- Change control procedures
- Technical review of operating system changes
- Restriction on changes to software packages
- Covert channels and Trojan code
- Outsourced software development

Business Continuity Management

Aspects of business continuity management
- Business continuity management process
- Business continuity and impact analysis
- Writing and implementing continuity plans
- Business continuity planning framework
- Testing, maintaining, and reassessing business continuity plans

Exhibit 31-6. Security services, practices, and objectives (Continued).

Compliance
Compliance with legal requirements
Identification of applicable legislation
Intellectual property rights
Safeguarding of organizational records
Data protection and privacy of personal information
Prevention of misuse of information processing facilities
Regulation of cryptographic controls
Collection of evidence
Reviews of security policy and technical compliance
Compliance with security policy
Technical compliance checking
System audit considerations
System audit controls
Protection of system audit tools

testing on the live system, all parties must be advised and agree to the risk associated with that practice.

- *Demonstration.* When testing is inappropriate, a demonstration may be substituted. For example, if a requirement calls for hard-copy output from the information system to be marked in a specific manner, personnel associated with the operation of the system could easily demonstrate that task.
- *Inspection.* Inspection is an appropriate test method for requirements such as having visiting personnel register their visit or a requirement that personnel display an identification card while in the facility.
- *Not evaluated.* This method should only be chosen at the direction of the Approving Authority. There are occasions where testing a requirement may cause harm to the system. For example, testing a requirement to physically destroy a hard disk prior to disposal would cause an irrecoverable loss. In cases such as these, the Approving Authority may accept the process as evidence that the requirement is met.
- *Too general.* Occasionally, requirements cannot be quantified in a pass-or-fail manner. This is usually due to a requirement that is too general. An example might be a requirement that the information system is operated in a secure manner. This requirement is simply too general to quantify and test.

Test Procedure. Identify the test procedure that is used to test the requirement. Building test scenarios and test scripts is discussed later in this chapter. The combination of these items forms a test procedure. The test procedures are identified in this column on the matrix.

Pass or Fail. The last column is a placeholder for a *pass* or *fail* designator. The Test Recorder will complete this column after the test is executed.

BUILDING A CERTIFICATION TEST PLAN

The Test Team has been established and appointed, and requirements have been identified and broken down into individual testable requirements. The Certification Test Plan can now be written. The Certification Test Plan will address test objectives and schedules; and it will provide a method for executing the individual tests and for recording, compiling, and reporting results. To maintain integrity of the system functional requirements, tests can be structured around real-life functional and operational scenarios. By so doing, the Certifying Authority and the Approving Authority can obtain a higher level of assurance that the system will not only be a more secure system but also will meet its operational mission requirements. Remember: Certification Testing is designed to show that the system meets the minimum requirements — not to show that security features and mechanisms are all installed, enabled, and configured to their most secure settings. This may seem somewhat contrary to good security practice; however, it is not. Most of the security engineering and architecture work would have been accomplished in the initial design and implementation phases for the system. Of course, it is incumbent upon information security professionals to identify those practices that introduce vulnerabilities into the system. Information security professionals must identify those weaknesses before entering into a Certification Test. Under these conditions, the test would proceed only after the managers and owners of the system agree to accept the risk associated with the vulnerabilities. The goal is to avoid any surprises introduced in the final report on the Certification Test.

Introduction and Background

The Certification Test Plan should begin with some introductory and background information. This information would identify the system under test, its mission and purpose. The Plan should identify the reasons for conducting the test, whether for initial accreditation and approval of the system or as part of an ongoing information security management program. This provides historical information to those who may wish to review the results in the future, and it also provides a framework for persons who may be involved in Independent Verification and Validation (IV&V) efforts and who may not be familiar with the system tested. Adequate detail should be provided to satisfy these two goals.

The Certification Test Plan should define its purpose. Providing a defined purpose will help to limit the scope of the Test Plan in order to avoid either testing too little, thereby rendering the test evidence inadequate to support conclusions in the test report, or testing too much, thereby rendering the test unmanageable and the results suspect.

The scope of the test should be identified. That is, the configuration boundaries should be defined and the limit of requirements and standards should be identified. These factors would have been identified prior to reaching this point in the process. It is important to document them in the Certification Test Plan because the supporting documentation upon which this plan is built may change in the future, causing a loss of the current frame of reference. For example, if a UNIX-based system is tested today, and it is retrofitted with a Windows-based system next year, the results of the test are not valid for the new configuration. If the test plan fails to identify its own scope, there is no basis for determining that the test results are still valid.

Assumptions and Constraints

Assumptions and Constraints must be identified. These items will cover topics like the availability of a test suite of equipment, disruption of mission operations, access to documentation such as policies and procedures, working hours for the test team, scheduling information, access to the system, configuration changes, etc.

Test Objectives

High-level Test Objectives are identified early in the certification test plan. These objectives should identify the major requirements tested. Test Scenarios will break down these overall objectives into specific requirements, so there is no need to be detailed in this section of the plan. High-level objectives can include items such as access control, authentication, audit, system architecture, system integrity, facility security management, standards, functional requirements, or incident response. Remember that a Requirements Matrix has already been built and that the Test Scenarios, discussed later in this chapter, will provide the detailed requirements and detailed test objectives. Here the reader of the Certification Test Plan is given a general idea of those objectives to which the system will be tested.

System Description

This section of the Certification Test Plan should identify and describe the hardware, software, and network architecture of the system under test. Configuration drawings and tables should be used wherever possible to describe the system. Include information such as make and model number, software release and version numbers, cable types and ratings, and any other information that may be relevant to conducting of the test.

Test Scenario

The next step in developing the Certification Test Plan is to generate Test Scenarios. The scenario can simulate real operational conditions. By

Exhibit 31-7. Sample test scenario.

Title:	Identification and Authentication Procedure
Number:	IA002
Purpose:	In this test procedure, a user will demonstrate the procedures for gaining access to the XYZ Information System. Evaluators and observers will verify that the procedure is followed as documented. This scenario is a prerequisite to other test scenarios that require access to the system and will, therefore, be tested many times during the course of the certification test.
Team Members Required:	Evaluators, Observers, User Representative, IV&V
Required Support:	User Representative
Evaluation Method:	Observation, Demonstration
Entrance Criteria:	(Identify tests that must be successfully completed before this test can begin)
Exit Criteria:	(Identify how the tester will know that this test is completed)
Test Scripts Included:	IA001S

Procedure:

1. Power on the workstation, if not already powered on.
2. Demonstrate the proper method of identifying the user to the system.
3. Demonstrate the proper method of authenticating the identified user to the system.
4. Observers will verify that all steps in the test script are executed.
5. Evaluators will complete the attached checklist.
6. Completed scripts, checklists, and observer notes will be collected and transmitted to the test recorder.

so doing, functional considerations are included within the Certification Test Plan. The members of the Test Team should be familiar with the operational and functional needs of the system in order to show that the security of the system does not adversely impact the functional and operational considerations. This is the reason system administration and system user representatives are members of the Test Team. The Test Scenario should identify the Test Objective and expected results of the scenario.

Using the example presented earlier in this chapter, an example Test Scenario is shown as Exhibit 31-7. In this scenario, user identification and authentication procedures are tested by having a user follow the published procedure to accomplish that task.

Test Script

The Test Scenario identifies Test Scripts that are attached to the Scenario. The persons actually executing the test procedures use Test Scripts. Persons most familiar with the operation of the system should prepare

Exhibit 31-8. Sample test script.

Title	Identification and Authentication Procedure	
Test Script Number:	IA002S	
Equipment:	Standard Workstation	
Step:	**Script**	**Pass/Fail**
1. Power on workstation	1.1. Determine if workstation is powered on.	
	1.2. If yes, go to step 2.	
	1.3. Power on workstation and wait for log-in prompt.	
2. Identify user to system	2.1. The user will place the right thumb on the thumbprint reader.	
	2.2. Wait for system to identify the user.	
3. Authenticate identity	3.1. The user will enter the personal identification number (PIN) using the keyboard.	
	3.2. Wait for authentication information to be verified by the system.	

Test Scripts. These people know how the system functions on a day-to-day basis. Depending on the stage of development of the system, those persons may be developers, system administrators, or system users. The Test Script will provide step-by-step instructions for completing the operations prescribed in the Test Scenario. Each step in the Script should clearly describe the expected results of the step. This level of detail is required to assure reproducibility. Test Results are worthless if they cannot be reproduced at a later date. Exhibit 31-8 is an example of a Test Script.

Test Results

The results of each individual test are recorded as the test is executed. This is the reason for adding the third column on the Test Script. This column is provided for the observer and evaluator to indicate that the step was successfully completed. Additionally, space should be provided or a separate page attached for observers and evaluators to record any thoughts or comments they feel may have an impact on the Certification Test Report. It is not necessary that all the observers agree on the results, but it is necessary that the team be as thorough as necessary to document what happened, when it happened, and whether did it happen as expected. This information will be consolidated and presented in the Certification Test Report, which becomes the basis for recommending certification of the system.

DOCUMENTING RESULTS

The next step in the process of system security certification is to document the results of the Certification Test. Remember that this document will become part of the accreditation package and must be presented fairly and completely. Security professionals should not try to skew the results of the test in favor of any party involved in the certification or accreditation process. Results must be presented in an unbiased fashion. This is necessary in order to preserve the security of the system and also the integrity of the profession.

Report

The Certification Test Report must be able to stand on its own. Sufficient information should be presented that the reader of the report does not need to refer to other documents to understand the report. As such, the report will document the purpose and scope of the test. It will identify mode of operation chosen for the system, the configuration and the perimeter of the system under test, and who was involved and the roles each person played. It will summarize the findings. Finally, the Certification Test Report will state whether the system under test meets the security requirements. Any other appropriate items should be included, such as items identified as meeting requirements but not meeting the security goals and objectives. For example, a system could have a user identification code of *userid*, and a password of *password*. While this may meet the requirement of having a username and password assigned to the user, it fails to meet security objectives because the combination is inadequate to provide a necessary level of protection to the system. The Certification Test Report should identify this as a weakness and recommend that a policy for username and password strength and complexity be adopted.

Completed Requirements Matrix

Among the various attachments to the Certification Test Report is the completed Requirements Matrix. The Test Recorder would transfer the results of the Test Scenarios to the Requirements Matrix. Presenting this information in this manner allows someone reviewing the report to easily scan the table for requirements that have not been met. These unsatisfied requirements will be of great interest to the Approving Authority because the legal and civil liabilities of accepting the risk associated with unsatisfied requirements will belong to that person. Exhibit 31-9 is an example of a completed Requirements Matrix.

RECOMMENDATIONS

Finally, the Certification Test Report will provide sufficient justification for the recommendations it makes. The report could make recommendations to the Certifying Authority, if prepared by the Test Director or person

Exhibit 31-9. Completed requirements matrix.

Req. No.	Category	Source Reference	Stated Requirement	Evaluation Method	Test Procedure	Pass	Fail
1	I&A	XYZ Security Policy	Users of the XYZ Information System will be required to identify themselves prior to being granted access to the information system.	Test	IA002S	X	
2	I&A	XYZ Security Policy	Users of the XYZ Information System will be required to authenticate their identity prior to being granted access to the information system.	Test	IA002S	X	
3	I&A	XYZ Security Plan	A thumbprint reader will be used to identify users of the XYZ Information System.	Demonstrate	IA003S	X	
4	Architecture	Derived	Thumbprint readers are installed on the target configuration.	Observation	AR001A	X	
5	I&A	XYZ Security Plan	Users who are positively identified by a thumbprint will then be required to enter a personal identification number (PIN) to authenticate their identification.	Demonstrate	IA003S	X	
6	Architecture	Derived	Keyboards are installed on the target configuration.	Observation	AR001A	X	

of similar capacity. The report could make recommendations to the Accrediting Authority, if prepared by the Certifying Authority. Regardless of the audience or the author of the report, it will contain recommendations that include those identified in the following paragraphs.

Certify or Not Certify

The recommendation either to certify or not certify is the professional opinion of the person or persons preparing the report. Just as a recommendation to certify must be justified by the material presented in the report, so should a recommendation not to certify. Documentation and justification are the keys to successfully completing a Certification Test. If it is discovered at this point in the process that there is insufficient information to justify the conclusion, it would be necessary to regress and acquire the necessary information. Security professionals must be prepared to justify the conclusion and provide the documentation to support it.

Meets Requirements but Not Secure

On rare occasions, it is necessary to identify areas of weakness that meet the requirements for the system but fail to satisfy system security objectives. Usually these are identified early in the certification process, when policies are reviewed and requirements are decomposed. If, however, one or more of these items should make it through the certification process, it would be incumbent upon security professionals to identify them in the Certification Test Report.

Areas to Improve

No system security approach is perfect. Total security is unachievable. With this in mind, the security professional should identify areas that could be improved. Certainly, if the recommendation were not to certify, this section of the Certification Test Report would include those items that need to be fixed before certification could be recommended. Likewise, if items are identified that do not meet the security objectives, a recommendation should be made regarding repairing the policies that allowed this situation to occur, along with a recommendation for improving the security of the system by fixing the technology, process, or procedure that is errant. Also, if the recommendation is to certify the system, all security approaches could use some improvements. Those items and recommendations should be identified in the report.

Recertification Recommendations

Conditions under which the certification becomes invalid should be identified in the Certification Test Report. Often these conditions are dictated by policy and are usually linked to the passage of time or to the

reconfiguration of the system. Regardless of whether these conditions are identified in the policies for the system, the Certification Test Report should identify them. A major reason for including this in the report is so that future uses of its contents will be within the context it is intended. For example, it would be inappropriate to use the results of the Certification Test from five years ago, when the hardware, software, and operating systems were different, to justify certification of the system as it exists today.

DISSENTING OPINIONS

Certification is not an exact science. Occasionally, there is a difference of opinion regarding the conclusions drawn against the evidence presented. The Certification Test Report must report those dissenting opinions because it is necessary that the Accrediting Authority have as much information as is available before formulating an informed opinion. Every effort should be made to resolve the difference of opinion; however, if a resolution cannot be found, it is the obligation of the security professional to report that difference of opinion.

Independent Verification and Validation (IV&V) will submit the report directly to the Accrediting Authority without consulting the Certifying Authority or the Certification Test Team. This independent opinion gives the Accrediting Authority another perspective on the results of the Certification Test results. There should be little, if any, difference between the findings in the Certification Test Report and those of the IV&V if the test was properly structured and executed.

FINAL THOUGHTS

Final thoughts are similar to initial thoughts. Computer systems large and small, or anywhere in between, are approved for use and are certified either by conscious and deliberate effort or blindly by default. It would be better to make an informed decision rather than rely on luck or probabilities. Granted, there is a possibility that the system will never be subject to attacks, whether physical or electronic. Taking that chance leaves one exposed to the associated legal, civil, or criminal liabilities. Security professionals should insist on some type of certification, formal or informal, before putting any computer system into production and exposing it to the communication world.

ABOUT THE AUTHOR

Kevin J. Davidson, CISSP, is a senior staff systems engineer with Lockheed Martin Mission Systems in Gaithersburg, Maryland. He earned a B.S. in computer science from Thornewood University in Amsterdam, the Netherlands. He has developed and performed certification tests for the U.S. Department of Defense and the U.S. Department of Justice.

103

System Development Security Methodology

Ian Lim, CISSP and Ioana V. Carastan, CISSP

Many organizations have a System or Software Development Lifecycle (SDLC) to ensure that a carefully planned and repeatable process is used to develop systems. The SDLC typically includes stages that guide the project team in proposing, obtaining approval for, generating requirements for, designing, building and testing, deploying, and maintaining a system. However, many SDLCs do not take security into consideration adequately, resulting in the productionalization of insecure systems. Even in cases where there are security components in the SDLC, security is oftentimes the sacrificial lamb in a compressed project delivery timeframe. This neglect brings risk to the organization, and creates an operational burden on the IT staff, resulting in the need for costly, difficult, and time-consuming security retrofitting. In a climate where the protection of information is increasingly tied to an organization's integrity, security needs to be strongly coupled with the system development process to ensure that new systems maintain or improve the current security level of the organization.

This chapter describes a System Development Security Methodology (SDSM), which is a *modus operandi* for incorporating security into the system development process. The SDSM is designed to be an extension, not a replacement, of an organization's preexisting SDLC. This pairing and differentiation is meant both to complement and draw attention to the importance of security in the SDLC. The SDSM is especially useful for organizations that have SDLCs that lack security considerations. Whereas the overall SDLC addresses all aspects and stages of the system, the SDSM focuses primarily on the system's security needs and is limited to the Requirements, Analyze, Design, Build and Test, and Deploy stages.

The SDSM's primary audience is the project team that will be developing a new system in-house, or evaluating a third-party system for purchase. The project team should incorporate the concepts from each phase of the SDSM into the corresponding phases of the organization's existing SDLC to ensure that security is appropriately considered and built into the system from the beginning stages. Inclusion of security in this way will result in a robust end system that is more secure, easier to maintain, and less costly to own.

System Development Security Framework

Exhibit 103.1 provides a framework for the System Development Security Methodology. Each step is described in detail later in this chapter.

System Development Security Methodology

The following sections describe in detail what the System Development Security Framework (Exhibit 103.1) depicts visually. Sections are numbered as in Exhibit 103.1.

Stage 1: Requirements

The high-level objectives of the requirements stage are to:

- Extrapolate information security requirements from business requirements

EXHIBIT 103.1 System Development Security Framework

Software Development Lifecycle	Stage				
	1. Requirements	**2. Analyze**	**3. Design**	**4. Build and Test**	**5. Deploy**
System development security	1.1 Identify information protection requirements	2.1 Identify risks and costs	3.1 Design system security components	4.1 Build secure environments	5.1 Secure code migration
	1.2 Identify corporatewide and regulatory security requirements	2.2 Conduct risk vs. cost analysis	3.2 Determine and establish development security needs	4.2 Enforce secure coding practices; build security components	5.2 Sanitize obsolete environments; secure production environment
	1.3 Identify user base and high-level access requirements	2.3 Determine security scope and finalize security requirements	3.3 Security procurement	4.3 Conduct code review	5.3 Secure deployment process
	1.4 Identify security audit requirements	2.4 Evaluate resource needs (time, budget, people)	3.4 Develop security testing approach	4.4 Conduct security testing	5.4 User awareness and training
Security deliverable/ endproduct	1.5 Detailed security requirements	2.5 Security project plan 2.6 Initial risk mitigation document	3.5 Security design 3.6 Security test plan	4.5 The prepilot environment	5.5 Completed risk mitigation document
Information security certification	Initial certification review		Certification checkpoint	Vulnerability assessment	Certification issuance

- Capture applicable security policies, standards, and guidelines from within the organization
- Capture applicable regulatory and audit requirements, such as GLBA, HIPAA, Common Criteria, etc.
- Create a detailed security requirements deliverable.

Step 1.1: Identify Information Protection Requirements

The typical SDLC tends to focus on business capabilities in the Requirements stage. The SDSM seeks to anchor the project team on the confidentiality, availability, and integrity of information early in the development process. Different industries and systems have dissimilar information protection requirements. For example, healthcare organizations might stress the confidentiality of patient records, whereas banking might be more concerned about the integrity of monetary transactions.

The project team needs to understand and capture what adequate protection of information means in their specific context. Organizations with an information or data classification policy(ies) are at an advantage here because the team could more conveniently identify the type of information that is processed as well as the organization's requirements as to how the information is to be protected. Once the types of information are identified, protection requirements should be organized further into areas such as storage and exchange, authentication, and access control. Requirements should be based, not only on the classification of the data (e.g., internal use, highly confidential), but also on the way in which data is accessed (e.g., via the Internet, remotely via leased lines, or from inside the organization), and the type of user (e.g., educated employees, public users, etc.), as well as the way in which access is managed (e.g., rule-based, role-based).

Step 1.2: Identify Organization and Regulatory Security Requirements

Of key importance is that the project team verifies and captures all applicable information security policies and standards pertaining to the system to be developed to ensure that the organization's security requirements are being met. Equally important is for the project team to be aware of current as well as pending federal, state, and local regulatory standards. Project teams should be aware that different states have begun

implementing bills specific to information security. For example, the California Senate Bill 1386, which became effective on July 1, 2003, requires a business to notify individuals if their personal information may have been compromised because of a security breach. Finally, the organization should document any requirements from the organization's audit and compliance group.

Step 1.3: Identify User Base and Access Control Requirements

The largest impact to a system's security is caused by users. It is important to know the user communities that will require access to the system, and how the system will identify, authenticate, and authorize the users in each community. As part of the access control mechanism, the project team should also consider the service requirement. If the team is evaluating or developing a system of critical importance that may be subject to service attacks, it is important that access be controlled to ensure that the most important users have priority when they need it. In most organizations, loss of service is an annoyance or results in loss of revenue. In the military, loss of service could result in loss of life.

Step 1.4: Identify Security Audit Requirements

Depending on the sensitivity or criticality of the information stored on the system, the organization may need to hold individual users highly accountable for their actions on the system. The SDLC tends to focus on error reporting and system events. It is not uncommon for systems to be built with little or no consideration for security auditing requirements. This neglect affects the accuracy and granularity of security-related event tracking, which in turn makes auditing and incident handling activities more complex. The project team should consider the following when identifying security audit requirements:

- Determine the alignment with organizationwide security auditing strategy
- Determine the audit approach: subject-oriented (uses, roles, groups) vs. object-oriented (files, transactions) vs. a hybrid approach
- Determine the level of granularity needed to provide a sufficient audit trail
- Determine the administration and protection of the audit logs
- Determine the life cycle of the audit logs (align with the organization's retention policies)
- Determine the interoperability of the auditing capability (operability with other repositories)

Step 1.5: Detailed Security Requirements Deliverable

The detailed security requirements deliverable should be a subset of the requirements document(s) produced in the SDLC process. Exhibit 103.2 provides a sample of sub-headings that should be included in this deliverable.

The detailed security requirements deliverable is a living document that may need updating in later stages. This document will be used in the Design stage to create a one-to-one mapping of functionality to requirements to ensure that all requirements have been addressed.

Stage 2: Analyze

The objective of the Analyze stage in the SDSM is to provide a dose of reality in the ideal world of the Requirements stage. The project team must determine the viability of designing and implementing the security requirements and adjust appropriately according to budget, resource, and timeline constraints. Subsequently, the final scope should be defined; the project deliverables, timelines, checkpoints, budget, and resources should be identified; and a security project plan should be created for incorporation into the overall SDLC project plan. A high-level information security risk document should also be prepared for presentation at the initial certification review (discussed later in the chapter).

It is critical that a thorough security analysis is done to ensure that the proper security elements are considered in the Design stage. An incomplete analysis could lead to a faulty design, which at best will lead to costly rework, and at worst will result in an insecure end product.

Step 2.1: Identify Risks and Costs

The project team should understand how the addition of a new system will impact the organization's existing IT architecture, and what new security risks the system could introduce into the environment. This exercise should identify the appropriate network location of the new system, and the security touchpoints between the system and the preexisting IT infrastructure.

Once the new system has been "placed" into the environment, the project team should identify all possible security threats to the system, including technical hazards (e.g., power outages, security vulnerabilities), man-made hazards (e.g., fire, sabotage), and natural hazards (e.g., floods, tornadoes). The team should then identify

EXHIBIT 103.2 Sample of Content that Should Be Included in the Detailed Security Requirements Deliverable

Subheadings	Content	Example
Information storage and exchange	Information classification Encryption requirements (if applicable) Information exchange control points (entry/exit)	Customer insurance policy information is classified as Confidential, and must be encrypted when transmitted over the Internet Customer insurance policy being transmitted to business partner must pass through a single entry/exit point
Identification/ authentication	User communities specification (external end users, internal end users, business partners, support, administrators, vendors, etc.) Authentication strength (password, strong passwords, two-factor, biometrics) Warning banner requirements Credential management requirements	Public end users must be uniquely identified and authenticated to the system using strong passwords
Authorization	Mode of access control (role-based, rule-based) Levels of access rights Access move, add, delete requirements	Role-based authorization must be used Users can have multiple roles
Reliability of service	High availability and redundancy requirements Fail-safe requirements Error and security notification requirements	Failure of the log-on mechanism must exit safely and not grant access to the requestor
Accountability	Security-related activities to be logged	Log-on failures must be timestamped and the user ID and number of attempts logged
Audit	Audit reporting functionality	Report of failed log-ons over the past 30 days

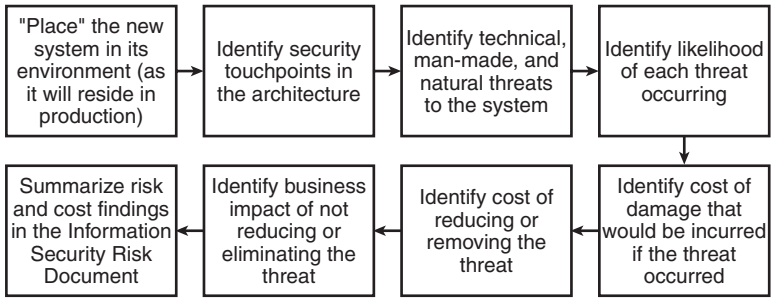


EXHIBIT 103.3 High-level flow depicting the process of identifying risks and costs of a new system..

the likelihood that each threat will occur, and estimate the cost of the potential damage. Next, the project team should estimate the cost to mitigate the risk, and determine the business impact if a risk is not addressed. Finally, the project team should highlight the most costly and complex security requirements, and document the risk and cost findings at a high level in the information security risk document. Exhibit 103.3 summarizes the process of identifying risks and costs.

Step 2.2: Risk vs. Cost Analysis

It is possible that the costs of implementing security outweigh the risks, in which case the requirements should be modified or an exception to the security requirement obtained. For example, a project team in the healthcare industry is building a capability that requires external e-mail exchange of personal health information (PHI). Encryption of PHI transmitted over public e-mail is a regulatory requirement. If the cost of deploying a secure interorganizational e-mail solution is beyond the budget of the project, an alternative may be to use "snail mail" or secure faxes. Another option is to propose a shared infrastructure for an enterprisewide secure e-mail solution and obtain an exception until this capability is built out.

Step 2.3: Determine Security Scope and Finalize Security Requirements

Once risks, costs, and impact have been analyzed, the project team should determine the system requirements to include or exclude based on cost, risk, complexity, timing, impact, etc. This determination should take into consideration the impact of security on end users, the potential damage that the end user could do to the system, other threats to the system (i.e., natural, technical, or man-made hazards), and business needs. The risk analysis should be consolidated, and the project team should formulate risk mitigation activities and prepare exception requests (discussed later).

The project team should also make a determination around building, buying, reusing, or outsourcing security components. In this decision, the cost of security vs. the value it adds should be considered, as well as the complexity and robustness of the solution options. Lastly, the requirements should be finalized.

Step 2.4: Evaluate Resource Needs

Once the final requirements have been established, the project team can identify timelines and checkpoints to build or configure the required functionality. The project team should also identify the project budget, and resources that will be conducting the design, build, test, and implement work, along with their roles and responsibilities. Resources performing security tasks should have a security background or should be supervised by someone who does. This may necessitate budgeting for internal or external security subject matter experts (SMEs) if security expertise is not available on the project team. Finally, the project team should plan time, effort, and resources for the certification process (discussed later).

Step 2.5: Security Project Plan

The security project plan deliverable should be a subset of the overall project plan produced in the SDLC process. The security project plan should include the subheadings listed in Exhibit 103.4.

Step 2.6: Initial Risk Mitigation Document

The risk mitigation document is a living document that is created in the Analyze stage and updated throughout the SDLC process to track information security risk. This document is completed at the end of the certification process in the Deployment stage. The risk mitigation document should identify assets that are affected by the new system; the threats to and vulnerabilities within those assets, including likelihood of occurrence; the business impact if a vulnerability is exploited; a prioritization of the risks in accordance with the likelihood of occurrence and impact to the business; and a mitigation plan for each risk.

Stage 3: Design

The high-level objectives of the SDSM Design stage are to:

- Formulate how security components are to be built and incorporated into the overall system design
- Define the environments for secure development
- Conduct vendor or capability selection
- Prototype designs and finalize procurement decisions
- Formulate security testing plans (component, integration, product)
- Pass the certification checkpoint (discussed later).

EXHIBIT 103.4 Subheadings that Should Appear in the Security Project Plan Deliverable, and Their Suggested Content

Subheadings	Content
Timelines and checkpoints	Convert security requirements into tasks and assign duration and FTE to tasks
	Identify tasks for security certification
	Establish checkpoints to monitor progress
Budget	Identify FTE cost
	Identify material cost (software, hardware, support, services)
	Identify project management cost
	Identify miscellaneous cost
Roles and responsibilities	Define organizational structure
	Define roles to complete security tasks
	Define responsibilities for each role

Step 3.1: Design System Security Components

At this point, the project team should define the design of security components that will meet the documented security requirements. These components include security functions within the system, such as access role definitions, or separate yet complementary security components, such as a single sign-on architecture. The objective here is to flesh out the various security components of the system to meet stated requirements. Success criteria should also be defined for each security component (to be used in security testing). Here are some security design principles to keep in mind:

- *Avoid security for security's sake:* Focus on the overall capability and the associated risk factors.
- *Address the key security areas:* Identification, authentication, authorization, confidentiality, integrity, availability, accountability, and where applicable, non-repudiation.
- *Forge multiple layers of controls:* Be wary of single-points-of-failure and the location of the weakest link.
- *Strive for transparent security:* It is an end user's best friend.
- *Keep security simple:* Complex designs have many secrets.
- *Consider the life cycle of the security component:* Start with secure defaults and end with fail-safe stance.
- *Favor mature and proven security technologies:* New is not always best, and organic is not always healthiest.
- *It is ready when you can take it to an expert:* Engage information security subject matter experts to review the soundness of the design.

Perform Prototype Testing to Validate the Capability. Prototype testing validates that the combined elements of a proposed design meet the security requirements. This should occur before the detailed design is complete. The prototype testing is also considered a precursor to the application testing. This may occur in a prototype or test-bed environment. Designers should choose the basic components that will constitute the system based on the assumption that the components possess the capabilities called for in the requirements.

Before time and effort is devoted to a detailed design, these assumptions must be verified and the risks must be evaluated. How this analysis is done (empirically, by developing a prototype of the proposed system, or less formally) will depend on the familiarity of the design team with the proposed architecture. In short, a gray area exists where the differences between verification and actual testing are ill defined. The project team should seek a level of rigor appropriate for the complexity of the system.

Step 3.2: Determine and Establish Development Security Needs

It is critical that the project team has an appropriate environment (or environments) in which to conduct the Build and Test stage. This environment should be documented as part of the Design stage. The project team should make arrangements to acquire development, testing, staging, and production environments that meet their needs. These environments should be physically or logically separate and properly secured. The project team should also define mechanisms to maintain the integrity, confidentiality, and availability of the source code by version control, checksums, access rights, logging, etc.

Access privileges should be defined according to roles and responsibilities. Access to source code, system utilities, developer privileges, and developer manuals should be restricted. Media should be protected and software properly licensed.

To ensure secure and smooth migration from one environment to the next, the project team should define change control and risk mitigation processes, including a secure code migration strategy.

Step 3.3: Security Procurement

To reduce costs and ensure interoperability with other systems in the organization, the project team should identify and procure any reusable security components, such as token or smart card technologies. If a third-party system is to be purchased, the project team should undergo a vendor selection process in which preexisting vendor relationships, industry recognition, company stability, support offering, product features, etc., are considered.

Once candidate components are procured, the project team should prototype potential solutions to verify capability, performance, interoperability, etc. When a vendor is selected, the project team should work with applicable legal or procurement representatives to establish contracts and agreements (Service Level Agreements, Operational Level Agreements, Nondisclosure Agreements, etc.).

Step 3.4: Develop Security Testing Approach

Security testing in the SDSM differs from functional testing in the SDLC. Security testing focuses, not only on those functions that invoke security mechanisms, but also on the least-used aspects of the mechanisms,

primarily because the least-used functions often contain flaws that can be exploited. As such, security testing usually includes a high number of negative tests whose expected outcomes demonstrate unsuccessful attempts to circumvent system security. By contrast, functional testing focuses on those functions that are most commonly used.

Develop a List of Assertions. A reasonable approach to testing is to begin by developing a list of assertions. Security test assertions are created by identifying the security-relevant interfaces of a component, reviewing the security requirements and design documentation, and identifying conditions that are security relevant and testable. A few examples of security-relevant interfaces include the password-changing module available to a user, the user administration module available to a security administrator, the application programming interface (API) available to an application programmer, and the console interface available to a network administrator.

Examine such interfaces and the documentation associated with them for testable assertions. For example, the statement "A user should be able to change his own password" is an assertion that might be found in design documentation; a test can be built around this assertion.

Distinguish between Different Types of Tests. Security test procedures will be needed for several types of tests:

- Prototype testing to validate the security capability
- Component testing to validate package, reuse, and custom security component tests
- Integration testing to validate security functionality in integration testing and product testing
- Volume testing to ensure that the system will process data across physical and logical boundaries
- Stress testing to ensure effective transaction processing immediately after system downtime, after network downtime, or during peak periods (denial-of-service conditions)
- Data recovery testing to investigate both data recovery capabilities and system restart capabilities for fail-over and redundancy
- Database security testing to ensure that access is not provided outside the system environment

Step 3.5: Security Design Deliverable

The security design deliverable should be a subset of the overall system design deliverable produced in the SDLC process. The format and subheadings of the security design deliverable should follow that of the overall system design deliverable.

Exhibit 103.5 provides a recommended listing of security subheadings for this document.

Step 3.6: Security Test Plan

The security test plan should be a subset of the overall test plan deliverable produced in the SDLC process. The format and subheadings of the security test plan should follow that of the overall test plan deliverable, as summarized in Exhibit 103.6.

Stage 4: Build and Test

The high-level objectives of the Build and Test stage are to:

- Build secure environments to foster system development integrity and protect preexisting infrastructure
- Promote secure coding practices to ensure the security quality of the finished product
- Enforce formal code review procedures to inculcate checks and balances into the code-development process
- Thoroughly test all security components to validate the design; build a pilot capability
- Resolve issues within the certification process and pass the vulnerability assessment (discussed later).

Step 4.1: Build Secure Environments

Due to the laxness that typically exists in nonproduction environments, preexisting and future production environments should be appropriately demarcated from development, testing, and training segments. The project team should also configure (or arrange for the configuration with the network support team) network control points (such as firewalls, routers, etc.) to meet development, administrative, and operational objectives. Furthermore, the development environment should mirror the production environment as closely as possible for system build because the system will ultimately have to function properly in the more rigorously controlled production environment.

EXHIBIT 103.5 Recommended Subheadings for the Security Design Deliverable, and Their Suggested Content

Subheadings	Content
Introduction	Purpose
	Context
	Scope
	References
Security requirements to design mapping	List security requirements
	List matching security components to meet each requirement
High-level description	Describe each security component design at a high level
	Describe interaction among security components, system architecture, and network infrastructure
	Describe information flow
	Describe environments
	Include diagrams and flow charts
Detailed design	Describe each security component in detail
	Describe software, hardware, service specifications
Environment design	Describe details of development, testing, staging, and production environments
	Describe code maintenance process
	Describe secure code migration strategy
	Describe media protection and licensing protocols
	Describe change control and risk mitigation processes
	Describe physical security of development servers and workstations

EXHIBIT 103.6 Recommended Subheadings for the Security Test Plan Deliverable, and Their Suggested Content

Subheadings	Content
Introduction	Purpose
	Context
	Scope
	References
Security design to test mapping	List security design
	List matching testing components to validate each design
High-level description	Describe test approach or process and documentation procedures (should be similar to SDLC)
	Describe each testing stage: component, integration, product
	Characterize test environments
	Specify entry/exit criteria
	Describe dependencies
Detailed design	Develop list of assertions
	Specify test input requirements
	Describe test cases
	Define each testing phase; provide entry/exit criteria for each phase
	Describe test procedures; specify "testware" to use
	Describe regression test approach and criteria
	Describe code fix criteria
	Describe testing deliverables

A key activity in the SDSM's Build stage is server hardening. Hardening is the process of removing or disabling unneeded services, reconfiguring insecure default settings, and updating systems to secure patch levels. A common fallacy in the SDLC process is that systems are developed on unhardened servers and server hardening takes place in the production build-out phase. This predicament makes deploying applications on hardened servers a crapshoot, often resulting in system anomalies, finger-pointing, delayed timelines, and worst of all, a permissive hardening stance to accommodate the application. A better approach is to ensure that development is done on hardened servers and that documentation of necessary services, protocols, system settings, and OS dependencies is captured through the development process.

Finally, to ensure availability, the project team should build or make arrangements for appropriate backup and availability capabilities.

Step 4.2: Enforce Secure Coding Practices and Build Security Components

Software developers must be educated in secure coding practices to ensure that the end product has the required security functionality. This is a challenge in most organizations because, historically, security techniques have not been taught in programming classes. Where possible, the organization should arrange for formal secure coding training for its developers.

The following paragraphs describe some high-impact recommendations for improving information security within an organization's application(s).

Encryption and Random Number Generators. The developer should use well-established cryptographic algorithms as opposed to implementing proprietary or obscure cryptographic algorithms. An example of published encryption standards and mechanisms recognized by the cryptographic community are those listed in the Federal Information Processing Standards (FIPS) publication.

Another fallacy related to cryptographic functions is the use of pseudorandom number generators (PSNG). Developers should evaluate their PSNG against the criteria set by RSA:*

- Random enough to hide patterns and correlations (i.e., distribution of 1s and 0s will have no noticeable pattern)
- Have a large period (i.e., it will repeat itself only after a large number of bits)
- Generate on average as many 1s as 0s
- Not produce preferred strings such as "01010101"
- Is a simple algorithm with good performance
- Knowledge of some outputs will not help predict past or future outputs
- The internal state of the PRNG will be sufficiently large and unpredictable to avoid exhaustive searches

Input Validation and Exception Checking. Always validate (user and application) input. Most of the exploits seen in recent years were a direct result of poor or incorrect input validation and mishandled exceptions. Independent of the platform, applications have been regularly broken by using attacks such as buffer overflows, format string vulnerabilities, utilization of shell escape codes, etc. Never trust input when designing an application and always perform proper exception checking in the code.

Authentication. Authentication strength is paramount to the security of the application or system, because other security controls, such as authorization, encryption, and auditing, are predicated on the authenticity of the user's identity. However, authentication strength must always be weighed against usability. Enforcing a 10-character password will only lead users to write passwords on Post-It notes and stick them next to the terminal.

Do not hardcode credentials into applications and do not store them in clear-text. Hardcoded passwords are difficult to change and sometimes even result in a clearly visible password in compiled application executables. A simple "string application_name" command on a UNIX host can reveal a password that is not encrypted. A good practice is always to encrypt authentication credentials. This is especially important in a Web application that uses cookies to store session and authentication information.

Favor centralized authentication where possible. Centralized authentication repositories allow for a standardized authentication policy across the enterprise, consistency in authentication data, and a single point of administration.

Authorization. The authorization control is only as strong as its link to the identity it is authorizing (this link is the main target of impersonation attacks). In building out the authorization model, it is critical to form a strong link to the identity through the life cycle of the authenticated session. This is of particular importance in Web applications or multi-layered systems where the identity is often propagated to other contexts.

Logging and Auditing. Logging and auditing can provide evidence of illegal or unauthorized access to an application and its data. It can become legal material if law enforcement authorities get involved. For this reason, logging and auditing should be designed to offer configurable logging and auditing capabilities, which allow the capturing of detailed information if necessary.

Code Dependencies. Code development, especially object-oriented programming, often depends on the use of third-party libraries. Only acquire and use libraries from established vendors to minimize the risk of

*http://www.rsasecurity.com/solutions/developers/whitepapers/Article4-PRNG.pdf

unknown vulnerabilities. Also, validate return code or values from libraries where possible. Similar precautions should be taken when relying on external subsystems for processing and input.

Error Messages and Code Comments. Error messages should not divulge system information. Attackers usually gather information before they try to break into an application or a network. For this reason, information given out to a user always should be evaluated under the aspect of what a user needs to know. For example, an error message telling the user that a database table is not available already contains too much information. Exception handling should log such an error and provide the user with a standard message, saying that the database is not available.

In the same vein, do not include comments in public viewable code that could reveal valuable information about the inner workings of the system. This is strictly targeted at Web applications where code (and its associated comments) resides on the browser.

Online Coding Resources. The following Web pages provide detailed practical assistance for programmers:

- C/C++: http://www.cultdeadcow.com/cDc_files/cDc-351/; http://www.securityfocus.com/data/library/P49-14.txt
- Perl: http://www.perl.com/CPAN-local/doc/manual/html/pod/perlsec.html
- Java: http://java.sun.com/products/jaas; http://java.sun.com/security/seccodeguide.html; http://dwheeler.com/javasec/
- UNIX: http://dwheeler.com/secure-programs; http://www.sans.org/
- ASP: http://msdn.microsoft.com/library/default.asp?url = /library/en-us/iisref/html/psdk/asp/aspguide.asp

Step 4.3: Conduct Code Review

Code review from the SDSM perspective has the objectives of checking for good security coding practices as well as auditing for possible backdoors in the code. It is a well-known fact that insiders conduct the majority of security exploits. Code developers are no exception to that rule.

Step 4.4: Conduct Security Testing

Security testing provides assurance that security was implemented to meet the security requirements and to mitigate the risks identified in the security design plan. Security testing ascertains that the proposed components actually perform as expected and that security requirements are met throughout the integrated solution.

The key aim of security testing is to search for exposures that might result in unauthorized access to the underlying operating system, application resources, audit or authentication data, network resources, or that could lead to denial-of-service attacks. Security testing also aims to identify and address the risk of noncompliant components. The risk and proposed mitigation plans should be captured in the project's risk mitigation document (which was created in the Analyze stage).

There are as many different breakdowns for testing phases as there are SDLCs. In the interest of simplicity, the SDSM has three broad test phases: component testing, integration testing, and product testing, as described in the following paragraphs.

Perform Component Testing. Many components combine to form a security infrastructure. In general this includes firewalls, authentication servers, encryption products, certificate servers, access control mechanisms, and routers. Configuration management is often the weak link that creates new exposures. Perform testing for these components individually to test the functionality and to identify any weaknesses in the configuration. The component testing should cover security functionality, performance, failure-proof or fail-safe ability (in case the individual component is compromised), logging and monitoring capability, and manageability.

Security testing should include stress testing. Stress testing and worst-case-scenario testing will help in exposing how well the component behaves under overloaded conditions. These types of testing will also indicate the capability's exposure to denial-of-service attacks.

Perform Integration Testing. The next phase of the testing should focus on integration testing. This phase focuses on how well each component integrates with the other components in the architecture. The objective is to ensure that security requirements are met throughout the environment. Migrations to new environments and integration of custom and packaged components should be thoroughly tested.

Perform Product Testing. Product test execution will occur only after all package, custom, and reuse components have completed integration testing. The product test execution may not end until the entire product test model has been executed completely and without discrepancies.

All pieces of the security solution are to be installed and configured in a test environment to mimic a production environment as closely as possible. For the best results, product testing should occur in a

production-readiness (staging) environment. This environment should include all packaged software and all hardware chosen for production.

When a new capability is introduced into an existing networked environment, the new capability inherits all the risks associated with that environment. Therefore it is extremely important to test how well the capability meets its security requirements within the production environment.

General Tips on Security Testing. The following list provides some general tips on testing for security:

- Discourage the use of production data in the test environment
- Do not use production passwords in the test environment
- Use strong passwords (minimum seven characters, alphanumeric, with mixed case and special characters) in the development environment to emulate production
- Educate the testing team on specific security concerns, such as buffer overruns in C, TCP/IP vulnerabilities, operating system bugs, and ActiveX, Java, and CGI code problems
- Purge test data appropriately so that residual data is not available in the operating environment after it is used
- Disable test accounts when they are no longer necessary
- Document, evaluate, and address security risks of a noncompliant component at each testing phase

Step 4.5: The Prepilot Environment

The prepilot environment should have full system functionality and have gone through and passed all testing stages. This environment should be part of the SDLC process. The additional security requirement here is getting the environment through the security certification process. This involves coordinating with the certification team to conduct a vulnerability assessment on the prepilot environment.

Stage 5: Deploy

The high-level objectives of the Deploy stage are to migrate systems safely from development through to production; systematically cleanse obsolete environments of security-sensitive information; ensure and preserve the confidentiality, integrity, and availability of the production environment(s); implement secure deployment of systems, user information and credentials, post-configuration information, etc.; employ secure code enhancement, software updates, and bug-fixes procedures; secure deliverables produced during the SDLC; and complete the risk mitigation document and obtain certification sign-off.

Step 5.1: Secure System Migration

A secure system migration process contributes to the goal of keeping the production environment as pristine as possible. To ensure that security is maintained throughout the migration process, the project team should assign migration owners and appropriate approval processes to ensure accountability and control during migration. Furthermore, least privilege should be used when granting access to personnel involved in the migration process.

The migration should be conducted using secure protocols and mechanisms across environments. Once the system has been migrated, integrity verifiers (e.g., checksums, message digests) should be used to verify the system's integrity. The project team should also identify and enforce security maintenance as part of regularly scheduled maintenance windows to ensure the continued integrity of the new system in production. Security regression testing should be incorporated in the maintenance cycle to validate the integrity of the system after scheduled changes.

Step 5.2: Sanitize Obsolete Environments and Secure Production Environment(s)

The project team should implement a process to identify and sanitize development, test, and staging computing resources or environments that are no longer needed. Passwords (root, system, administrative, default, etc.) used in predeployment activities should be changed in all environments, especially production. The project team should also conduct a formalized transition of relevant credentials, system information, processes, documentation, licenses, etc., to the permanent operations or production team.

During the SDLC process, a number of deliverables were produced that contain sensitive information, such as architecture specifics and risk analyses. Such deliverables must be kept for auditing and historical purposes, but they must be controlled to avoid improper disclosure of the information they contain.

Finally, the project team should ensure that the new system has adequate physical security when placed in production.

Step 5.3: Secure Deployment

In the rush of making production deadlines, it is not uncommon for user password lists and other sensitive material to be mass distributed. These types of information could be used at a later time to gain unauthorized access into the system. The SDSM seeks to raise awareness of this issue. During deployment, the collection, setup, and distribution of credentials (passwords, tokens, etc.), and post-configuration information (gateway, required ports, environment variables, etc.) should be appropriately controlled, monitored, and accounted for. When granting access to personnel involved in deployment activities, and to permanent system users, least privilege should be used. All user access should be documented.

Step 5.4: User Awareness and Training

It is difficult to maintain the security of a system without properly educating the users of that system. It is important that the project team raise user awareness on how to create good passwords, protect credentials, and promote understanding of other security-specific features, such as timeout mechanisms, account lockout, etc.

The project team should identify user support activities and set up caller authentication procedures to verify the identities of users calling the help desk for assistance, and users should be made aware of help desk authentication practices to avoid social engineering attacks.

Step 5.5: Completed Risk Mitigation Document

The risk mitigation document is a living document that was created in the Analyze stage and updated throughout the SDLC process to track information security risk. The project team should confirm that all open risk items have been adequately mitigated or have appropriate exception approvals. The completed risk mitigation document should be signed-off as part of the certification issuance process.

Certification Framework

Throughout this chapter the concept of certification has been alluded to. A certification framework is critical to ensuring the sustenance and improvement of the organization's information security baseline. The objectives of certification are to:

- Ensure correct interpretation of security policies and standards
- Assess and manage risk throughout the capability development life cycle
- Formalize the confirmation of compliance to security policies and standards
- Formalize the acknowledgment and acceptance of information security risks
- Facilitate resolutions, suggest alternatives, and authorize waivers to achieve compliance
- Authorize and track waivers and postponements

It is highly recommended that the organization develop an internal certification process in conjunction with the internal audit and compliance group. An internal certification process can be implemented instead of or in preparation for a formal, external certification such as SAS 70 or ISO 17799, or for a government certification and accreditation. The following paragraphs describe the certification components that have been referenced throughout this chapter.

Initial Certification Review

The initial certification review takes place after the Requirements and Analyze stages and before the Design stage. The objectives of this review can be seen from two sides — the certification team and the project team. For the certification team, this review is an introduction to the project and allows the team to get acquainted with the project's key players as well as the overall capability that is being proposed. For the project team, the objectives of the review are to familiarize them with the certification process, raise exceptions issues, and glean security subject matter expertise from the certification team. The benefits of the initial certification review are early identification of noncompliant issues, facilitation of exceptions requests, and knowledge sharing.

In the initial certification review, the certification team will conduct requirements review and interview sessions with relevant individuals, collect and document the project's alignment with security policies and standards, and provide project teams with resources (e.g., templates, information from similar projects) to facilitate the certification process. The certification team will also review any exception requests that have already been documented, and facilitate the approval or denial of those requests. It should be noted that although the certification team is comprised of security professionals, the individual that certifies the system or approves an exception is a functional owner, who is in a position to accept the risk for the organization.

Prior to entering the initial certification review, the project team must have obtained and reviewed all pertinent information security policies and standards, business requirements, and external regulatory requirements, and produced a detailed security requirements document, a security project plan, an initial risk mitigation document, and any initial exception requests.

Upon completion of the initial certification review, the project team will be provided with approvals or denials of all initial exception requests, and they will have all the information necessary to create the risk analysis document for the Requirements and Analyze stages, which capture risk issues, policies, standards, and regulations that are violated, business impact, likelihood of risk, the discovery timeframe, and the cost to fix. The document also contains a listing of risks that are ranked, an outline of mitigations, and timeframes for compliance.

Certification Checkpoint

The certification checkpoint takes place after the Design stage and before the Build and Test stage. The purpose of this checkpoint is to keep the channels of communication and feedback open between the certification team and the project during the Design stage.

At this time the certification team validates the project team's security design against stated security requirements. The certification team also reviews the security designs to identify noncompliant issues and potential security implications with the enterprisewide security posture. Handling exceptions should also be a common activity during the certification checkpoint. Finally, the certification team should also provide cross-enterprise resources to the project team. For example, the certification team would know of previously certified projects that have a secure file transfer design similar to the needs of the current project.

Prior to entering the certification checkpoint, the project team must have a completed security design document. After the checkpoint, the project team will receive approvals and denials on any new exception requests, based upon which they will need to update the risk analysis document.

Vulnerability Assessment

The goal of the certification team during the vulnerability assessment is to test and identify noncompliant areas prior to deployment. In so doing, the certification team should exercise best effort to minimize disruption to project productivity. As a result of the vulnerability assessment, the certification team will provide empirical data to the project team, so they can update the risk mitigation document. The certification team also facilitates discussions with project teams to establish detailed activities for certification issuance at this point.

The certification team's activities during a vulnerability assessment are to:

- Understand and analyze the environment by conducting interview sessions with relevant parties
- Obtain and review environment documentation
- Assess threat factors and identify application, system, infrastructure, and process vulner-abilities
- Perform a vulnerability assessment with automated scanning tools and selected manual exploits
- Present security analysis findings to the project team
- Discuss security implications and project mitigation activities
- Establish and gain consensus for the completion of the risk mitigation document
- Establish a timeline and checkpoints for certification issuance

Prior to entering the vulnerability assessment, the project team must have an updated risk mitigation document, as well as completed build and test deliverables.

Once the vulnerability assessment has been completed, the certification team provides the project team with a security assessment report, which contains the findings from the assessment. At this time, the project team can update the risk analysis document for the Build and Test stage, as well as the risk mitigation document.

Certification Issuance

The purpose of certification issuance is to formalize the confirmation of compliance to security policies and standards, as well as the acknowledgment and acceptance of information security risks.

Prior to certification issuance, the certification team must validate the completion of the risk mitigation document; ensure that all design, build, and test deliverables have been finalized; and that all exceptions have been approved or that risks for denied exceptions have been mitigated. At this time, the certification team makes a recommendation to the certification issuer about whether or not the system should be certified.

Upon completion of this phase, the project team has completed risk mitigation and risk analysis documents, and a certification issuance decision.

Summary

To those unfamiliar with the SDLC and SDSM processes, the information presented in this chapter may seem daunting and unrealistic. Implementing such a methodology is in fact mostly a cultural issue, because it requires that project and development teams be more disciplined. It can also extend the project timeline a bit longer than management would like. However, the additional time and due diligence exercised prior to implementation has proven time and again to pay dividends in the long run, by producing systems that are robust, secure, and that do not require costly redesign. Those organizations that have undergone the growing pains have found that it was well worth the effort.

For the implementation of an SDSM or the larger SDLC to be successful, full management support and attention are needed. Also, a complete methodology must be developed by each organization with much more detail than was provided here, in terms that are specific to the needs of the individual organization. Furthermore, such a methodology must be maintained over time to ensure relevance. The technology focus at the writing of this chapter includes things like application servers and CGI scripts, but by the time this text is published, the hot technology will be Web services. Although the base methodology of Requirements–Analyze–Design–Build and Test–Deploy and certification will stand the test of time, the technical details will change frequently, and project teams and developers must keep up.

104

A Security-Oriented Extension of the Object Model for the Development of an Information System

Sureerut Inmor, Vatcharaporn Esichaikul, and Dencho N. Batanov

The meaning of computer system security varies, depending on the assets that the security mechanism is designed to protect. The main objective of all security mechanisms is that a system's assets perform their tasks according to authorized user expectations, while maintaining the confidentiality, integrity, and availability of information (CIA). The security aspects can be categorized, based on related assets, into four types: hardware security, software security, network security, and information system security.

Hardware security relates to the security of computer-related equipment, and requires physical access control mechanisms. Software security relates to the security of application programs, the database management system, and the operating system. The security mechanism might require both physical access control and access control via an authentication process. Network security is required when the computer system performs its tasks through some network connection. The security mechanism should emphasize data communication, such as transmission protocol security and data encryption. The information system security relates to how to analyze and design the organizational information system in such a way that this valuable data is protected against improper disclosure or modification.

From a security perspective, system analysts and developers should be most concerned with information system security. By contrast, providing security for hardware, software, and a network requires specific technical knowledge, and several vendors offer efficient security mechanisms and tools. These mechanisms are called infrastructure security and are already available through middleware products such as WebSphere and WebLogic. WebSphere is infrastructure software for dynamic E-business, developed by IBM business partners.[1] WebLogic Server is application infrastructure software, which was developed by BEA Systems, Inc. The security framework in BEA's WebLogic Server 7.0 has enabled the developer to unify security infrastructure to secure interactions among objects in an application system.[2] The analysts and developers cannot do much about these security infrastructures because commercial software is tested prior to acquisition and accepted as is. For their part, system analysts and developers should concentrate on what directly affects their tasks — that is, information system security.

Because information system security is the most manageable security requirement and the most important for system analysts, this chapter deals with how to analyze and design a security-oriented information system. Use of the information system will vary, depending on the type of organization and the kind of information or valuable data that each organization maintains. Information system security requirements are unique to

each organization, varying with business needs and types, as well as kinds of users — who may differ in terms of trustworthiness.

It is no longer sufficient to provide information system security in the traditional way, that is, providing an access control mechanism at the user interface level after developing and implementing the application system. Now there are mechanisms that concentrate on the object, mainly supporting the control of all direct access to objects. Several studies show how to provide suitable security to the system in this manner.[3-5] None of them, however, concentrates on how to design an object model to support the security requirements.

The need for information security is common to all organizations. In addition, the National Academy of Sciences (United States) has noted that poor analysis and design methods of developing information systems are major factors causing security problems in computer-based information systems.[6] Therefore, we propose an Object-Oriented Security Model (OOSM) with a "security-oriented extension of the object model" that can be used and implemented in the system analysis and design phase of system development. To design an object model that satisfies most of the security requirements is very important because it is the foundation of the overall security of an information system. There should be a useful guideline for the system developer on how each method (part of the object model) can be designed, in order to fulfill the system's need for security. Integrating security into an information system should start as early as possible. Our security model suggests integrating minimum-security requirements when each method is created. That is, developers should carefully control each method that is designed in order to provide an application system with satisfactory security requirements.

Most often, however, security requirements are left to the security administrator. As a result, access control mechanisms are put into the system after system development is done, in order to control how the end user gains access to the system's user interface. The system designer has little or no guidance in designing the system to keep the participating objects secure.

Security-Oriented Analysis of the Domain's Object Model Elements

In an object-oriented information system, the typical object model has three parts: the object name, the attributes and structural properties, and the operation that is required to access and maintain the object attributes, as shown in Exhibit 104.1.

All the operations in an object class comprise the object interface because they are the only way that other objects can "collaborate" with this object. There are three forms of object interface, according to Fayad et al.:[7]

1. The *attribute interface* provides access to the attributes of an object. The attribute interface can be used in three different categories:
   a. To return the value of an attribute
   b. To initiate the value of an attribute
   c. To notify other related attributes when the value of one attribute changes
2. The *action interface* provides access to other objects. The action consists of a task, such as displaying an object's attributes or adding one object to another set of objects. These actions can be implemented as a *public member* function;
3. The *event interface* provides notification to other objects when one object changes its state.

EXHIBIT 104.1 A Typical Object Model

Object name (class)	
Structural properties (attributes)	
• XXXXXXX	• XXXXXXX
• XXXXXXX	• XXXXXXX
Operation required to access and maintain object	
• V1:XXXXXXX	• V2:XXXXXXX
• V3:XXXXXXX	• Vn:XXXXXXX

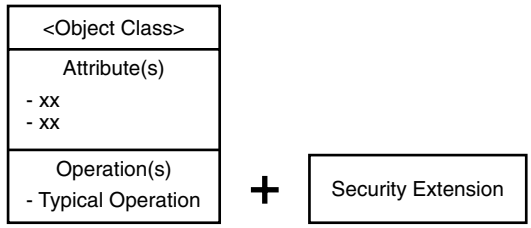


EXHIBIT 104.2 An object model with security extension.

The typical object model in Exhibit 104.1 is the starting point of an object model designed with security considerations. For each operation in an object model, we suggest that the analyst or developer consider security requirements of an application system and finally integrate them with the typical function. The information on security requirements comes from the security specifications through the software prototype technique.

From the security specifications, which were captured by the software prototype, the analyst or developer will understand the object relationship of an application system and also realize which operations have significant meaning from a security aspect. Those operations require a carefully designed object, which results in security extension to each typical operation, as shown in Exhibit 104.2.

The key difference between the traditional operation of the object model development and the proposed method is that our model adds an extra mechanism to each operation to ensure the security requirements. The model includes guidelines for designing each type of operation to satisfy the system's confidentiality, integrity, and availability needs.

The model extension, which performs on system domain objects, aims at helping the designer solve difficult information system design problems while satisfying security requirements. If each operation category is designed with security in mind from the beginning, the overall information system security should be significantly improved.

In each object model, operations are classified according to their purpose when interacting with an object instance. In general, there are four operation types:

1. *Query operation.* This type of operation displays the attribute value of the destination object instance.
2. *Update operation.* This operation makes some modifications to the attribute value of the selected instance.
3. *Terminate operation.* This operation terminates the object instance from any object class. After termination, the object is no longer an instance of any object type.
4. *Create operation.* This operation adds a new object instance into an existing object class.

Each operation category maintains extra information, listing conditions for invoking the operation to satisfy security requirements. Each operation also has an extra function: to perform secure operation invocation handling. The result of operation invocation depends on whether or not access control is currently accepted according to security requirements. Any possible operation invocation that will make the system vulnerable will not be allowed. The security function may be applied to any operation that is considered important for security, by specifying the pre- and post-conditions for every protected operation, as shown in Exhibit 104.3:

- A *pre-condition* is a set of security functions invoked prior to the invocation of the specified operation.
- A *post-condition* is a set of security functions performed after the operation finishes executing its task.

The pre- and post-conditions to normal operation will expend some execution time overhead, but this is the trade-off for security. It is the designer who decides how and in which operation security is applied.

We propose a classification of security functions for use with our model extensions. These security functions, which are in addition to the typical operation, can be categorized as follows:

- *Set membership test*. In some application systems, there exists a specific rule if the new object instance is to be added to an existing object class. This security function will check the condition to ascertain that the new object instance can be added properly.

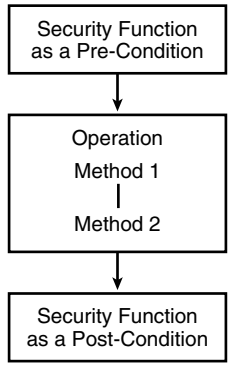


EXHIBIT 104.3 Normal operation integrated with security functions.

- *Terminate membership test*. The condition to remove an object instance from an existing object class must be checked. This function must guarantee that the object instance's termination will not cause any problem to other related domain objects.
- *Relationship cardinality test*. If the relationship cardinality between objects in a system is significant for security, one of the object attributes should maintain the value of cardinality. This kind of attribute can be used for checking the relationship cardinality.
- *State change permission test*. The state of the object depends on the different states that object of that class may have, as well as the event that will make it change its state. This security function can be implemented by defining conditions for restricting the possible state change in both pre- and post-conditions.
- *Correctness of input data test*. This function can be implemented as the pre-condition to limit the scope of the input parameters. The tasks of this function include:
  - — Character checks
  - — Range checks
  - — Relationship checks
  - — Reasonableness checks
  - — Transaction limits
- *Correctness of output data test*. This function is implemented in the form of post-condition to limit the range of the output parameters. The suggested tasks are as follows:
  - — Character checks
  - — Range checks
  - — Relationship checks
  - — Reasonableness checks
  - — Transaction limits
- *Notification*. According to the dependency relationship among objects, when one object changes its state, all its dependents should be notified and updated, to maintain consistency between related objects. For example, if a librarian decides to remove an out-of-date magazine from the library, what the system should do is:
  - — Remove magazine title in Title Object
  - Notify dependent objects, which are Item Object, Magazine Title Object, and Reservation Object, to update their instances.
- The notification function is also mentioned in an observer pattern in Gamma et al.[8]
- *Control condition for synchronizing series of operations.* To perform a specific operation, it is sometimes necessary to control the concurrent execution of transactions. In applying this kind of condition, we classify a series of operations into pre- and post-functions:

— A pre-function lists all the functions that must be executed before the system can invoke this operation.
— A post-function lists all the functions that the system should invoke in the next operation, after executing this operation.

- *Organizational policy.* The policy of each organization in the application system should be explicitly stated. For example, a university library system should have a policy on how to accept membership, the period for borrowing items, and the condition for specifying items as damaged or lost, among others. The user role, which can perform important operations, is also an organizational policy that must be specified in the operational design.
- *Audit trail.* This function will perform the following tasks:
  — Record all necessary information for future investigation.
  — Record the date, time, and user who invoked the protected operation.
  — The information comes from the authentication process.
- *Permission test for operation invocation.* This function is to assign a group of users/operations that have the right to invoke a protected operation.

Embedding security functions into a normal operation requires applying the pre- and post-conditions to an existing operation. To illustrate our concept, we employ a well-known object-oriented system analysis and design example from a university library system discussed by Eriksson and Penker.[9]

The use of software prototypes in the software development community is widespread, the main purpose being to present the user with the first version of software. A use case diagram can capture user functional requirements at the early phase of system analysis. But to ascertain that the analyst understands the user requirement correctly, the software prototype could be an essential tool for this task. The use of a software prototype from the OOSM viewpoint is to capture the users' and applications' security requirements. The design of a software prototype for this purpose should be a multilevel menu with a necessary access control mechanism. The feedback from users will help the analyst better understand the security requirements of an application system. The suggestions on how to develop a software prototype with the OOSM are as follows:

- *Prototyping language.* The designer should use the same programming language in both prototyping and the final software product. As in the OOSM, Visual C++ has been used both in prototype and software development;
- *Prototyping tools.* The application generator is a faster way to produce a prototype. If the language used does not have this tool, simple program construction could be used instead.

The software prototype in the OOSM is intended for the sole purpose of capturing security requirements. A description on how to translate this prototype into an object model can be found in Krief.[10] In the OOSM, the prototype is created using Visual C++, which uses the application generator plus additional programming language as necessary. The multilevel menu interface is applied with the interface of this prototype. The resulting software prototype, after discussion with the user, will provide the analyst with the security requirements of an application system. The analyst then considers which operation has significance in the security perspective and continues working with a carefully designed prototype of that operation.

Methodology for Applying the OOSM to Information System Development

The objective of the OOSM is to provide guidelines and procedures for an application system designer to use as a part of the analysis model. As a result, when the design is derived from the model, security will be well integrated into the application system. Before the application of the OOSM is explained, it is necessary to define the term "security" in this model. The model aims to meet three aspects of security:

1. *Confidentiality.* Information is not revealed to an unauthorized object in the system. This can be implemented by restricting access, that is, determining a set of objects that are allowed to request the execution of operation v from object x. In this model, a set of objects refers to an operation group name. An operation group and an operation differ as follows:

a. An operation group refers to the name of the task that the user needs to perform, such as borrowing a book or returning a rented tape.
b. An operation refers to each method in an object class that must be performed to accomplish one operation group or user task (e.g., checking for borrower identification, or retrieving information from the borrower record).

2. *Integrity.* The system's information maintains integrity with respect to overall information. Integrity is the ability of software systems to protect their various components (programs, data, and documents) against unauthorized access and modification.
3. *Availability.* The system provides necessary information to other objects on request, given the authorization to do so.

The OOSM is designed to be used side by side with the traditional Object-Oriented System Analysis and Design (OOSAD). The objective of OOSM is to be used with the information system development, which has a special need in security requirements. A use-case approach to system analysis and design as described in the UML standard is to be used as the main diagram to capture the system's functional requirements at the beginning. The additional diagrams are role diagram, operation structure diagram, sensitivity level diagram, and use-case diagram for security purpose.

Another tool in the analysis phase is the software prototype. The use-case diagram and the role diagram will be used together, mainly as resources to create the software prototype. An overview of OOSM with the traditional OOSAD is illustrated in Exhibit 104.4.

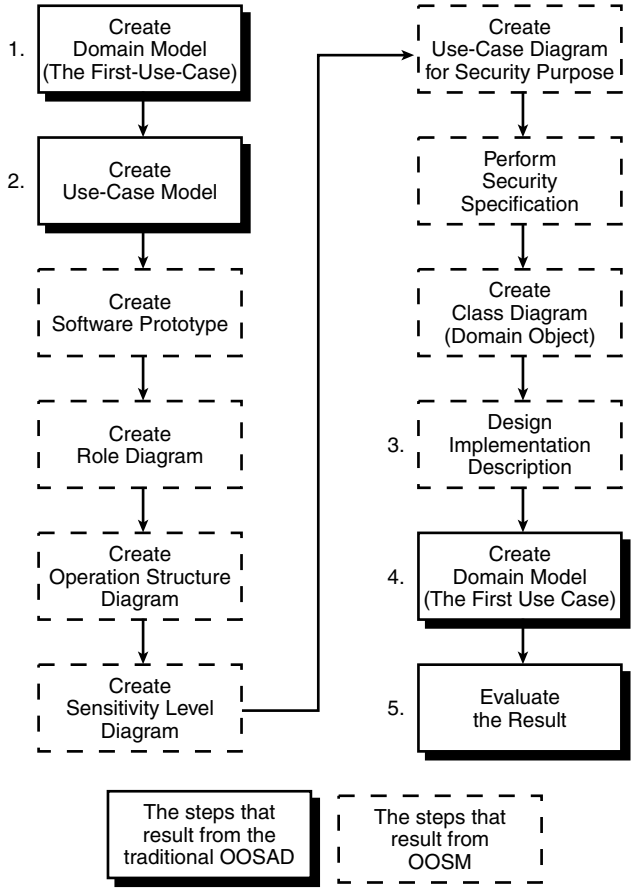


EXHIBIT 104.4 Overview of OOSM with the traditional OOSAD.

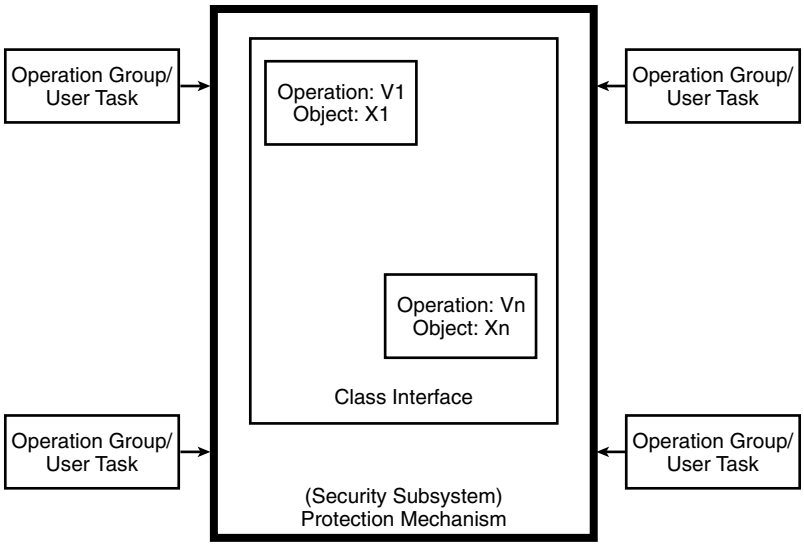


EXHIBIT 104.5 Protection mechanism for operation invocation using the model extension.

From the use-case diagram for security purposes, the analyst also provides new information for the security subsystem, which comprises all the operation groups or user tasks. Instead of directly invoking each operation group with the operation in the class interface, the model extension will provide a protection mechanism for operation invocation, as shown in Exhibit 104.5.

Every request for an operation must go through the process of access checking in the security subsystem. Therefore, the security subsystem must maintain all necessary information needed to accomplish the task of access checking. This model extension has a role in providing safeguards for the application system during the development process.

As shown in Exhibit 104.6, the system analyst or the system designer will use this model extension as a part of his or her design. After implementation, the model extension will be the part of the operation in which the application programmer or the client programmer will be directly involved.

Illustrative Example

We use a university library system as our sample application system. This example (taken from a CD-ROM[9]) is a typical object-oriented application system, and widely used to describe the object-oriented analysis and design process. The analyst meets with the domain experts and users of the application system to create a use-case diagram to capture the application's main functional requirements, as shown in Exhibit 104.7.

A use-case diagram is employed to capture the main functional requirements of the application system. It is the tool for communication between the user and the analyst or developer. In Exhibit 104.7, the library system use case is shown as two primary actors of the system, which are librarian and borrower. The librarian is referred to as the internal actor and the borrower as the external actor. These two actors will need to be classified in more specific categories at the next stage of the analysis process.

The use-case diagram in Exhibit 104.7 will be used as the main source to create the software prototype. The multilevel menu interaction for the library system is to be constructed as shown in the diagram in Exhibit 104.8.

This operation structure diagram is very similar to the menu for the most-privileged user of an application system. The way that groups of operation are separated here is a suggestion, and not recommended to be used as a standard. Each application can have its own security policy and requirements that would also affect the grouping of operations. Therefore, analysts and developers should carefully analyze the specific application they are working on.

After the creation of the use-case diagram in Exhibit 104.7, the analysts and developers will make a decision to create another use case based on system security requirements. This use case is the new diagram created

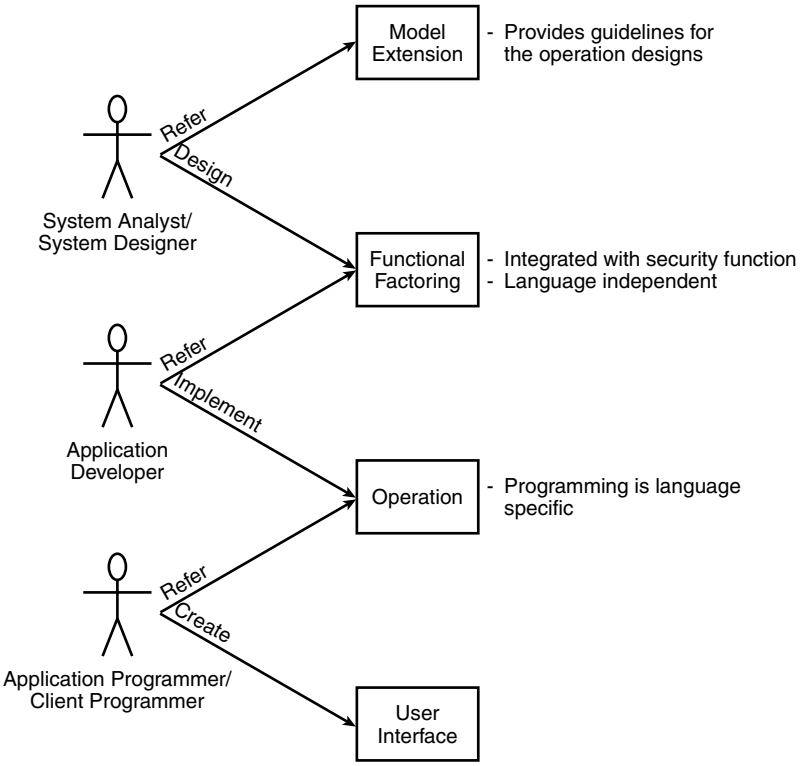


EXHIBIT 104.6 How the security model fits in the application system life cycle.

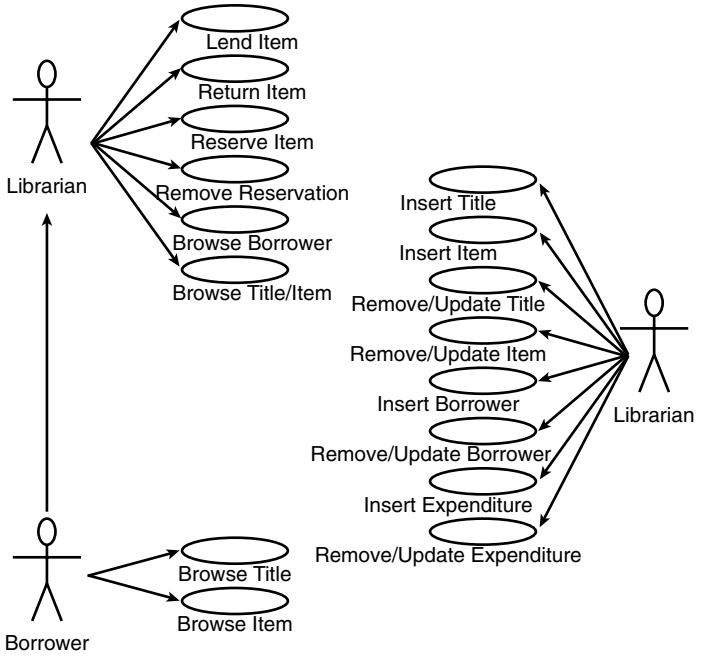


EXHIBIT 104.7 A use-case diagram for the library system, representing all operation group names.

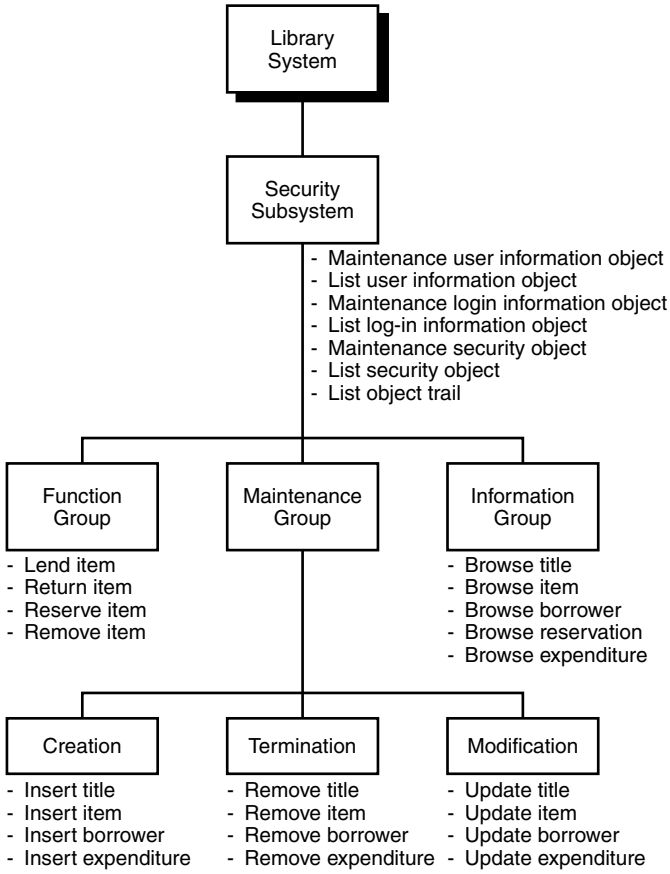


EXHIBIT 104.8 The multilevel menu for the software prototype.

with the main purpose of capturing the security function requirements. The relevant data come from several tools and diagrams:

- *Software prototype* is the starting point in the design of a security mechanism in an application system. The accurate security specification depends on the shared work between system user and analyst.
- *Role diagram* gives data concerning each user role that is significant in the security aspect.
- *Operation structure diagram* groups the operations by similarity in access privileges.
- *Sensitivity level diagram* presents data about each group of users' access privilege to the group of operations.

The use-case diagram for security purposes presents each role of the users and how each of them has a privilege to access the group of functions. With the help of this use- case, the analyst and developer will understand the role of each user more clearly toward security requirements and specifications. Exhibit 104.9 presents a use-case diagram for security purposes as mentioned above.

The separation of groups of functions in Exhibit 104.9 is only a suggestion. It is not meant to set an example for other applications. The diagram only shows the result of security specification through the software prototype. This diagram is created with the objective to help system analysts and developers clearly understand the role of each actor (user) in the operation group in an application system. The information from this diagram could give analysts and developers a basic understanding of how to design an information system that meets the security requirements.

Examples of how to integrate the security function with normal operations are shown in Exhibit 104.10 through Exhibit 104.13.

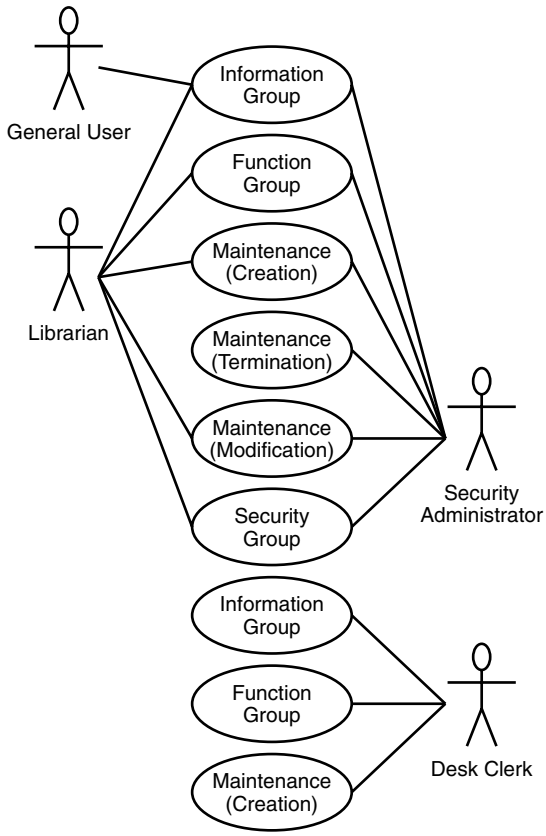


EXHIBIT 104.9 A use-case diagram for security purpose.

Query Operation

This displays the attribute value, for example, lists overdue books and borrowed books for a specific member (see Exhibit 104.10). The query operation will follow the encapsulation mechanism in an object-oriented paradigm. The suggestion is that all attributes of an object model should have access specifiers as "private" or can be accessed only by the function in the same class.

One of the security functions of the query operation that can be used as an extension of the model is "the permission test for operation invocation." This function is an important part if the analysis shows that this attribute is crucial for security purposes. Along with the permission test, another security function that can be implemented with it is the audit trail function. This function will record all necessary information for future reference.

Update Operation

This makes some modification to the attribute value, for example, changes the status of a member from a master to a doctoral student and changes the member's address (see Exhibit 104.11). This operation is also called an object state change.

The security requirement for an update operation is to ascertain that the object's attributes can be changed in a way that does not violate the relationship's rule among domain objects model. The general requirement is that the invocation of this function will be given only to an authorized user or operation group.

EXHIBIT 104.10 The Query Operation with Security Functions

Pre-Condition	
Type:	Permission test for operation invocation
Content:	Give permission only to an authorized user
Type:	Correctness of input data test
Content:	Test for query condition
Operation	
Type:	Query
Content:	List book's price
Post-Condition	
Type:	Correctness of output data test
Content:	Test for the result
Type:	Audit trail
Content:	Record all necessary information about transaction

EXHIBIT 104.11 The Update Operation with Security Functions

Pre-Condition	
Type:	Permission test for operation invocation
Content:	Give permission only to an authorized user
Type:	Correctness of input data test
Content:	Checking for the value of all attributes
Type:	State change permission test
Content:	The set of data values that the object can change its state to at a certain time
Operation	
Type:	Modify
Content:	Changing membership's status in Borrower Information object
Post-Condition	
Type:	Audit trail
Content:	Record all necessary information such as the authorized person who is permitted to invoke this operation
Type:	Notification
Content:	Notify the related objects, such as Loan, Reservation
Type:	State change permission test
Content:	Check for the object state and the output values after the method invocation

Terminate Operation

This operation terminates the object instance, for example, removes a damaged or lost book from circulation (see Exhibit 104.12). In an application system, while in an operation process, some events or transactions might result in the deletion of any object instances from the class. To do so, the terminate operation should have security functions as the pre- and post-conditions presented in Exhibit 104.12.

The terminator and destructor are not the same function. The destructor function cleans up the memory allocation for an object when the application system has finished its execution. The terminator's main objective is to terminate an object's instance from a specific object class according to some predefined rules. Because this is a very important function, and considering how this can relate to other objects in the same application system, the permission to use this function should be carefully checked.

EXHIBIT 104.12 The Terminate Function with Security Functions

Pre-Condition	
Type:	Permission test for operation invocation
Content:	Give permission only to an authorized user
Type:	Terminate membership test
Content:	Test that current state of book is not on Loan
Type:	Notification
Content:	Notify dependent object and update — Book Title — Item — Expenditure — Reservation
Operation	
Type:	Terminate
Content:	Remove damaged book
Post-Condition	
Type:	Audit trail
Content:	Record date/time of remove transaction, including reason

EXHIBIT 104.13 The Create Operation with Security Functions

Pre-Condition	
Type:	Correctness of input data test
Content:	Test for all attributes
Type:	Relationship cardinality test
Content:	Set relationship of book and title
Type:	Set membership test
Content:	Test for the population of object
Operation	
Type:	Create
Content:	Add new library book
Post-Condition	
Type:	Correctness of output data test
Content:	Test for all attributes

Create Operation

The create operation adds a new object instance, for example, buys a new book, receives a new dissertation, or receives a new CD-ROM (see Exhibit 104.13).

An application system with maximum-security requirements trades off usability against system performance. The security functions, such as an audit trail, provide higher security for each function but also consume much of the system execution time. Therefore, it is the system designer or security analyst who makes the final decision on whether to put minimal security requirements in the application system.

The update operation demonstrates how this model extension could be used, employing the Visual C++ syntax to illustrate the concept. Exhibit 104.14 shows a typical update operation without a security function.

In the university library system example, the group name of the user is very important from a security standpoint. The user interface menu differs, depending on the group to which each user belongs. Therefore, the function that changes this attribute's content should be carefully designed, using a security-oriented methodology. Exhibit 104.15 presents a way to design and implement the same update operation, using a security-oriented model extension.

Additional security functions that are added to the original update operation are `CheckGroupname()` and `CheckUserRight()`, and they are updated in the audit trail attribute. The audit trail attribute in this

EXHIBIT 104.14 A Typical Update Operation

```
void ChangeGroupnameNOSecurity(CString ChangeGroupname)
{
  m_Groupname = ChangeGroupname;
}
```

EXHIBIT 104.15 The Update Operation with Security Functions

```
ChangeGroupnameWSecurity(CString Groupname,
                         CString UserName,
                         CString UserGroup)
{
  CString Username, Message;
  CTime present_time = CTime::GetCurrentTime();
  CString newtime = present_time.Format("Data has been changed at
                    %H:%M:%S");
  CSecurityObj::CheckGroupname(Groupname);
  //Searching to compare "groupname" in
  //the security object which has list of
  //user group name that can access the system
  //This security function is called
  //"Correctness of input data test"
  CSecurityObj::CheckUserRight(UserGroup);
  //Checking for the right of user
  //who invokes this operation
  Message = "\n" + newtime;
  Message += "\nBy Username " + UserName+ " of " + " Group " +
             UserGroup;
  Message += "\nfrom " +m_Groupname+" to "+Groupname;
  m_Groupname = Groupname;
  m_BorrowerAudit = Message;
}
```

sample application is called `m_BorrowerAudit`. This attribute tracks all changes that are made to all of the attributes. Only the changes that have significant meaning from a security standpoint are recorded for future reference.

Exhibit 104.16 illustrates the output of these two update operations, showing how an extension object model can increase the security of each important function. The security of each function also contributes to the security of the overall system.

This example aims at keeping the program as simple as possible. By omitting unnecessary features such as a graphical user interface and persistent utilities, the program is easy to understand but still clearly illustrates the model extension. Implementation of the model extension may vary with the operation, and the security function details will differ somewhat.

Conclusions and Future Work

The objective of this article is to propose a security model, OOSM, that can be used as a model to analyze and design a security-oriented information system. The process starts with the design of a domain object model. As a result, each domain object will have the additional security functions as a model extension. These security functions will be implemented as the pre- and post-conditions to the typical function in an object model. The guidelines for how each type of security function can be used with the typical function are described in the article. The analyst or developer can utilize the model throughout the process of system analysis and design. The application system resulting from this model should satisfy most organizational security requirements.

The implementation process described in this article uses the Visual C++ programming language. No special language features were used, to keep the program as simple as possible, while providing enough detail to show

EXHIBIT 104.16 The Output of an Update Operation with Security Function

```
Please enter user name (prog1/prog2/prog3) prog1
Group Name   : Student  Borrower ID  : IMD979813
Borrower Name: Sureerat Borrower Addr: SV9B
Borrower Audit:
-------------------------------------------------
Enter Group Name you want to change? Librarian
Librarian Group name is correct
The user has no right to invoke this operation
Press any key to continue
Please enter user name (prog1/prog2/prog3) prog3
Group Name   : Student  Borrower ID  : IMD979813
Borrower Name: Sureerat Borrower Addr: SV9B
Borrower Audit:
-------------------------------------------------
Enter Group Name you want to change? Librarian
Librarian Group name is correct
The user has a right to invoke this operation
After invoking change Group name with security function
=================================================
Group Name   : Librarian        Borrower ID  : IMD979813
Borrower Name: Sureerat Borrower Addr: SV9B
Borrower Audit:
Data has been changed at 20:06:07
By Username prog3 of  Group Security
from Student to Librarian
Press any key to continue
```

how to implement the model extension. The evaluation process of this model, however, could not be measured on an empirical basis. But when compared with different software process models such as the Spiral model and the Waterfall model, the OOSM has the benefit of giving the analyst or developer a better understanding of how to integrate the system's security requirements in the early phase of system development. The OOSM also solves the problem of retrofitting the security mechanism into an application system that is already developed.

The use of OOSM along with the traditional process model, OOSAD, could enhance the security of the overall application system. What we thought of as a problem at first, the success of the security mechanism depending on an individual analyst or developer, could not be solved entirely. The model provides some design guidelines and also additional tools and techniques to help solve the design problem. There are still some difficulties when mapping the design into the implementation process. The problems vary according to the experience of developer, the programming language used, the nature of the problem domain, and the specific security requirements.

Solving the design problem requires more than providing the design guidelines and tools. We planned to move the OOSM from guidelines to the Design Pattern. Gamma et al.[8] explained the meaning of the design pattern as "*descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context.*" By creating the design pattern for security-oriented systems, the analyst or developer can get the pattern and implement it in a more efficient way. The design pattern also helps prevent the developer from inaccurate interpretation of any guidelines. The programming language source code should accompany each pattern in order to give the developer a better understanding of the pattern.

References

1. IBM Inc., June 2002, "IBM WebSphere: WebSphere Software Platform," http://www-3.ibm.com.
2. BEA Systems Inc., June 2002, "Product Brief: BEA WebLogic Server," http://www.bea.com.

3. Dewan, P. and Shen, H., 1998, "Controlling Access in Multiuser Interface," *ACM Transactions on Computer-Human Interaction,* 5(1), 34, March 1998.
4. Richardson, J., Schwarz, P., and Cabrera, L., "CACL: Efficient Fine-Grained Protection for Objects," *OOPSLA'92 Conference Proceedings. Object-Oriented Programming Systems, Language and Applications,* 27(10), 263, 1992.
5. Overbeck, J. and Stary, C., 1995, "What Designers Need to Know about Privacy," *TOOLS USA '95 (Technology of Object-Oriented Languages and Systems),* Prentice Hall, p. 115.
6. Baskerville, R., 1993. "Information Systems Security Design Methods: Implications for Information Systems Development," *ACM Computing Surveys,* 25(4), 375, December 1993.
7. Fayad, M., Schmidt, D., and Johnson, R., 1999, *Building Application Frameworks: Object-Oriented Foundations of Framework Design,* John Wiley & Sons.
8. Gamma, E., Helm, R., Johnson, R., and Vlissides, J., 1995, *Design Patterns: Elements of Reusable Object-Oriented Software,* Addison-Wesley Publishing Company.
9. Eriksson, H. and Penker M., 1996, *UML Toolkit,* John Wiley & Sons.
10. Krief, P., 1996, *Prototyping with Objects,* Prentice-Hall International.

107

Methods of Auditing Applications

David C. Rice, CISSP and Graham Bucholz

Introduction: Ubiquitous Insecurity

The microprocessor — the computer — is the seventh simple machine. Like its predecessors, the wheel, the incline plane, and the lever, the microprocessor performs simple tasks, and therefore makes it easier to accomplish more. Moreover, as production costs and the size of the processors shrink, the silicon chip is becoming inexpensive and tiny enough to slip into every object we manufacture.

As the number of devices containing microprocessors increases, so too will the impact on daily life. Personal computers, the most popular and well known of devices containing a microprocessor, are only one example, but there are many, many others as well. Microprocessors are embedded in everything: cell phones, watches, microwave ovens, automobiles, stereos, and even rice cookers. These "noncomputer" chips already number in the billions. Devices are getting smarter and smaller, but there is more to the story: a single microprocessor can only do so much. Sure it may be fast, and the microprocessor may be smart, but the technological revolution occurs when microprocessors start talking to one another. In other words, the microprocessor by itself is an impressive invention, but interconnected microprocessors, well, that is momentous. Whether personal computer, BlackBerry, PalmPilot, AutoPC, or refrigerator, we are attempting to connect everything to everything else through copper, radio, infrared, and fiber.

Of course, distributed and decentralized computing is nothing new, but it is the scope and scale of microprocessor technology and communication protocols over the last three decades that has allowed decentralized computing to attain new heights. Microprocessors are talking on more devices than ever before, but more importantly these microprocessors are *listening* on more devices than ever before. This grand network of protoconsciousnesses is creating an environment of ubiquitous computing and pervasive connectivity that surrounds and infuses the everyday life of humanity. Underlying this marvelous development of universal computing, however, is something completely transparent to the everyday user: software.

"Connecting all to all" becomes possible only because software, or code, *makes* it possible. Paralleling the rapid expansion of microprocessors into virtually all areas of our business and private lives is the expansion — and dependency — on code. Wherever microprocessors can be found, so must software.

Most computer users probably have not heard of languages like C, C++, Java, and COBOL. If they have, they most likely shun the very mention of them, warily avoiding such cryptic lexicons. Yet these languages and many, many others shape the function — and ultimately the devices — that serve humanity.

If microprocessors are finding their way into our traffic lights, medical devices, airplanes, homes, business supply chains, enterprise management systems, transportation systems, and household appliances, then so too is software. Ubiquitous computing *means* ubiquitous software. Code therefore is quickly becoming the foundation of civilization.

As reliance on software grows, so do the consequences of software failure. If code is becoming the foundation of civilization, then civilization is only as durable as the code.

A majority of consumers would never settle — let alone pay for — homes, automobiles, or buildings constructed as poorly as many software applications are today. Software bugs seem to be an accepted part of the computing environment. However, if the software is buggy, what does that say about the software's security? Bugs are indicative of a greater problem, yet they are often eschewed as "the cost of doing business."

The heavier the reliance on software in our everyday existence, the higher the exposure to risk if that software should fail or be leveraged for malicious intent. Couple this risk to a highly networked, distributed environment — an environment that almost insists on pervasive communications — and the potential for havoc becomes highly feasible. If ubiquitous computing means ubiquitous software, then ubiquitous software means ubiquitous insecurity.

Perhaps the reader's first inclination is to state the effectiveness of firewalls, intrusion detection systems, and virus detectors against insecurity. Ironically, many of these security applications are no better designed or implemented than the applications they are attempting to protect. If the software on security systems is flawed, so is the security the device provides. However, too much faith in security systems or encryption masks the real problem. Firewalls and intrusion detection systems are really just a network response to a software-engineering problem, and for the most part, do not and cannot protect from ZeroDay[1] events. This is not to say security systems are entirely useless. Security systems can be a valuable addition to a network's defense, but do nothing to solve the problem of insecurity, only delay it.

Ubiquitous insecurity stems from our unwillingness and inability to unravel the software-engineering predicament at its root: code. The pronoun "our" in the previous sentence is left purposely nebulous because the software-engineering problem belongs to everyone — government, industry, consumer, and developer. As long as insecure code is developed and purchased, whether by private consumer, corporate entity, or government institution, ubiquitous insecurity will imperil the foundations of the civilization being built today. This is not to say that the future is unequivocally doomed; every civilization has faced a foreboding, dark shadow threatening its very survival, but few civilizations have willingly created and installed a nemesis within their fledgling critical infrastructure.

Although the seventh simple machine can be a great servant to humanity, it can also be an appalling master if not supervised appropriately. The inventor of the wheel could never imagine to what ends the wheel would be used, no more than the future utilization of the microprocessor can be foreseen at this moment in time. We must create software worthy of the title "foundation."

Scope of Discussion

This chapter is intended to inform technical managers and developers about the mistakes and bad coding practices that make ubiquitous insecurity a reality. What follows in this chapter is a description of vulnerability discovery methods or attack techniques used to audit and evaluate applications for insecurities. These same techniques can also be used to subvert applications for gain, curiosity, or otherwise. In no way do the authors encourage illegal behavior.

The methods discussed in this chapter apply mainly to binary applications, and do not address Web applications or Web services directly, though some techniques may be leveraged to do so. Web applications are avoided as a topic of discussion mainly because they are site-specific and techniques are not easily generalized.

Every section deserves to be its own book, but by necessity only a subset of relevant topics can be discussed within the limits of a single chapter. Therefore, exhaustive technical depth must give way to brevity in a majority of this discussion; the reader will not be able to put down this book and immediately begin subverting applications. However, the authors have made every attempt to keep this chapter meaningful and informative.

Setting the Stage

Meaningful vulnerability discovery requires a nontrivial skill set, one that requires an extraordinary amount of patience, time, resources, and exhaustive technical knowledge to acquire. The world of vulnerability discovery is not for the indolent or the faint of heart, nor is that world abundantly populated. To some extent, the difficulty in acquiring the necessary skills to discover unique and original vulnerabilities should comfort those who use the digital world on a daily basis. In laymen's terms, vulnerability discovery is not an amateur endeavor. However, these skills are not impossible to learn, and even a rank amateur can attain some modicum of success.

As code infiltrates and delineates our critical infrastructure, more individuals will be enticed to acquire these skills.

In this section, we summarize the required knowledge base for a software/application auditor to understand binary applications, the tools required, and the crucial mindset for executing successful vulnerability discovery. There always will be exceptions to this list, and also unfortunate omissions, but what follows is a good starting point.

Mindset: "There Is No Box"

> The world is seamless, with no boundaries or dividing walls…
>
> **— Ikkyu, Abbot, Buddhist Daitokuji Monastery, Kyoto, Japan**

The foundation for continued, successful vulnerability discovery is the right mindset. Although it seems most of corporate culture, political leadership, and mid-level managers spend time striving to "think-outside-the-box," great hackers[2] — truly great hackers — know *there is no box*.

This apparently esoteric point is absolutely necessary to understand why great hackers are so good at discovering vulnerabilities or subverting applications, networks, and just about anything else they get their hands on. It is also absolutely necessary to understand the concept of "no box" to comprehend why corporate leadership feels hoodwinked when their intranet gets compromised despite liberal firewall placement.

"No Box"

The "box" is simply the identification of "what is possible or acceptable," based on a given body of knowledge or assumptions. What is considered possible articulates the boundaries of the box. The paradox of "thinking outside the box" is the box immediately expands to include that which escapes it; that is, original thought is quickly burdened with the onus of formulization and imitation.

Boundaries, or boxes, are created by the human mind for the benefit of perception; the mind *must* classify, it must distinguish between good and bad strategies, between "this" and "that" for a matter of survival, but reality is by no means ruled by the mind's perceptions.

Great hackers comprehend the digital world, like the real world, as "seamless, with no boundaries or dividing walls." The digital world is not cordoned off by firewalls nor defined by applications. The digital world is not illuminated by intrusion detection systems nor bounded by user interfaces. The digital world is influenced by these abstractions, no doubt; but the digital world is not beholden to any authority save one — code. Code determines how bits, the 1s and 0s, are created, stored, and transformed into usable information humans can digest. Code is law in the digital world, but it is not absolute. In other words, boxes are a manifestation of code, but code transcends the nature of boxes. Unlike in the real world, where the gravitational constant in one part of the universe is the same as in another, code determines which rules are applied in the digital world and to what extent.[3] Changing the code changes the rules. In a sense, there are those individuals who are impressed with their ability to "think outside the box," and then there are those who create the boxes in the first place.

Often, in the authors' evaluation of software applications, the comments "that's not what the application was designed to do" or "you shouldn't be able to do that," have been heard regularly. From the client's perspective, this is certainly true, but only from that singular perspective. A developer looks at an application as a collection of well-behaved components. A user sees applications as a collection of desktops, windows, and icons. A network administrator sees the network as an amalgam of switches, routers, and proxies.

However, the digital world is by no means ruled by these perceptions. Great hackers see the digital world without the assumptions placed on it by developers, users, and marketing divisions; great hackers see through convenient distinctions as the illusory boxes they are. Hackers see bits only[4] as perhaps scientists see only matter or energy.

Great hackers, however, are not all-powerful deities roaming the digital landscape, changing the rules at whim. For the most part, such a description is inappropriate, but not wholly inaccurate. Code can be a great servant of mankind, but it can also be an appalling master, even to those who know its nuances. Acknowledging "no box" is an important realization, but one that does not confer magical powers upon the enlightened. What is essential after this relatively inexpensive epiphany is a strong, practical foundation in the skills software developers possess.

Knowledge Set

Knowledge of the intended target is vital. In large part, the required knowledge set is target dependent, and increases in importance the deeper into the technical architecture one travels. While the "no box" mindset may permit the application attacker to view the digital world in an entirely different way, the current rules (i.e., code) in place must first be understood before they can be altered.

The first requirement is to identify the target's platform. A platform is defined by a combination of a microprocessor architecture (Intel, Motorola, AMD, etc.) and an operating system (Windows, Linux, MacOS, etc.). It is not necessary to understand the target platform in its entirety — a task that is almost impossible for any single person — but it is essential to understand a majority of the platform's functional aspects, including input/output, security implementation (if any), file access, memory management, and process creation.

The second requirement is knowledge of a programming language. Languages such as C/C++ are most common, but knowledge of other languages such as Java, COBOL, and Ada may be required; which language is necessary will depend on the target application. Also, knowledge of the assembly language the microprocessor architecture executes can be extremely helpful.

Programming languages are the prime vehicles for exploring a target in-depth. Knowledge of how applications are designed and written assists in analysis. The public interface conceals much of the underlying operations an application performs. The more adept an auditor is at programming, the more portions of an application unexposed by the public interface may be examined. Additionally, programming skills may accelerate the vulnerability discovery processes by automating many common testing procedures and, if a flaw is discovered, to verify the flaw's potential as a vulnerability. Without a doubt, programming skills will augment the auditor's tool set.

The third requirement is knowledge of communication protocols, both network and host-based. TCP/IP is the most common network communication protocol and any application capable of internetworking will usually employ it, but knowledge of other network protocols, such as NetBIOS and IPX/SPX, may be required. The requisite network protocol will depend on the target application.

Host-based communication protocols are those that involve intra-computer communication such as interprocess communication (IPC) or serial/parallel ports. This is one area where developers often devise their own proprietary protocols; however, understanding standard protocol implementations, such as TCP/IP, along with their respective strengths and weaknesses, will help in deciphering and analyzing these proprietary protocols.

The fourth and final requirement is a willingness to learn. It takes time and effort to acquire this body of knowledge and apply it accordingly. Although a computer science degree would be helpful in learning the above-mentioned requirements, it is not mandatory. Knowledge can be acquired by anyone. Every individual has the potential to become a proficient vulnerability researcher with work and practice; a degree is not necessary, it just lowers the learning curve.

Tools of the Trade

Possessing the basic knowledge described previously is often not enough; having the proper tools is also important. Tools of the trade not only include specialized software applications, but also the people you know and the books you read.

There is a number of specialized software applications available free from the Internet or for purchase on the open market. These tools allow auditors to increase their understanding of a particular system. A number of the tools application developers utilize to debug their applications are similarly useful for the auditor in analyzing the same application. Two such tools for software auditing come to the forefront: Numega's SoftICE and DataRescue's IDAPro.

SoftICE is a dynamic debugger for Intel's x86 architecture, capable of interrupting an application while it is executing, permitting the examination of the application's current internal state. This is especially useful for examining current operations that the application is performing that are not observable through the public interface.

IDAPro is an interactive static disassembler capable of displaying the operations for more than 30 different microprocessor architectures on which an application may execute. Much like SoftICE, IDAPro allows the auditor to view operations not observable through the public interface; however, unlike SoftICE, IDAPro examines the entire application without execution. By loading an executable into IDAPro, the auditor may view all possible instructions an application may perform in an easily readable document-like format. However,

IDAPro does not support run-time evaluation so the auditor cannot view which instructions are actually executed.

Other tools frequently needed are binary editors, network protocol analyzers, and various forensic programs and devices to display the current state of the system. Binary editors are frequently used to modify programs or files that reside on disk. Forensic programs provide a window into the system's current state without altering data or interrupting program execution. Network protocol analyzers allow an auditor to capture and view inter-application network traffic. Whichever tools the auditor selects is usually dependent on the target environment, economic factors (some tools are more expensive than others, and usually a similar freeware program can be found), and personal preferences.

The expertise of others is one tool most often overlooked. As stated earlier, most modern applications and operating systems are too complicated for any one person to know everything in detail. However, there are experts on facets of every platform, willing to share their insight. This sharing usually manifests in newsgroups, mailing lists, lectures, application documentation, and books, books, books. Usually a good starting point to answer any question may be found in one of the aforementioned forums.

Attack Methodology

An Art Built on a Science

Currently, auditing programs is still more of an art than a science. There is no "right way" to go about probing an application for security vulnerabilities. Although the methodology for attacking applications mirrors the scientific method, it also has a lot to do with intuition, viewpoint (i.e., "no box"), previous experience, and innovation. These four traits make successful auditors. However, without the patience of a scientist and the critical mindset, most auditors would simply yield to frustration.

Information Gathering

The first step in any process in auditing an application is gathering as much information as possible. Without defining and describing the target, it is difficult to see the full picture, and obvious flaws might be missed. The first place to look is product documentation. Documentation is a great way to see how developers presume their product is supposed to work, and is usually available for applications in varying forms and degrees. Usually, documentation regarding the internal structure of an application is unavailable, but information on a majority of the public interfaces and functionality is included for the benefit of the average user.

After the basics of the application are understood, other information should be gathered to flesh out the picture and focus the search. Most modern applications for personal computers are so large that trying to examine the entire package at once is not feasible, especially if there is a deadline. Good places to look for more information are newsgroups and mailing lists for any mention of the product. Reading other users' experiences can lead to insight into how the product is actually being used (or misused) in the real world. Also, any mention of difficulties or problems using the application should be noted, as this might be indicative of a flaw in the application.

As well as looking for information on the specific product, looking for information on similar products and on different products by the same vendor/developer also can lead to insights. Certain types of applications have specific concerns, regardless of the vendor who created the application, so difficulties in a different application might lead to ideas as to what to explore in the evaluated application. The same concept can be used with different programs from the same vendor. Often applications from a vendor are created by the same developers, or by developers that program with the same corporate mindset. So flaws found in other, unrelated products by the same vendor might also lead to thoughts as to where to focus attention in the targeted application.

Mainly, the purpose of this step is to gain a thorough understanding of the application, and uncover as many potential problems as possible. All this information is then fed into the next step: analysis.

Analysis

Once the raw information is gathered from the preceding step, it must be collated and whittled down into a number of specific areas that might be vulnerable to attack. How this narrowing of possibilities is done is most

often based on the past experiences of the person performing the application audit. There really is no right or wrong way to complete this step. Often, the information from previously discovered vulnerabilities is reused against the current application, such as testing user inputs for buffer overflows. Truly unique vulnerabilities are discovered most often by understanding the application as specified in the documentation and then observing the application acting in an inconsistent way. By definition, these types of vulnerabilities are the hardest to find because there is no historical precedent for them. They must be discovered by understanding how the system works, how the application is actually working (as opposed to how the documentation says it works), and often by a good dose of luck.

Once the list of possible vulnerable areas is sorted based on probability of success and resources available, the list is then used in the next step: hypothesis.

Hypothesis

From the last step, a list of possible vulnerabilities was produced. For each of these, a hypothesis should be generated. A hypothesis allows the parameters for each vulnerability to be specified, making it easier to both develop a test for the vulnerability and to more easily see what assumptions may have caused the test to fail. Also, having a semiformal statement of what is being considered is good for documenting the actual testing. Nothing is worse than coming back a few months after an audit, or being handed someone else's work, and not knowing what was done. Usually a hypothesis takes the form, "If we do X, then Y will (or will not) occur." In the application-testing arena, an example may be, "if a large string is entered into a specific field, then an access violation will occur." A true hypothesis would be more specific than that example, but that is the idea. These hypotheses can then be developed into actual tests to be run against the application.

There are a number of common classes of vulnerabilities that should be looked for. Following is our list of them. It is by no means a complete list.

Input Validation

Previously highlighted as an example, input validation tests whether an application properly handles input from an external source. In this context, an external source could be the user, another program, operating system, or anything outside the application destined for internal processing. The most common types of input validation errors result in buffer overflow, format string, or denial-of-service exploits. Because developers can accidentally overlook input validation, this type of error occurs frequently.

Most often this form of testing is accomplished by sending varying amounts of data — both properly and improperly formatted — into an application and viewing the results. Application response will help determine if this application is potentially vulnerable to the aforementioned exploits.

Angry Monkey

In this method, an automated program randomly performs input validation against the target. Angry Monkey, as any other input validation test, focuses on the application's ability to handle input; however, no criteria are established for external interfaces of the application. Any component of the application may be tested with randomly generated data, in no particular order or for any particular reason.

Session Management Validation

Network applications need to manage multiple conversions with numerous communication partners often at the same time. State variables such as session IDs, cookies, and secret keys uniquely identify these sessions. These variables are often randomly generated values that are assigned to a particular communications channel for a limited period of time.

Testing an application for session management vulnerabilities consists of attempting to guess, capture, and modify any of these state variables to elicit undesirable results from the application. By altering these variables, access may be gained to other communication channels that could lead to privilege escalation, loss of privacy or data confidentiality, or unauthorized access to resources.

Race Condition Analysis

Applications perform numerous operations in the course of completing any function, including security-related functions. In general, a race condition exists when there is a window of time between a security operation and the general function it applies to. This window of opportunity can allow security measures to be circumvented. An example of this is an application first creating a new file and then applying security to that file. Racing the

application attempts to access the file between the time the application creates it and when it actually applies the security. Identifying and testing for race conditions can be difficult due to very short windows of opportunity.

Cryptographic Analysis

Applications may handle sensitive data, such as passwords, credit card information, company trade secrets or intellectual property, or private personal information. This data is frequently protected by cryptographic methods. There are a lot of different cryptographic algorithms available for applications to use, both public and private. Experts have created and extensively examined some of them, and the vendors themselves have developed others. Those subjected to public scrutiny by experts are believed to be much stronger and more resilient to attack than private algorithms created by vendors. Determining what algorithm an application uses may lead to knowledge of its strengths and weaknesses. However, regardless of the strength of the algorithm used by the application, if the vendor uses it incorrectly, the data may not be protected as advertised. Errors could include improper creation, handling, or storage of the cryptographic keys. Examination, then, needs to include both the algorithm itself and the key management mechanism.

Code Coverage Analysis

Applications need to make numerous decisions in the course of performing their tasks. Each end result of these decisions should be secure. Code coverage analysis usually employs source code (or disassembled code) to ensure that proper security measures are taken on all possible paths of execution. There may exist execution paths through the application that allow for security to be bypassed, leaving the system in a vulnerable state. This analysis can take an extremely large amount of resources, both in people and time, depending on the size and complexity of the application. If at all possible, this type of analysis should be done in stages during the development of an application before it is ever considered ready for production.

Testing

The final step is taking the first hypothesis and actually testing it. How this is exactly accomplished all depends on both the application and the hypothesis. Sometimes it can be as simple as changing a setting and observing the effect. Other times a complex set of interactions between the application, the system, and possibly some custom-designed code must be choreographed.

Additionally, because applications today are such complex pieces of code, the results of testing a hypothesis can as varied as all the possible tests. However, if the hypothesis was sufficiently developed before testing, success or failure should be fairly easy to determine.

The most difficult part of testing is not finding vulnerability, though. It is proving (at least to whatever level of satisfaction required) that the application is not vulnerable to a specific test. If the test failed to prove the hypothesis, then the next step is to decide whether it failed because the parameters and assumptions being operated under were invalid, or because the hypothesis is wrong. In the previous example, if a long string is entered and does not cause an access violation, is it because the input was correctly handled, or was the string not long enough? Questions like this must be considered, and the hypothesis must be restated to correct any faults, or the results must be accepted and the next hypothesis on the list can be addressed. How concerns like this are handled are more often a matter of policy than of a technical nature.

Conclusion

Software development is an error-prone process; flaws inevitably creep into any product despite quality control efforts. The prevalence of software in nearly every aspect of modern life leads to reliance on software and as that reliance grows, so do the consequences of software failure or exploitation. No one can say when or why an application will be attacked, so finding and preventing these failures before they occur becomes an important endeavor.

Remember, application auditing is a nontrivial task that requires a special set of knowledge, skills, and resources. While it does not take a genius to succeed, it does require focused effort, patience, and a little bit of luck. The information and methodology described herein are good first steps toward learning what is required. However, it needs to be said that there is not, nor will there ever be, a last step when it comes to application auditing. There is no single solution to solving the problem of insecure applications. Even by

auditing an application, there may remain undiscovered weaknesses that will surface months, years, or even decades later. Every weakness found and fixed, however, is one less that threatens the stability of modern life.

Notes

1. ZeroDay events refer to a newly released exploit into the public domain for which no signature is available to identify it. Because security devices are in large part knowledge-based, the security device must have knowledge of the exploit to protect against it. If the security device is not aware of the exploit, it cannot protect against it until a signature is made available. For those exploits that are not made public, most security devices are unable to protect their respective networks from exploitation.
2. The authors purposely avoid distinguishing "hackers" from "crackers," mostly due to the amount of paper wasted explaining the difference between the two. The Dark Side of the Force can seduce great hackers; get over it.
3. This point is especially meaningful with the introduction of XML. XML can describe all the information about Mozart's Symphony No. 40. A user might want to listen to the file or print out the sheet music, but depending on what the user wants, data is transformed appropriately to meet the request.
4. In the physical world, manipulating atoms is not practical for the average human being. We see a cup, move it, drink from it, break it, but we are handicapped about altering how the atoms form the cup. If the cup were made out of bits, however, we could alter each bit, perhaps changing the color of the cup, or even making the cup into a song or picture. In the digital world, you can do anything you want with bits; shape, form, even behavior are not immutable.

106

Malware and Computer Viruses

Robert M. Slade, CISSP

Malware is a relatively new term in the security field. It was created to address the need to discuss software or programs that are intentionally designed to include functions for penetrating a system, breaking security policies, or carrying malicious or damaging payloads. Because this type of software has started to develop a bewildering variety of forms such as backdoors, data diddlers, DDoS, hoax warnings, logic bombs, pranks, RATs, Trojans, viruses, worms, zombies, etc., the term *malware* has come to be used for the collective class of malicious software. The term is, however, often used very loosely simply as a synonym for virus, in the same way that virus is often used simply as a description of any type of computer problem. This chapter attempts to define the problem more accurately and to describe the various types of malware.

Viruses are the largest class of malware, both in terms of numbers of known entities and in impact on the current computing environment. Viruses will, therefore, be given primary emphasis in this chapter but will not be the only malware type examined.

Programming bugs or errors are generally not included in the definition of malware, although it is sometimes difficult to make a hard and fast distinction between malware and bugs. For example, if a programmer left a buffer overflow in a system and it creates a loophole that can be used as a backdoor or a maintenance hook, did he do it deliberately? This question cannot be answered technically, although we might be able to guess at it, given the relative ease of use of a given vulnerability.

In addition, it should be noted that malware is not only a collection of utilities for the attacker. Once launched, malware can continue an attack without reference to the author or user; and in some cases it will expand the attack to other systems. There is a qualitative difference between malware and the attack tools, kits, or scripts that have to operate under an attacker's control and which are not considered to fall within the definition of malware. There are gray areas in this aspect as well, because RATs and DDoS zombies provide unattended access to systems but need to be commanded in order to deliver a payload.

Potential Security Concerns

Malware can attack and destroy system integrity in a number of ways. Viruses are often defined in terms of the ability to attach to programs (or to objects considered to be programmable) and so must, in some way, compromise the integrity of applications. A number of viruses attach themselves to the system in ways that either keep them resident in the system or invoke them each time the system starts, and they compromise the overall system even if individual applications are not touched. RATs (remote-access Trojans/tools, basically remotely installed backdoors) are designed to allow a remote user or attacker to completely control a system, regardless of local security controls or policies. The fact that viruses modify programs is seen as evidence that viruses inherently compromise systems, and therefore the concept of a *good* or even *benign* virus is a contradiction in terms. The concept of good viruses will be discussed more in the detailed section concerning virus functions.

Many viruses or other forms of malware contain payloads (such as data diddlers) that may either erase data files or interfere with application data over time in such a way that data integrity is compromised and data may become completely useless.

In considering malware, there is an additional type of attack on integrity. As with attacks where the intruder takes control of your system and uses it to explore or assail further systems in order to hide his own identity, malware (viruses and DDoS zombies in particular) is designed to use your system as a platform to continue further assaults, even without the intervention of the original author or attacker. This can create problems within domains and intranets where equivalent systems trust each other, and it can also create bad will when those with whom you do business find out that your system is sending viruses or probes to theirs.

As noted, malware can compromise programs and data to the point where they are no longer available. In addition, malware generally uses the resources of the system it has attacked; and it can, in extreme cases, exhaust CPU cycles, available processes (process numbers, tables, etc.), memory, communications links and bandwidth, open ports, disk space, mail queues, etc. Sometimes this can be a direct denial-of-service (DoS) attack, and sometimes it is a side effect of the activity of the malware.

Malware, such as backdoors and RATs, is intended to make intrusion and penetration easier. Viruses such as Melissa and SirCam send data files from your system to others (in these particular cases, seemingly as a side effect of the process of reproduction and spread). Malware can be written to do directed searches and send confidential data to specific parties, and it can also be used to open covert channels of other types.

The fact that you are infected with viruses, or compromised by other types of malware, can become quite evident to others. This compromises confidentiality by providing indirect evidence of your level of security, and it may also create seriously bad publicity.

The Computing Environment With Regard to Malware

In the modern computing environment, everything — including many supposedly isolated mainframes — is next to everything else. Where older Trojans relied on limited spread for as long as users on bulletin board systems could be fooled, and early-generation viruses required manual disk and file exchange, current versions of malware use network functions. For distribution of contemporary malware, network functions used can include e-mail of executable content in file attachments, compromise of active content on Web pages, and even direct attacks on server software. Attack payloads can attempt to compromise objects accessible via the Net, can deny resource services by exhausting them, can corrupt publicly available data on Web sites, or spread plausible but misleading misinformation.

It has long been known that the number of variants of viruses or other forms of malware is directly related to the number of instances of a given platform. The success of a given piece of malware is also associated with the relative proportion of a given platform in the overall computing environment. Attacks are generally mounted at least semirandomly; attacks on incompatible targets are wasted and, conversely, attacks on compatible targets are successful and may help to escalate the attack.

Although it may not seem so to harried network administrators, the modern computing environment is one of extreme consistency. The Intel platform has severe dominance in hardware, and Microsoft has a near monopoly of operating systems and applications on the desktop. In addition, compatible application software (and the addition of functional programming capabilities in those applications) can mean that malware from one hardware and operating system environment works perfectly well in another.

The functionality added to application macro and script languages has given them the capability either to directly address computer hardware and resources or to easily call on utilities or processes that have such access. This means that objects previously considered to be data, and therefore immune to malicious programming, must now be checked for malicious functions or payloads.

In addition, these languages are very simple to learn and use; and the various instances of malware carry their own source code, in plaintext and sometimes commented, making it simple for individuals wanting to learn how to craft an attack to gather templates and examples of how to do so — without even knowing how the technology actually works. This enormously expands the range of authors of such software.

Overview and History

We are faced with a rapid evolution of computer viruses, and we are experiencing difficulties in addressing the effects of these viruses, just as in the biological world. IBM's computer virus research team has extensively examined the similarities and differences between biological and computer viruses and epidemiology. Many excellent papers are available through their Web site at http://www.research.ibm.com/antivirus/.

The evolution of computer viruses is dramatically accelerated when compared to the development of their biological counterparts. This is easy to understand when you examine the rapid development of computer technology as well as the rapid homogenization of computers, operating systems, and software.

Many claims have been made for the existence of viruses prior to the 1980s, but so far these claims have either been unaccompanied by proof or have referred to entities that can be considered viruses only under the broadest definition of the term. The Core Wars programming contests did involve self-replicating code, but usually within a structured and artificial environment. Examples of other forms of malware have been known almost since the advent of computing.

At least two Apple II viruses are known to have been created in the early 1980s. Fred Cohen's pioneering academic research was undertaken during the middle of that decade, and there is some evidence that the first viruses to be successful in the normal computing environment were created late in the 1980s. However, it was not until the end of the decade (and 1987 in particular) that knowledge of real viruses became widespread, even among security experts. For many years, boot-sector infectors and file infectors were the only types of common viruses. These programs spread relatively slowly, primarily distributed on floppy disks, and were thus slow to disseminate geographically. However, the viruses tended to remain in the environment for a long time.

During the early 1990s, virus writers started experimenting with various functions intended to defeat detection. (Some forms had seen limited trials earlier.) Among these were polymorphism, to change code strings in order to defeat scanners, and stealth, to attempt to confound any type of detection. None of these virus technologies had a significant impact. Most viruses using these advanced technologies were easier to detect because of a necessary increase in program size.

Although demonstration programs had been created earlier, the middle 1990s saw the introduction of macro and script viruses in the wild. These were initially confined to word-processing files, particularly files associated with the Microsoft Office suite. However, the inclusion of programming capabilities eventually led to script viruses in many objects that would normally be considered to contain data only, such as Excel spreadsheets, PowerPoint presentation files, and e-mail messages. This fact led to greatly increased demands for computer resources among anti-viral systems because many more objects had to be tested, and Windows OLE (Object Linking and Embedding) format data files presented substantial complexity to scanners. Macro viruses also increased new variant forms very quickly because the viruses carried their own source code, and anyone who obtained a copy could generally modify it and create a new member of the virus family.

E-mail viruses became the major new form in the late 1990s and early 2000s. These viruses may use macro capabilities, scripting, or executable attachments to create e-mail messages or attachments sent out to e-mail addresses harvested from the infected machine or other sources. E-mail viruses spread with extreme rapidity, distributing themselves worldwide in a matter of hours. Some versions create so many copies of themselves that corporate and even service provider mail servers are flooded and cease to function. Prolific e-mail viruses are very visible and thus tend to be identified within a short space of time, but many are macros or scripts and generate many variants.

With the strong integration of the Microsoft Windows operating system with its Internet Explorer browser, Outlook mailer, Office suite, and system scripting, recent viruses have started to blur the normal distinctions. A document sent as an e-mail file attachment can make a call to a Web site that starts active content, which installs a remote-access tool acting as a portal for the client portion of a distributed denial-of-service network. This convergence of technologies is not only making discussion more difficult but is also leading to the development of much more dangerous and (from the perspective of an attacker) effective forms of malware.

Because the work has had to deal with detailed analyses of low-level code, virus research has led to significant advances in the field of forensic programming. However, to date computer forensic work has concentrated on file recovery and decryption, so the contributions in this area still lie in the future.

Many computer pundits, as well as some security experts, have proposed that computer viruses are the result of the fact that currently popular desktop operating systems have only nominal security provisions. They further suggest that viruses will disappear as security functions are added to operating systems. This thesis ignores the facts — well established by Cohen's research and subsequently confirmed — that viruses use the most basic of computer functions, and a perfect defense against viruses is impossible. This is not to say that an increase in security measures by operating system vendors could not reduce the risk of viruses — the current danger could be drastically reduced with relatively minor modifications to system functions.

It is going too far to say (as some have) that the very existence of viral programs, and the fact that both viral strains and the numbers of individual infections are growing, means that computers are finished. At the

present time, the general public is not well informed about the virus threat, so more copies of viral programs are being produced than are being destroyed.

Indeed, no less an authority than Fred Cohen has championed the idea that viral programs can be used to great effect. An application using a viral form can improve performance in the same way that computer hardware benefits from parallel processors. It is, however, unlikely that viral programs can operate effectively and usefully in the current computer environment without substantial protective measures built into them.

Malware Types

Viruses are not the only form of malicious software. Other forms include worms, Trojans, zombies, logic bombs, and hoaxes. Each of these has its own characteristics, and we will discuss each of the forms below. Some forms of malware combine characteristics of more than one class, and it can be difficult to draw hard and fast distinctions with regard to individual examples or entities; but it is important to keep the specific attributes in mind.

It should be noted that we are increasingly seeing convergence in malware. Viruses and Trojans are used to spread and plant RATs, and RATs are used to install zombies. In some cases, hoax virus warnings are used to spread viruses. Virus and Trojan payloads may contain logic bombs and data diddlers.

Viruses

A computer virus is a program written with functions and intent to copy and disperse itself without the knowledge and cooperation of the owner or user of the computer. All researchers have not yet agreed on a final definition. A common definition is "a program that modifies other programs to contain a possibly altered version of itself." This definition is generally attributed to Fred Cohen from his seminal research in the middle 1980s, although Dr. Cohen's actual definition is in mathematical form. (The term *computer virus* was first defined by Dr. Cohen in his graduate thesis in 1984. Cohen credits a suggestion from his advisor, Leonard Adelman [of RSA fame], for the use of the term.) Another possible definition is an entity that uses the resources of the host (system or computer) to reproduce itself and spread without informed operator action.

Cohen's definition is specific to programs that attach themselves to other programs as their vector of infection. However, common usage now holds viruses to consist of a set of coded instructions that are designed to attach to an object capable of containing the material, without knowledgeable user intervention. This object may be an e-mail message, program file, document, floppy disk, CD-ROM, short message system (SMS) message on cellular telephones, or any similar information medium.

A virus is defined by its ability to reproduce and spread. A virus is not merely anything that goes wrong with a computer, and a virus is not simply another name for malware. Trojan horse programs and logic bombs do not reproduce themselves.

A worm, which is sometimes seen as a specialized type of virus, is currently distinguished from a virus because a virus generally requires an action on the part of the users to trigger or aid reproduction and spread. (There will be more on this distinction in the section on worms later in this chapter.) The actions on the part of the users are generally common functions, and the users generally do not realize the danger of their actions or the fact that they are assisting the virus.

The only requirement that defines a program as a virus is that it reproduces. There is no necessity that viruses carry a payload, although a number of viruses do. In many cases (in most cases of successful viruses), the payload is limited to some kind of message. A deliberately damaging payload, such as erasure of the disk or system files, usually restricts the ability of the virus to spread because the virus uses the resources of the host system. In some cases, a virus may carry a logic bomb or time bomb that triggers a damaging payload on a certain date or under a specific, often delayed, condition.

Because a virus spreads and uses the resources of the host, it affords the kind of power to software that parallel processors provide to hardware. Therefore, some have theorized that viral programs could be used for beneficial purposes, similar to the experiments in distributed processing that are testing the limits of cryptographic strength. (Various types of network management functions and updating of system software are seen as candidates.) However, the fact that viruses change systems and applications is seen as problematic in its own right. Many viruses that carry no overtly damaging payload still create problems with systems. A number of virus and worm programs have been written with the obvious intent of proving that viruses could carry a

useful payload, and some have even had a payload that could be said to enhance security. Unfortunately, all such viruses have created serious problems. The difficulties of controlling viral programs have been addressed in theory, but the solutions are also known to have faults and loopholes. (One of the definitive papers on this topic is available at http://www.frisk.is/~bontchev/papers/goodvir.html.)

Types of Viruses

There are a number of functionally different types of viruses, such as a file infector, boot-sector infector (BSI), system infector, e-mail virus, multipartite, macro virus, or script virus. These terms do not necessarily indicate a strict division. A file infector may also be a system infector. A script virus that infects other script files may be considered to be a file infector — although this type of activity, while theoretically possible, is unusual in practice. There are also difficulties in drawing a hard distinction between macro and script viruses.

Later in this chapter there is a section enumerating specific examples of malware, where the viruses noted in the next few paragraphs are discussed in detail. We have tried to include examples that explain and expand on these different types.

File Infectors

A file infector infects program (object) files. System infectors that infect operating system program files (such as command.com in DOS) are also file infectors. File infectors can attach to the front of the object file (prependers), attach to the back of the file and create a jump at the front of the file to the virus code (appenders), or overwrite the file or portions of it (overwriters). A classic is Jerusalem. A bug in early versions caused it to add itself over and over again to files, making the increase in file length detectable. (This has given rise to the persistent myth that it is a characteristic of a virus that it will fill up all disk space eventually; by far, the majority of file infectors add minimally to file lengths.)

Boot-Sector Infectors

Boot-sector infectors (BSIs) attach to or replace the master boot record, system boot record, or other boot records and blocks on physical disks. (The structure of these blocks varies, but the first physical sector on a disk generally has some special significance in most operating systems and usually it is read and executed at some point in the boot process.) BSIs usually copy the existing boot sector to another unused sector, and then copy themselves into the physical first sector, ending with a call to the original programming. Examples are Brain, Stoned, and Michelangelo.

System Infectors

System infector is a somewhat vague term. The phrase is often used to indicate viruses that infect operating system files, or boot sectors, in such a way that the virus is called at boot time and has or may have preemptive control over some functions of the operating system. (The Lehigh virus infected only COMMAND.COM on MS-DOS machines.) In other usage, a system infector modifies other system structures, such as the linking pointers in directory tables or the MS Windows system registry, in order to be called first when programs are invoked on the host computer. An example of directory table linking is the DIR virus family. Many e-mail viruses target the registry: MTX and Magistr can be very difficult to eradicate.

Companion Virus

Some viral programs do not physically touch the target file at all. One method is quite simple and may take advantage of precedence in the system. In MS-DOS, for example, when a command is given, the system checks first for internal commands, then .com, .exe, and .bat files in that order. The .exe files can be infected by writing a .com file in the same directory with the same filename. This type of virus is most commonly known as a companion virus, although the term *spawning virus* is also used.

E-Mail Virus

An e-mail virus specifically, rather than accidentally, uses the e-mail system to spread. While virus-infected files may be accidentally sent as e-mail attachments, e-mail viruses are aware of e-mail system functions. They generally target a specific type of e-mail system (Microsoft's Outlook is the most commonly used), harvest e-mail addresses from various sources, and may append copies of themselves to all e-mails sent or generate e-mail messages containing copies of themselves as attachments. Some e-mail viruses may monitor all network traffic and follow up legitimate messages with messages that they generate. Most e-mail viruses are technically

considered to be worms because they do not often infect other program files on the target computer, but this is not a hard and fast distinction. There are known examples of e-mail viruses that are file infectors, macro viruses, script viruses, and worms. Melissa, LoveLetter, Hybris, and SirCam are all widespread current examples, and the CHRISTMA exec is an older example of the same type of activity.

E-mail viruses have made something of a change to the epidemiology of viruses. Traditionally, viruses took many months to spread but stayed around for many years in the computing environment. Many e-mail viruses have become "fast burners" that can spread around the world, infecting hundreds of thousands or even millions of machines within hours. However, once characteristic indicators of these viruses become known, they die off almost immediately when users stop running the attachments.

Multipartite

Originally the term *multipartite* was used to indicate a virus that was able to infect both boot sectors and program files. (This ability is the origin of the alternate term *dual infector.*) Current usage tends to mean a virus that can infect more than one type of object or that infects or reproduces in more than one way. Examples of traditional multipartites are Telefonica, One Half, and Junkie, but these programs have not been very successful.

Macro Virus

A macro virus uses macro programming of an application such as a word processor. (Most known macro viruses use Visual Basic for Applications in Microsoft Word; some are able to cross between applications and functions in, for example, a PowerPoint presentation and a Word document, but this ability is rare.) Macro viruses infect data files and tend to remain resident in the application by infecting a configuration template such as MS Word's Normal.dot. Although macro viruses infect data files, they are not generally considered to be file infectors; a distinction is generally made between program and data files. Macro viruses can operate across hardware or operating system platforms as long as the required application platform is present. (For example, many MS Word macro viruses can operate on both the Windows and Macintosh versions of MS Word.) Examples are Concept and CAP. Melissa is also a macro virus, in addition to being an e-mail virus; it mailed itself around as an infected document.

Script Virus

Script viruses are generally differentiated from macro viruses in that script viruses are usually stand-alone files that can be executed by an interpreter, such as Microsoft's Windows Script Host (.vbs files). A script virus file can be seen as a data file in that it is generally a simple text file, but it usually does not contain other data and generally has some indicator (such as the .vbs extension) that it is executable. LoveLetter is a script virus.

Virus Examples and Encyclopedias

Examples of recent viruses, in very brief form, can be found at http://www.osborne.com/virus_alert/. More comprehensive information on a much greater number of viruses can be found at the various virus encyclopedia sites. Two of the best are:

1. F-Secure: http://www.f-secure.com/v-descs/
2. Sophos: http://www.sophos.com/virusinfo/analyses/

Others can be found at:

- http://www.viruslist.com/eng/viruslist.asp http://www.symantec. com/avcenter/vinfodb.html
- http://www.antivirus.com/vinfo/virusencyclo/ http://www.cai.com/ virusinfo/encyclopedia/
- http://antivirus.about.com/library/blency.htm http://vil.mcafee.com/
- http://www.pandasoftware.com/library/defaulte.htm

Virus Structure

In considering computer viruses, three structural parts are considered important: the replication or infection mechanism, the trigger, and the payload.

Infection Mechanism

The first and only necessary part of the structure is the infection mechanism. This is the code that allows the virus to reproduce and thus to be a virus. The infection mechanism has a number of parts to it.

The first function is to search for, or detect, an appropriate object to infect. The search may be active, as in the case of some file infectors that take directory listings in order to find appropriate programs of appropriate sizes; or it may be passive, as in the case of macro viruses that infect every document as it is saved. There may be some additional decisions taken once an object is found. Some viruses may actually try to slow the rate of infection to avoid detection. Most will check to see if the object has already been infected.

The next action will be the infection itself. This may entail the writing of a new section of code to the boot sector, the addition of code to a program file, the addition of macro code to the Microsoft Word Normal.dot file, the sending of a file attachment to harvested e-mail addresses, or a number of other operations. There are additional subfunctions at this step as well, such as the movement of the original boot sector to a new location or the addition of jump codes in an infected program file to point to the virus code. There may also be changes to system files, to try to ensure that the virus will be run every time the computer is turned on. This can be considered the insertion portion of the virus.

At the time of infection, a number of steps may be taken to try to keep the virus safe from detection. The original file creation date may be conserved and used to reset the directory listing to avoid a change in date. The virus may have its form changed in some kind of polymorphism. The active portion of the virus may take charge of certain system interrupts in order to make false reports when someone tries to look for a change to the system. There may also be certain prompts or alerts generated in an attempt to make any odd behavior noticed by the user appear to be part of a normal, or at least innocent, computer error.

Trigger

The second major component of a virus is the payload trigger. The virus may look for a specific number of infections, a certain date or time, or a particular piece of text. A section of code does not have to contain either a trigger or a payload to be defined as a virus.

Payload

If a virus does have a trigger, then it usually has a payload. The payload can be pretty much anything, from a simple one-time message, to a complicated display, to reformatting the hard disk. However, the bigger the payload, the more likely it is that the virus will get noticed. A virus carrying a very destructive payload will also eradicate itself when it wipes out its target. Therefore, while you may have seen lists of payload symptoms to watch for, such as text messages, ambulances running across the screen, letters falling down, and such, checking for these payloads is not a very good way to keep free of viruses. The successful ones keep quiet.

Stealth

A great many people misunderstand the term *stealth*. It is often misused as the name of a specific virus. At other times, there are references to stealth viruses as if they were a class such as file infectors or macro viruses. In fact, stealth refers to technologies that can be used by any virus and by other forms of malware as well, and often it is used as a reference to all forms of anti-detection technology. Stealth is used inconsistently even within the virus research community.

A specific usage of the term refers to an activity also known as *tunneling*, which (in opposition to the usage in virtual private networks) describes the act of tracing interrupt links and system calls in order to intercept calls to read the disk, or performing other measures that could be used to determine that an infection exists. A virus using this form of stealth would intercept a call to display information about the file (such as its size) and return only information suitable to the uninfected object. This type of stealth was present in one of the earliest MS-DOS viruses, Brain. (If you gave commands on an infected system to display the contents of the boot sector, you would see the original boot sector and not the infected one.)

Polymorphism (literally many forms) refers to a number of techniques that attempt to change the code string on each generation of a virus. These vary from using modules that can be rearranged to encrypting the virus code itself, leaving only a stub of code that can decrypt the body of the virus program when invoked. Polymorphism is sometimes also known as self-encryption or self-garbling, but these terms are imprecise and not recommended. Examples of viruses using polymorphism are Whale and Tremor. Many polymorphic viruses

use standard mutation engines such as MtE. These pieces of code actually aid detection because they have a known signature.

A number of viruses also demonstrate some form of active detection avoidance, which may range from disabling on-access scanners in memory to deletion of anti-virus and other security software (Zonealarm is a favorite target) from the disk.

Worms

A worm reproduces and spreads, like a virus and unlike other forms of malware. Worms are distinct from viruses, although they may have similar results. Most simply, a worm may be thought of as a virus with the capacity to propagate independently of user action. That is, they do not rely on (usually) human-initiated transfer of data between systems for propagation; instead, they spread across networks of their own accord, primarily by exploiting known vulnerabilities in common software.

Originally, the distinction was made that worms used networks and communications links to spread and that a worm, unlike a virus, did not directly attach to an executable file. In early research into computer viruses, the terms *worm* and *virus* tended to be used synonymously because it was felt that the technical distinction was unimportant to most users. The technical origin of the term *worm program* matched that of modern distributed processing experiments: a program with segments working on different computers, all communicating over a network (Shoch and Hupp, 1982).

In fact, the use and origin of the term *worm* in relation to computer programs is rather cloudy. There are references in early computing to *wormhole* programs that escaped from their assigned partitions. The wormhole reference may note the similarity that random damage bears to the characteristic patterns of holes in worm-eaten wood, or relate to the supposition in science fiction stories that wormholes may carry you to random places. The Shoch and Hupp article contains a quote from John Brunner's novel, *The Shockwave Rider,* that describes a *tapeworm* program, although this entity bears little resemblance to modern malware.

The first worm to garner significant attention was the Internet Worm of 1988. Recently, many of the most prolific virus infections have not been strictly viruses, but have used a combination of viral and worm techniques to spread more rapidly and effectively. LoveLetter was an example of this convergence of reproductive technologies. While infected e-mail attachments were perhaps the most widely publicized vectors of infection, LoveLetter also spread by actively scanning attached network drives and infecting a variety of common file types. This convergence of technologies will be an increasing problem in the future. Code Red and a number of Linux programs (such as Lion) are modern examples of worms. (Nimda is an example of a worm, but it also spreads in a number of other ways; so it could be considered to be an e-mail virus and multipartite as well.)

Hoaxes

Hoax virus warnings or alerts have an odd double relation to viruses. First, hoaxes are usually warnings about "new" viruses — new viruses that do not, of course, exist. Second, hoaxes generally carry a directive to the user to forward the warning to all addresses available to them. Thus, these descendants of chain letters form a kind of self-perpetuating spam.

Hoaxes use an odd kind of social engineering, relying on the naturally gregarious nature of people and their desire to communicate a matter of urgency and importance, using the human ambition to be the first to provide important new information.

Hoaxes do, however, have common characteristics that can be used to determine whether their warnings are valid:

- Hoaxes generally ask the reader to forward the message.
- Hoaxes make reference to false authorities such as Microsoft, AOL, IBM, and the FCC (none of which issue virus alerts), or to completely false entities.
- Hoaxes do not give specific information about the individual or office responsible for analyzing the virus or issuing the alert.
- Hoaxes generally state that the new virus is unknown to authorities or researchers.
- Hoaxes often state that there is no means of detecting or removing the virus.

- Many of the original hoax warnings stated only that you should not open a message with a certain phrase in the subject line. (The warning, of course, usually contained that phrase in the subject line. Subject-line filtering is known to be a very poor method of detecting malware.)
- Hoaxes often state that the virus does tremendous damage and is incredibly virulent.
- Hoax warnings very often contain A LOT OF CAPITAL-LETTER SHOUTING AND EXCLAMATION MARKS!!!!!!!!!!
- Hoaxes often contain technical-sounding nonsense (technobabble) such as references to nonexistent technologies like "nth complexity binary loops."

It is wisest in the current environment to doubt all virus warnings, unless they come from a known and historically accurate source such as a vendor with a proven record of providing reliable and accurate virus alert information, or preferably an independent researcher or group. It is best to check *any* warnings received against known virus encyclopedia sites. It is best to check more than one such site — in the initial phases of a fast burner attack, some sites may not have had time to analyze samples to their own satisfaction; and the better sites will not post unverified information.

A recent example of a hoax, referring to SULFNBK.EXE, got a number of people to clear this legitimate utility off their machines. The origin was likely the fact that the Magistr virus targets Windows system software, and someone with an infection did not realize that the file is actually present on all Windows 98 systems.

Trojans

Trojans, or Trojan horse programs, are the largest class of malware aside from viruses. However, use of the term is subject to much confusion, particularly in relation to computer viruses.

A Trojan is a program that pretends to do one thing while performing another, unwanted action. The extent of the pretense may vary greatly. Many of the early PC Trojans merely used the filename and a description on a bulletin board. Log-in Trojans, popular among university student mainframe users, mimicked the screen display and the prompts of the normal log-in program and could, in fact, pass the username and password along to the valid log-in program at the same time as they stole the user data. Some Trojans may contain actual code that does what it is supposed to be doing while performing additional nasty acts.

Some data security writers consider that a virus is simply a specific example of the class of Trojan horse programs. There is some validity to this usage because a virus is an unknown quantity that is hidden and transmitted along with a legitimate disk or program, and any program can be turned into a Trojan by infecting it with a virus. However, the term *virus* more properly refers to the added, infectious code rather than the virus/target combination. Therefore, the term *Trojan* refers to a deliberately misleading or modified program that does not reproduce itself.

An additional confusion with viruses involves Trojan horse programs that may be spread by e-mail. In years past, a Trojan program had to be posted on an electronic bulletin board system or a file archive site. Because of the static posting, a malicious program would soon be identified and eliminated. More recently, Trojan programs have been distributed by mass e-mail campaigns, by posting on Usenet newsgroup discussion groups, or through automated distribution agents (bots) on Internet relay chat (IRC) channels. Because source identification in these communications channels can be easily hidden, Trojan programs can be redistributed in a number of disguises, and specific identification of a malicious program has become much more difficult.

Social Engineering

A major aspect of Trojan design is the social engineering component. Trojan programs are advertised (in some sense) as having a positive component. The term *positive* can be in dispute, because a great many Trojans promise pornography or access to pornography — and this still seems to be depressingly effective. However, other promises can be made as well. A recent e-mail virus, in generating its messages, carried a list of a huge variety of subject lines, promising pornography, humor, virus information, an anti-virus program, and information about abuse of the recipient's e-mail account. Sometimes, the message is simply vague and relies on curiosity.

It is instructive to examine some classic social engineering techniques. Formalizing the problem makes it easier to move toward effective solutions and making use of realistic, pragmatic policies. Effective implemen-

tation of such policies, however good they are, is not possible without a considered user education program and cooperation from management.

Social engineering really is nothing more than a fancy name for the type of fraud and confidence games that have existed since snakes started selling apples. Security types tend to prefer a more academic-sounding definition, such as the use of nontechnical means to circumvent security policies and procedures. Social engineering can range from simple lying (such as a false description of the function of a file), to bullying and intimidation (in order to pressure a low-level employee into disclosing information), to association with a trusted source (such as the username from an infected machine), to dumpster diving (to find potentially valuable information people have carelessly discarded), to shoulder-surfing (to find out personal identification numbers and passwords).

Remote-Access Trojans (RATs)

Remote-access Trojans are programs designed to be installed, usually remotely, after systems are installed and working (and not in development, as is the case with logic bombs and backdoors). Their authors would generally like to have the programs referred to as *remote administration tools* so as to convey a sense of legitimacy.

All networking software can, in a sense, be considered remote access tools — we have file transfer sites and clients, World Wide Web servers and browsers, and terminal emulation software that allows a microcomputer user to log on to a distant computer and use it as if on-site. The RATs considered to be in the malware camp tend to fall somewhere in the middle of the spectrum. Once a client such as Back Orifice, Netbus, Bionet, or SubSeven is installed on the target computer, the controlling computer is able to obtain information about the target computer. The master computer will be able to download files from, and upload files to, the target. The control computer will also be able to submit commands to the victim, which basically allows the distant operator to do pretty much anything to the prey. One other function is quite important: all of this activity goes on without any alert given to the owner or operator of the targeted computer.

When a RAT program has been run on a computer, it will install itself in such a way as to be active every time the computer is started subsequent to the installation. Information is sent back to the controlling computer (sometimes via an anonymous channel such as IRC) noting that the system is active. The user of the command computer is now able to explore the target, escalate access to other resources, and install other software, such as DDoS zombies, if so desired.

Once more, it should be noted that remote access tools are not viral. When the software is active, the master computer can submit commands to have the installation program sent on, via network transfer or e-mail, to other machines. In addition, RATs can be installed as a payload from a virus or Trojan.

Rootkits, containing software that can subvert or replace normal operating system software, have been around for some time. RATs differ from rootkits in that a working account must be either subverted or created on the target computer in order to use a rootkit. RATs, once installed by a virus or Trojan, do not require access to an account.

DDoS Zombies

DDoS (distributed denial-of-service) is a modified denial-of-service (DoS) attack. Denial-of-service attacks do not attempt to destroy or corrupt data; rather, they attempt to use up a computing resource to the point where normal work cannot proceed. The structure of a DDoS attack requires a master computer to control the attack, a target of the attack, and a number of computers in the middle that the master computer uses to generate the attack. These computers in between the master and the target are variously called agents or clients, but are usually referred to as running zombie programs.

Again, note that DDoS programs are not viral, but checking for zombie software protects not only your system but also prevents attacks on others. It is, however, still in your best interest to ensure that no zombie programs are active. If your computers are used to launch an assault on some other system, you could be liable for damages.

The efficacy of this platform was demonstrated in early 2000 when a couple of teenagers successfully paralyzed various prominent online players in quick succession, including Yahoo!, Amazon, and eBay.

Logic Bombs

Logic bombs are software modules set up to run in a quiescent state — but to monitor for a specific condition or set of conditions and to activate their payloads under those conditions. A logic bomb is generally implanted in or coded as part of an application under development or maintenance. Unlike a RAT or Trojan, it is difficult to implant a logic bomb after the fact. There are numerous examples of this type of activity, usually based upon actions taken by a programmer to deprive a company of needed resources in the event of employment termination.

A Trojan or a virus may contain a logic bomb as part of the payload. A logic bomb involves no reproduction and no social engineering.

A persistent legend in regard to logic bombs involves what is known as the *salami scam*. According to the story, this involves siphoning off small amounts of money (in some versions, fractions of a cent) and crediting it to the account of the programmer over a very large number of transactions. Despite the fact that these stories appear in a number of computer security texts, this author has a standing challenge to anyone to come up with a documented case of such a scam. Over a period of eight years, the closest anyone has come is a story about a fast-food clerk who diddled the display on a drive-through window and collected an extra dime or quarter from most customers.

Pranks

Pranks are very much a part of the computer culture — so much so that you can now buy commercially produced joke packages that allow you to perform "Stupid Mac (or PC, or Windows) Tricks." There are countless pranks available as shareware. Some make the computer appear to insult the user; some use sound effects or voices; some use special visual effects. A fairly common thread running through most pranks is that the computer is, in some way, nonfunctional. Many pretend to have detected some kind of fault in the computer (and some pretend to rectify such faults, of course making things worse). One entry in the virus field is Parascan, the paranoid scanner. It pretends to find large numbers of infected files, although it does not actually check for any infections.

Generally speaking, pranks that create some kind of announcement are not malware; viruses that generate a screen or audio display are actually quite rare. The distinction between jokes and Trojans is harder to make, but pranks are intended for amusement. Joke programs may, of course, result in a denial of service if people find the prank message frightening.

One specific type of joke is the *Easter egg*, a function hidden in a program and generally accessible only by some arcane sequence of commands. These may be seen as harmless but they do consume resources, even if only disk space, and also make the task of ensuring program integrity much more difficult.

Malware and Virus Examples

It is all very well to provide academic information about the definitions and functions of different types of malware. It may be difficult to see how all this works in practice. In addition, it is often easier for people to understand how a particular technology works when presented with an actual example.

Here, then, are specific examples of viruses and malware. All of these have been seen and been successful, to an extent, in the wild (outside of research situations). One benefit of looking at malware in this way is that the discussion is removed from the realms of the possible to the actual. For example, there has been a great deal of debate over the years about whether a virus can do damage to hardware. Theoretically, it is possible. In actual fact, it has not happened.

Viruses do dominate in this section, and there are reasons for this. First, there are more examples of viruses to draw from. This chapter is not meant to, and cannot, be an encyclopedia of the tens of thousands of viruses; but it is important to give examples of the major classes of viruses. Second, the possible range of Trojans is really only limited by what can be done with software. People generally do not feel that there is much difference between a Trojan that reformats the hard disk and one that only erases all the files. From the user's perspective, the effect is pretty much the same; and the defensive measure that should have been taken (do not run unknown software) is also identical.

This material not only provides technical details but also looks at the history and some social factors involved. Social engineering is often involved in malware, and it is instructive to look at strategies that have been successful to determine policies that will protect users.

Boot-Sector Infectors

Brain

Technically, the Brain family (Pakistani, Pakistani Brain, Lahore, and Ashar), although old and seldom seen anymore, raises a number of interesting points. Brain itself was the first known PC virus, aside from those written by Fred Cohen for his thesis. Unlike Cohen's file viruses, however, Brain is a boot-sector infector.

Brain has been described as the first stealth virus. A request to view the boot sector of an infected disk on an infected system will result in a display of the original (pre-infection) boot sector. However, the volume label of an infected diskette is set to "©Brain," "©Ashar," or "Y.C.1.E.R.P," depending on the variant. Every time a directory listing is requested, the volume label is displayed; so it is difficult to understand why the virus uses stealth in dealing with the display of the boot sector. In one of the most common Brain versions, there is unencrypted text giving the name, address, and telephone numbers of Brain Computer Services in Pakistan. The virus is copyrighted by "Ashar and Ashars" or "Brain & Amjads."

Brain is not intentionally or routinely destructive, and it is possible that the virus was intended to publicize the company. This was the earliest known PC virus, and viruses did not inspire the same revulsion that they tend to do today. Even some time after the later and more destructive viruses, Lehigh and Jerusalem, viruses were still seen as possibly neutral or even in some way beneficial. It may be that the author saw a self-reproducing program that lost, at most, 3 kb of disk space as simply a novelty. In a way, such a virus as this would not be dissimilar to the easter egg applet pranks used by programmers working for major application publishers to express their individuality.

Fridrik Skulason, whose F-Prot has provided the engine for a number of anti-virus products over the years, exhaustively analyzed the later Ohio and Den Zuk versions of the Brain virus.

The Ohio (Den Zuk 1) and Den Zuk (Venezuelan, Search) variants contain some of the same code as Brain in order to prevent overlaying by Brain. However, Ohio and Den Zuk identify and overwrite Brain infections with themselves. They can be described as single-shot anti-virus utilities targeting the Brain virus (at the expense, however, of causing the Ohio and Den Zuk infections). Skulason also found that the Den Zuk version would overwrite an Ohio infection. (This seeking activity gives rise to one of Den Zuk's aliases: *Search*.)

It was also suspected that denzuko might have referred to the Search for Brain infections. Extensive searches for the meaning of the words *den zuk* and *denzuko* in a number of languages, as an attempt to find clues to the identity of the virus author, turned up closely related words meaning *sugar* and *knife* as well as *search*. However, these turned out to be quite beside the point.

There is text in both Den Zuk and Ohio that suggests they were written by the same author. Ohio contains an address in Indonesia (and none in Ohio — the name derives from Ohio State University, where it was first identified). Both contain a ham-radio license number issued in Indonesia. Both contain the same programming bug. The FAT (file allocation table) and data areas are overwritten if a floppy disk with a higher capacity than 360 kb is infected. Den Zuk is a more sophisticated exercise in programming. Skulason concluded, therefore, that Ohio was in fact an earlier version of Den Zuk.

The virus' author, apparently a college student in Indonesia, confirmed Skulason's hypotheses. There had been attempts to trace the virus' origins through the words *denzuk* and *denzuko*. In fact, Den Zuko turned out to be the author's nickname, derived from John Travolta's character in the movie *Grease*.

Stoned (and Variants)

The Stoned virus seems to have been written by a high school student in New Zealand — hence its other main alias, *New Zealand*. All evidence suggests that he wrote the virus only for study and that he took precautions against the release of the code. These safeguards proved insufficient, as it turned out. It is reported that his brother stole a copy and decided to infect the machines of friends.

The original version of Stoned is said to have been restricted to infecting floppy disks. The current, most common version of Stoned, however, infects all disks. It is an example of a second class of boot-sector-infecting viral programs in that it places itself in the master boot record or partition boot record of a hard disk instead

of the boot sector (as it does on floppy disks). In common with most BSIs, Stoned moves the original sector into a new location on the disk. On hard disks and double-density floppies, this movement is not usually a problem. On high-density floppies, however, system information can be overwritten, resulting in loss of data. One version of Stoned reportedly does not infect 3½-inch diskettes; this version may well be the template for Michelangelo, which does not infect 720 kb disks either.

Michelangelo, Monkey, and Other Stoned Variants

Stoned has spawned a large number of mutations ranging from minor variations in the spelling of the payload message to the functionally different Empire, Monkey, and No-Int variations.

Michelangelo is generally believed by researchers to have been built on or mutated from the Stoned virus. The similarity of the replication mechanism, down to the inclusion of the same bugs, puts this theory beyond any reasonable doubt. Any successful virus is likely to be copied. Michelangelo is unusual only in the extent to which the payload has been modified.

Roger Riordan reported and named the virus in Australia in February of 1991. He suspected that the virus had entered the victim company on disks of software from Taiwan, but this hypothesis remains unproven. The date indicates the existence of the virus prior to March 6, 1991. This demonstrates that the virus can survive its own deletion of disk information every March 6, even though it destroys itself along with the system tracks of disks overwritten on that date. This resiliency is not really surprising — few computer users understand that boot viruses can, in principle, infect any disk from any other disk, regardless of whether the disk is bootable, contains any program files, or contains any files at all.

Riordan determined that March 6 was the trigger date. It is often assumed from the name of the virus that it was intended to trigger on March 6 because that is the birthday of Michelangelo Buonarotti, the Renaissance artist, sculptor, and engineer. However, there is no text in the body of the virus, no reference to Michelangelo, and no evidence of any sort that the author of the virus was aware of the significance of that particular date. The name is simply the one that Riordan chose to give it, based on the fact that a friend with the same birth date knew that it was also Michelangelo's.

By the beginning of 1992, commercial production software was being shipped on Michelangelo-infected floppies, and at least one company was shipping infected PC systems. It has been suggested that, by the end of February of that year, when the general public was becoming aware of the problem, the number of infected floppies out in the field may have been in the millions. Fortunately, most infected machines were checked and diagnosed before March 6 of that year.

The replication mechanism of Michelangelo is basically that of Stoned. It replaces the original boot sector on a floppy disk with a copy of itself. The virus moves the original boot sector to sector 3 (for 360 kb diskettes) or 14 (for 1.2 or 1.44 MB diskettes), and the virus contains a "loader" that points to this location. After the virus loads itself into memory, the original boot sector is run; to the user, the boot process appears to proceed normally. On hard disks, the original partition sector is moved to (0,0,7).

Michelangelo is no stealth virus. Examination of the boot blocks shows a clear difference between a valid sector and the one that is infected. (The absence of the normal system messages should be a tip-off — Michelangelo contains no text whatsoever.) In addition, Michelangelo reserves itself 2 kb at the top of memory. A simple run of DOS' CHKDSK utility will show total conventional memory on the system; and if a 640 kb machine shows 655,360 bytes, then the computer is not infected with Michelangelo. (If the number is less, there may still be reasons other than a virus; and if the number is 655,360, that does not, of course, prove that no virus is present or active.)

Removal is a simple matter of placing the original sector back where it belongs, thus wiping out the infection. This can be done with sector-editing utilities, or even with DEBUG, although it would normally be easier and safer to simply use an anti-virus utility. There have been many cases where a computer has been infected with both Stoned and Michelangelo. In this situation, the boot sector cannot be recovered, because both Stoned and Michelangelo use the same "landing zone" for the original sector; and the infection by the second virus overwrites the original boot sector with the contents of the first virus.

When an infected computer boots up, Michelangelo checks the date via Interrupt 1Ah. If the date is March 6, the virus then overwrites the first several cylinders of the disk with the contents of memory. Interrupt 1Ah is not usually available on the earliest PCs and XTs (with some exceptions). However, the disk that is overwritten is the disk from which the system is booting; a hard disk can be saved simply by booting from a floppy. Also, the damage is triggered only at boot time, although this is not altogether a positive. The fact that the damage

occurs during the boot process means that the payload, like the infection mechanism, is no respecter of operating systems — it and can and does trash non-DOS operating systems such as UNIX.

A number of suggestions were made in early 1992 as to how to deal with Michelangelo without using anti-virus software. Because so many anti-viral programs — commercial, shareware, and freeware — identified the virus, it seems odd that people were so desperate to avoid this obvious step of using a scanning program to find the virus.

Some people recommended backing up data, which is always a good idea. And, given that Michelangelo is a boot-sector infector, it would not be stored on a tape backup. However, diskettes are a natural target for BSIs. Today, diskettes are much less favored for major backup purposes. Zip disks, tapes, and other high-capacity writeable media are cheap and highly available. At that time, however, many popular backup programs used proprietary non-DOS disk formats for reasons of speed and additional storage. These, if infected by Michelangelo, would become unusable.

Changing the computer clock was also a popular suggestion. Because Michelangelo was set to go off on March 6, theoretically you could just set the computer clock to make sure that it never reached March 6. However, many people did not understand the difference between the MS-DOS clock and the system clock read by Interrupt 1Ah. The MS-DOS DATE command did not always alter the system clock. Network-connected machines often have time-server functions so that the date would be reset to conform to the network. The year 1992 was a leap year, and many clocks did not deal with it properly. Thus, for many computers, March 6 came on Thursday, not Friday. This suggestion comes up time and again for dealing with viruses with a known trigger date (CIH, for example) and was trotted out again for dealing with the Y2K bug.

An even sillier suggestion was to test for Michelangelo by setting the date to March 6 and then rebooting the computer. This strategy became known as *Michelangelo roulette*. One vendor actually reported an incident where a customer switched on a machine on the fatal morning and when the machine promptly died, the customer switched on the other machines in the office to see if the same thing happened. It did.

Many people suggested a modem avoidance strategy. Such a strategy is, of course, no defense worth mentioning against any boot-sector virus. Neither the master/partition boot record nor the boot sector is an identifiable, transferable file. Neither can be transmitted by an everyday user as a file over a modem or Ethernet connection, although an infected disk can be transferred over a network connection as a binary image. Although dropper programs are theoretically possible, they are rarely used as a means of disseminating a virus through unsuspecting users. The danger of getting a Michelangelo infection from a BBS was, therefore, so small that for all practical purposes it did not exist. Warning against bulletin boards, or, more recently, Web sites, merely proscribes a major source of advice and utility software.

Unlike the Columbus Day/Datacrime hypefest of 1989, the epidemic of Michelangelo in the spring of 1992 had its basis in fact. Vendors were making unsubstantiated claims for the numbers of infections, which, in retrospect, turned out to have been surprisingly accurate. More importantly, the research community as a whole was seeing large numbers of infections. The public was seeing them as well. No fewer than 15 companies shipped commercial products that turned out to be infected with the Michelangelo virus.

Two producers of commercial anti-viral programs released crippled freeware versions of their scanners. The programs did briefly mention that they checked only for Michelangelo, but certainly gave users the impression that they were checking the whole system. Happily, the trend over recent years has been to produce small, single-shot programs for dealing urgently with high-profile viruses rather than a crippled version of a free package. Even this approach has its drawbacks — recently, there was an instance where a Hybris infection was almost overlooked because the freeware program used could detect only a single variant. Oddly, it was a later variant than the one actually found on the machine in question. It seems that the vendor assumed that anyone using it would already have updates of their product for the previous versions. Because the vendor in question was also responsible for one of the free Michelangelo scanners, perhaps the average vendor's sense of ethical responsibility has not been raised as far as one could hope.

Because of the media attention, a number of checks were made that would not have been done otherwise. Hundreds and even thousands of copies of Michelangelo were found within single institutions. Because many copies had been found and removed, the number of hits on March 6 was not spectacular. Predictably, perhaps, media reports on March 6 started to dismiss the Michelangelo scare as another over-hyped rumor, completely missing the reality that millions of machines had possibly been struck.

File Infectors

Lehigh

Lehigh only infects COMMAND.COM, the operating system interpreter program in MS-DOS, which rather restricts its capacity to spread because bootable floppy disks became much less common with the rise of hard disk drives and almost completely vanished with the advent of Windows. (The target of infection means that Lehigh can be considered a system infector under the more recent definition of that term.) Nevertheless, it received a great deal of publicity and had a direct impact on the anti-virus scene. Ken van Wyk, who was working at Lehigh at the time (and went on to join CERT [Carnegie Mellon University's Computer Emergency Response Team]), set up the VIRUS-L/comp.virus mailing list and newsgroup. Unfortunately, VIRUS-L seems to have disappeared, but it was for a number of years the primary source of accurate virus information and, in large measure, responsible for ensuring that the anti-virus research community did in fact become a community.

The Lehigh virus overwrote the slack space at the end of the COMMAND.COM file. This meant that the virus did not increase the size of infected files. A later report of a 555-byte increase in file size was due to confusion over the size of the overwriting code. When an infected COMMAND.COM was run (usually upon booting from an infected disk), the virus stayed resident in memory. When any access was made to another disk, via the TYPE, COPY, DIR, or other normal DOS commands, COMMAND.COM files would be infected. The virus kept a counter of infections: after four infections, the virus would overwrite the boot and FAT areas of disks with bytes copied from BIOS.

Lehigh (the virus, not the campus) is remarkably stealth free. The primary defense of the virus was that, at the time, no one would have been looking for it. The virus altered the date stamp of infected COMMAND.COM files. If attempting an infection on a write-protected disk, the virus would not trap the Write Protect Error message. This message is a serious giveaway if seen as a result of typing dir — generating the directory listing should not require writing to the diskette (unless output is redirected).

The virus was limited in its target population to those disks that had a COMMAND.COM file and, more particularly, those that contained a full operating system. The virus was also self-limiting in that it would destroy itself once activated and would activate after only four reproductions. The Lehigh virus never did spread beyond the campus in that initial attack. Although it is found in a number of private virus collections and may be released into the wild from time to time, the virus has no real chance of spreading.

Jerusalem

In terms of the number of infections (copies or reproductions) that a virus produces, boot-sector viral programs long held an advantage in the microcomputer environment. Among file-infecting viral programs, however, the Jerusalem virus was the clear winner. It has another claim to fame as well: it almost certainly has the largest number of variants of any virus program known to date, at least in its class of parasitic file infectors.

Initially known to some as the Israeli virus, the version reported by Y. Radai in early 1988 (also sometimes referred to as *1813* or *Jerusalem-B*) was the most commonly encountered version. Although it was the first to be widely disseminated and was the first to be discovered and publicized, analysis suggests that it was the outcome of previous viral experiments.

A few things are common to pretty much all of the Jerusalem family. They usually infect both .com and .exe files. When an infected file is executed, the virus "goes TSR (terminate and stay resident)" — that is, it installs itself into memory. Thus, it remains active even after the originally infected program is terminated. The .exe programs executed after the program goes resident are infected by appending the virus code to the end of the file. Prepending code infects .com files. Most variants carry some kind of date logic-bomb payload, often triggered on Friday the 13th. Sometimes the logic bomb is simply a message; often, it deletes programs as they are accessed.

Although Jerusalem tends to work well with .com files, the differing structure of .exe files has presented Jerusalem with a number of problems. Early versions of Jerusalem, not content with one infection, will reinfect .exe files again and again so that they continually grow in size. This growth renders pointless the attempt at stealth that the programmer built in when he ensured that the file creation date was conserved and unchanged in an infected file. Also, .exe programs that use internal loaders or overlay files tend to be infected in the wrong place and have portions of the original program overwritten. Although the virus was reported to slow down systems that were infected, it seems to have been the continual growth of .exe files that led to the detection of the virus.

The great number of variants has contributed to severe naming and identification problems. Because a number of the variants are based on the same code, the signatures for one variant often match another — thus generating even more naming confusion. This confusion is not unique to the Jerusalem family, of course, and is an ongoing concern in the anti-virus research community, while systems administrators are growing increasingly forceful and vociferous in their demands for a unified nomenclature.

An early infection was found in an office belonging to the Israeli defense forces, giving rise to the occasional synonym IDF. This synonym was actually problematical because it was more often used as a synonym for the unrelated Frodo virus.

The common Jerusalem payload of file deletion on Friday the 13th (yet another alias) begged a question as to why the logic bomb had not gone off on Friday, November 13, 1987. Subsequent analysis has shown that the virus will activate the payload only if the year is not 1987. The next following Friday the 13th was May 13th, 1988. Because the last day that Palestine existed as a nation was May 13, 1948, it was felt that the virus might have been an act of political terrorism. This supposition led to another alias, the PLO virus. However, Israel celebrates its holidays according to the Jewish calendar (no surprises there), and the independence celebrations were slated for three weeks before May 13, 1988. These facts, and the links between Jerusalem and the sURIV family, suggest that there is no intentional political link. It is almost certain that the Jerusalem virus is, in fact, two viral programs combined. The two viruses, and others in the development family, have been found.

sURIV 1.01 is a .com-file infector — .com is the easier file structure and therefore the easier program to infect. sURIV 2 is an .exe-only infector and has considerably longer and more complex code. sURIV 3 infects both types of program files and has considerable duplication of code; it is, in fact, simply the first two versions concatenated together.

Although the code in the sURIV programs and the 1813 version of Jerusalem is not absolutely identical, all the same features are duplicated. The payload date for sURIV is April 1, and the year has to be later than 1988. Although this seems to suggest that sURIV is a descendant of Jerusalem, the reverse is probably the case. Certainly the code is less sophisticated in the sURIV variants.

More recent viruses that infect Windows portable executable (PE) files, as well as Lindose/Winux, which infects both Windows PE and Linux ELF files, are considered to be an advance in virus technology. In fact, they are simply following in the footsteps of Jerusalem.

The Jerusalem virus was immensely successful as a template for variants. The code is reasonably straightforward and, for those with some familiarity with assembly programming, an excellent primer for writing viral programs affecting both .com and .exe files. It has a number of annoying bugs, however. It can misinfect some .exe files. It can conflict with Novell NetWare, which requires the use of Interrupt 21h subfunctions that are also used by the virus. One of the *Sunday* variants is supposed to delete files on the seventh day of the week. The author did not realize that computers start counting from zero and that Sunday is actually the *zero* day of the week — so there is no seventh day, and the file deletions never actually happen.

E-mail Viruses

CHRISTMA Exec

CHRISTMA exec, the Christmas Tree virus/worm, sometimes referred to as the BITNET chain letter, was probably the first major malware attack across networks. It was launched on December 9, 1987, and spread widely on BITNET, EARN, and IBM's internal network (VNet). It has a number of claims to a small place in history. It was written, unusually, in REXX. It was mainframe-hosted (on VM/CMS systems) rather than microcomputer-hosted — quaint as that distinction sounds today, when the humblest PC can run UNIX.

CHRISTMA presented itself as a chain letter inviting the recipient to execute its code. This involvement of the user led to the definition of the first e-mail virus rather than a worm. When it was executed, the program drew a Christmas tree and mailed a copy of itself to everyone in the account holder's equivalent to an address book, the user files NAMES and NETLOG. Conceptually, there is a direct line of succession from this worm to the social engineering worm/Trojan hybrids of today.

W97M/Melissa (Mailissa)

She came from alt.sex.

Now, as the old joke goes, that I have your attention …

In this instance, however, the lure of sex was certainly employed to launch the virus into the wild. The source of the infestation of the Melissa Word macro virus (more formally identified as some variation on W97M/Melissa) was a posting on the Usenet newsgroup alt.sex. The message had a Word document attached. (More details of macro viruses are given later in regard to the Concept virus.) The posting suggested that the document contained account names and passwords for Web sites carrying salacious material. As one might expect in such a newsgroup, a number of people read the document. It carried a macro that used the functions of Microsoft Word and the Microsoft Outlook mailer program to reproduce and spread itself — rather successfully, as it turns out. Melissa is not the fastest-burning e-mail-aware malware to date, but it certainly held the record for awhile.

Many mail programs, in the name of convenience, are becoming more automated. Much of this automation has focused on running attached files, or scripting functions included in HTML-formatted messages, without requiring the intervention of the victim.

To be susceptible to the effects of Melissa, a victim needed to be running Microsoft Word 97 or later, or Microsoft Outlook 98 or later. It was also necessary to receive an infected file and read it into Word without disabling the macro capability. However, all of these conditions are normal for many users. Receiving infected documents has never been a problem, from WM/Concept onward. Melissa increased the likelihood that any given individual user would eventually receive an infected document by the sheer weight of numbers. However, by judicious social engineering, the virus also increased the chances of persuading a victim to open an infected document. Many mail programs will now detect the type of a file from its extension and start the appropriate program automatically.

On execution, the virus first checks to see whether an infectable version of Word is running. If so, Melissa reduces the level of security on Word so that no future warnings of macro content are displayed. Under Word 2000, the virus blocks access to the menu item that allows you to raise your security level and sets your macro virus detection to the lowest level — that is, to none. Restoring the security level requires the deletion of the Normal.dot file and the consequent loss of legitimate macros and customizations.

The virus checks for the registry key HKEY_CURRENT_USER\Software\Microsoft\Office\Melissa?\ with a value of "… by Kwyjibo." (The "Kwyjibo" entry seems to be a reference to the "Bart the Genius" episode of *The Simpsons* television cartoon program wherein Bart Simpson used this word to win a Scrabble match.) If that key is not found, the macro starts up Outlook and sends itself as an attachment to the top 50 names in each of your address lists. Most people have only one (the default is Contacts); but if there is more than one, then Outlook will send more than 50 copies of the message. Outlook also sorts address lists so that other mailing lists are at the top of the list. In addition, under a Microsoft Exchange Server, the macro can send copies out to the global address lists on the server. Therefore, a single infected machine may distribute far more than 50 copies of the message/virus in the next "hop."

Like most macro viruses, Melissa worked by infecting the global template and infecting all documents thereafter. Each document created or reviewed was infected when closed. Each infected document activated the macro when the file was opened. Avoiding Outlook did not offer protection from the virus; it only meant that the 50 copies would not be sent out automatically. If Microsoft Word was used, but not Outlook, the machine would still be infected, and infected documents could still be sent out in the normal course of operations.

The virus cannot invoke the mass-mailer dispersal mechanism on Macintosh systems, but it can be stored and resent from Macs.

As with any Word macro virus, the source code travels with the infection and it was very easy to create modifications to Melissa. Many Melissa variants with different subjects and messages started to appear shortly after the original virus appeared. The first similar Excel macro virus was called *Papa*, although this and its progeny never had the same global impact as Melissa. In fact, the source code was published more widely than usual in newgroups, on the Web, and elsewhere.

In one distressing instance, a major security organization issued a flash advisory including a range of information of varying quality and relevance. Unfortunately, it also included the entire source code, trivially modified so that it would not run without some tweaking.

As with many more recent mail-borne nuisances, a number of fixes such as sendmail and procmail recipes for mail servers and mail filtering systems were devised very quickly. However, these fixes were often not fully tested or debugged. One version would trap most of the warning messages about Melissa. Mail filters can, of course, become problems. In the mailing of the author's initial report on the virus, it bounced from one system because of an automated filter that interpreted the message as a hoax virus warning.

W95.Hybris

The Hybris worm started to make its mark in late September 2000. It is disseminated by an e-mail message that is often but by no means always sent from hahaha@sexyfun.net. This address is forged to make it harder to trace the infected source. However, the sexyfun.net domain was later set up and used as a Hybris information resource. The worm may sometimes check the language settings of the host computer and select a "story" relating to Snow White and the Seven Dwarfs in English, French, Spanish, or Portuguese, used as message text to accompany the copy of the worm when it is mailed out, and implying that the attached file is a kind of pornographic screen saver.

When the worm attachment is executed, the WSOCK32.DLL file is modified or replaced so that it can track e-mail and other Internet traffic. When the worm detects an e-mail address, it sends infected e-mail to that address. It also connects to alt.comp.virus and uploads encrypted plug-in modules to the group. If it finds newer plug-ins, the worm downloads them for its own use. For several months, alt.comp.virus was almost unusable because of the sheer numbers of plug-ins clogging the group.

Worms

The Morris Worm (Internet Worm)

In the autumn of 1988, most people were blissfully ignorant of viruses and the Internet. However, I recall that Virus-L had been established and was very active. At that time the list was still an exploder re-mailer, rather than a digest; but postings were coming out pretty much on a daily basis. However, there were no postings on November 3 or on November 4. It was not until November 5, actually, that I found out why.

The Morris Worm did not actually bring the Internet in general and e-mail in particular to the proverbial grinding halt. It was able to run and propagate only on machines running specific versions of the UNIX operating system on specific hardware platforms. However, given that the machines that are connected to the Internet also comprise the transport mechanism for the Internet, a "minority group" of server-class machines, thus affected, degraded the performance of the Net as a whole. Indeed, it can be argued that, despite the greater volumes of mail generated by Melissa and LoveLetter and the tendency of some types of mail servers to achieve meltdown when faced with the consequent traffic, the Internet as a whole has proved to be somewhat more resilient in recent years.

During the 1988 mailstorm, a sufficient number of machines had been affected to impair e-mail and distribution-list mailings. Some mail was lost, either by mailers that could not handle the large volumes that backed up or by mail queues being dumped in an effort to disinfect systems. Most mail was substantially delayed. In some cases, mail would have been rerouted via a possibly less efficient path after a certain time. In other cases, backbone machines, affected by the problem, were simply much slower at processing mail. In still others, mail-routing software would crash or be taken out of service, with a consequent delay in mail delivery. Ironically, electronic mail was the primary means of communication of the various parties attempting to deal with the trouble. By Sunday, November 6, mail was flowing, distribution lists and electronic periodicals were running, and the news was getting around. However, an enormous volume of traffic was given over to one topic — the Internet worm.

In many ways, the Internet worm is the story of data security in miniature. The worm used trusted links, password cracking, security holes in standard programs, standard and default operations, and, of course, the power of viral replication.

"Big Iron" mainframes and other multi-user server systems are generally designed to run constantly, and they execute various types of programs and procedures in the absence of operator intervention. Many hundreds of functions and processes may be running all the time, expressly designed to neither require nor report to an operator. Some processes cooperate with each other; others run independently. In the UNIX world, such small utility programs are referred to as daemons, after the supposedly subordinate entities that take over mundane tasks and extend the power of the wizard, or skilled operator. Many of these utility programs deal with the communications between systems. Mail, in the network sense, covers much more than the delivery of text messages between users. Network mail between systems may deal with file transfers, the routing of information for reaching remote systems, or even upgrades and patches to system software.

When the Internet worm was well established on a machine, it would try to infect another. On many systems this attempt was all too easy — computers on the Internet were meant to generate activity on each other, and some had no protection in terms of the type of access and activity allowed.

The finger program is one that allows a user to obtain information about another user. The server program *fingerd* is the daemon that listens for calls from the finger client. The version of fingerd common at the time of the Internet Worm had a minor problem: it did not check how much information it was given. It would take as much as it could hold and leave the rest to overflow. The *rest*, unfortunately, could be used to start a process on the computer, and this process was used as part of the attack. This kind of buffer overflow attack continues to be very common, taking advantage of similar weaknesses in a wide range of applications and utilities.

The sendmail program is the engine of most mail-oriented processes on UNIX systems connected to the Internet. In principle, it should only allow data received from another system to be passed to a user address. However, there is a debug mode that allows commands to be passed to the system. Some versions of UNIX were shipped with the debug mode enabled by default. Even worse, the debug mode was often enabled during installation of sendmail for testing and then never turned off.

When the worm accessed a system, it was fed with the main program from the previously infected site. Two programs were used, one for each infected platform. If neither program could work, the Worm would erase itself. If the new host was suitable, the worm looked for further hosts and connections.

The program also tried to break into user accounts on the infected machine. It used standard password-cracking techniques such as simple variations on the name of the account and the user. It carried a dictionary of words likely to be used as passwords, and would also look for a dictionary on the new machine and attempt to use that as well. If an account were cracked, the worm would look for accounts that this user had on other computers, using standard UNIX tools.

The worm did include a means of checking for copies already running on a target computer. However, it took some time to terminate the program; and the worm regularly produced copies of itself that would not respond to the request for termination at all. The copies of the Worm did destroy themselves — having first made a new copy. In this way, the identifying process ID number would continually change.

The worm was not intentionally destructive. However, the mere presence of the program had implications for the infected systems and for those associated with them. The multiple copies of the program that ran on the host machines had a serious impact on other processes. Also, communications links and processes were used to propagate the worm rather than to support the legitimate work for which they were intended.

Linux Worms

By spring 2001, a number of examples of Linux malware had been seen. Interestingly, while the Windows viruses generally followed the CHRISTMA exec style of having users run the scripts and programs, the new Linux worms were similar to the Internet/Morris/UNIX worms in that they rely primarily on bugs in automatic networking software.

Ramen

The Ramen worm makes use of security vulnerabilities in default installations of Red Hat Linux 6.2 and 7.0 using specific versions of the wu-ftp, rpc.statd, and LPRng programs. The worm defaces Web servers by replacing index.html and scans for other vulnerable systems. It does this initially by opening an ftp connection and checking the remote system's ftp banner message. If the system is vulnerable, the worm uses one of the exploitable services to create a working directory; it then downloads a copy of itself from the local (attacking) system.

Compromised systems send out e-mail messages to two Hotmail and Yahoo! accounts, and ftp services are disabled. Ramen's SYN scanning may disrupt network services if multicasting is supported by the network.

Lion

Lion uses a buffer overflow vulnerability in the bind program to spread. When it infects, Lion sends a copy of output from the ifconfig commands etc/passwd and /etc/shadow to an e-mail address in the china.com domain. Next, the worm adds an entry to etc/inetd.conf and restarts inetd. This entry would allow Lion to download components from a (now closed) Web server located in China. Subsequently, Lion scans random class B subnets in much the same way as Ramen, looking for vulnerable hosts. The worm may install a rootkit onto infected systems. This backdoor disables the syslogd daemon and adds a Trojanized SSH (secure shell) daemon.

The worm replaces several system executables with modified versions. The /bin/in.telnetd and /bin/mjy files provide additional backdoor functionality and attempt to conceal the rootkit's presence by hiding files and processes.

Adore (Linux/Red)

Adore is a Linux worm similar to Linux/Ramen and Linux/Lion. It uses vulnerabilities in wu-ftpd, bind, lpd, and RPC.statd that enable an intruder to gain root access and run unauthorized code. The worm attempts to send IP configuration data, information about running processes, and copies of /etc/hosts and /etc/shadow to e-mail addresses in China. It also scans for class B IP addresses.

Adore drops a script called 0anacron into the /etc/cron.daily directory so that the script runs as a daily cron job. The cron utility executes scheduled tasks at predetermined times. This script removes the worm from the infected host. A modified version of the system program /bin/ps that conceals the presence of the worm's processes replaces the original.

Code Red

Code Red uses a known vulnerability to target Microsoft IIS (Internet Information Server) Web servers. Despite the fact that a patch for the loophole had been available for five months prior to the release of Code Red, the worm managed to infect 350,000 servers within nine to thirteen hours.

When a host gets infected, it starts to scan for other hosts to infect. It probes random IP addresses, but the code is flawed by always using the same seed for the random number generator. Therefore, each infected server starts probing the same addresses that have been done before. (It was this bug that allowed the establishment of such a precise count for the number of infections.)

During a certain period of time the worm only spreads, but then it initiates a denial-of-service (DoS) attack against www1.whitehouse.gov. However, because this particular machine name was only an overflow server, it was taken offline prior to the attack and no disruptions resulted.

The worm changed the front page of an infected server to display certain text and a background color of red — hence the name of the worm.

Code Red definitely became a media virus. Although it infected at least 350,000 machines within hours, it had probably almost exhausted its target population by that time. Despite this, the FBI held a rather ill-informed press conference to warn of the worm.

Code Red seems to have spawned quite a family, each variant improving slightly on the random probing mechanism. In fact, there is considerable evidence that Nimda is a descendent of Code Red.

Nimda variants all use a number of means to spread. Like Code Red, Nimda searches random IP addresses for unpatched Microsoft IIS machines. Nimda will also alter Web pages in order to download and install itself on computers browsing an infected Web site using a known exploit in Microsoft Internet Explorer's handling of Java. Nimda will also mail itself as a file attachment and will install itself on any computer on which the file attachment is executed. Nimda is normally e-mailed in HTML format and may install automatically when viewed using a known exploit in Microsoft Internet Explorer. Nimda will also create e-mail and news files on network shares and will install itself if these files are opened.

Macro Viruses

Concept

WM/Concept was by no means the first macro virus. HyperCard viruses were already commonplace in the Macintosh arena when WM/Concept appeared, and a number of anti-virus researchers had explored WordBasic and other malware-friendly macro environments (notably Lotus 1–2–3) long before the virus appeared in 1995.

However, WM/Concept was the first macro virus to be publicly described as such, and certainly the most successful in terms of spread. For awhile, it was easily the most widely found virus in the world. Oddly enough, however, its appearance was greeted with disbelief in some quarters. After all, a Word file is usually thought of as data rather than a program file.

People cling to the belief that, because executable files run programs and data files contain data, there is a clear-cut distinction between the two file types. In fact, this has never been true; and the von Neumann architecture makes such a differentiation impossible. What may be perceived as a data file may be, in reality,

a program. A PostScript file is, in fact, a program read and acted upon by a PostScript interpreter program. A printer normally executes this program, but a program such as GhostView can also interpret a PostScript file and print it to the screen on the host computer.

The first in-the-wild examples specifically targeted Microsoft Word v6.0, but code for viruses infecting Excel and Ami Pro also appeared very quickly. All versions of Word for Windows and Word 6 and later for the Macintosh include a sophisticated macro language (WordBasic in older versions, and later Visual Basic for Applications, or VBA). Such applications are capable of all the functions normally associated with a high-level programming language such as Basic. In fact, macro languages used by Windows applications are based on Microsoft's Visual Basic.

Concept spread far and (for its time) rapidly. It got something of a boost when two companies accidentally shipped it in infected documents on CD-ROM. The first instance was a Microsoft CD called MicroSoft Windows '95 Software Compatibility Test. The CD was shipped to a number of large original equipment manufacturing (OEM) companies in the summer of 1995 as a means of checking compatibility with Windows 95, which was due for imminent release. However, the CD contained a document called oemltr.doc, which was infected with Concept. A few months later, Microsoft UK distributed the virus on another CD, The Microsoft Office 95 and Windows 95 Business Guide, in a document called helpdesk.doc.

Concept was fairly obvious and could be forestalled and even fixed (with patience) without the aid of anti-virus software. When a Concept-infected file was opened, a message box appeared containing the number 1 and an OK button. You could also detect the virus' presence by checking the Tools/Macros submenu for the presence of macros.

A WM/Concept.A infection is characterized by the presence of the macros AAAZFS, AAAZAO, AutoOpen, Payload, and FileSaveAs. Any document might legitimately use AutoOpen or FileSaveAs. However, macros with the names Payload, AAAZFS, and AAAZAO are something of a giveaway. The macros are not encrypted, so it is easy to spot the virus. On the other hand, this lack of encryption also made it easy to modify the code. Virus writers learned almost immediately to conceal the internals of their macros by implementing them as execute-only macros, which cannot be edited or easily viewed.

Although Concept.A has a payload macro, it has no actual payload. Famously, it contains the string "That's enough to prove my point," which explains the name Concept (as in "proof of concept").

Concept.A was a fairly harmless affair, as viruses go: it tampered with Word 6's global template (normally Normal.dot, or Normal on a Macintosh) so that files were saved as templates and ran the infective AutoOpen macro. This gave Mac users an additional advantage in that template files on the Mac have a different icon to document files. As long as the virus infected only template files, this icon was a frequently found heads-up to Mac users that they might have a virus problem. However, in later versions of Word, the distinction between documents and templates is less absolute; and that particular heuristic has become less viable.

In a sense, the main importance of Concept was that the code could be altered very quickly to incorporate a destructive payload, alternative infection techniques, and evasion of the first attempts at detecting it. This virus has been described as the first cross-platform virus in that it works on any platform. However, this description is not altogether accurate: it only infected systems running Word 6 or Word 95, although versions are known that can infect Word 97 and later.

Script Viruses

VBS/LoveLetter

LoveLetter first hit the nets on May 3, 2000. It spread rapidly, arguably faster than Melissa had the previous year.

The original LoveLetter came in an e-mail with a subject line of "I LOVE YOU." The message consisted of a short note urging you to read the attached love letter. The attachment filename, LOVE-LETTER-FOR-YOU.TXT.vbs, was a fairly obvious piece of social engineering. The .TXT bit was supposed to make people think that the attachment was a text file and thus safe to read. At that point, many people had no idea what the .vbs extension signified; and in any case they might have been unaware that, if a filename has a double extension, only the last filename extension has any special significance. Putting vbs in lower case was likely meant to play down the extension's significance. However Windows, like DOS before it, is not case sensitive when it comes to filenames, and the .vbs extension indicates a Visual Basic script.

If Windows 98, Windows 2000, Internet Explorer 5, Outlook 5, or a few other programs are installed, then so is Windows Script Host (WSH); and there is a file association binding the .vbs extension to WSCRIPT.EXE. In

that case, double-clicking on the file attachment is enough to start WSH and interpret the contents of the "love letter."

The infection mechanism included the installation of some files in the Windows and System directories. These files were simply copies of the original .vbs file — in one case keeping the name of LOVE-LETTER-FOR- YOU.TXT.vbs, but in other cases renaming files to fool people into thinking that they were part of the system (MSKERNEL32.vbs and WIN32DLL.vbs).

The virus made changes to the registry so that these files would be run when the computer started up. Today, many organizations routinely quarantine or bounce files with a .vbs extension (especially a double extension) at the mail gateway.

LoveLetter infects files with the extensions .vbs, .vbe, .js, .jse, .css, .wsh, .sct, .hta, .jpg, .jpeg, .mp2, and .mp3. The infection routine searches local drives and all mounted network drives, so shared directories can be an additional source of infection. The routines overwrite most of these files with a copy of the script (that is, the original file is not preserved anywhere, although the new file has a different name) and change the filenames from (for example) picture.jpg to picture.jpg.vbs. In some cases, the virus simply deletes the original file. MPEGs, however, are not overwritten. The original file, say song.mp3, is marked as hidden; and a new file, song.mp3.vbs, is created with a copy of the virus. The .vbs extension must, of course, be added for the virus to be effective.

Once the virus has copied itself all over a host machine, it starts to spread to other machines. If Outlook is present, the virus will use any addresses associated with the mail program to send copies of itself (but once only). As with Melissa, this means that when a copy of LoveLetter was received, it would appear to come from someone known to the recipient. In addition, the program tries to make a connection to IRC, using the mIRC chat program, and spread that way. The Love Bug (as it was also known) creates another copy of the file, LOVE-LETTER-FOR-YOU.HTM, in the Windows System directory, and then sends that copy to any user who joins the IRC channel while the session is active.

When a system is infected, the worm attempts to download a Trojan application from a Web site in the Philippines by changing the start-up URL in Internet Explorer. The file, named WIN-BUGSFIX.exe, will try to collect various password files and e-mail them to an address in the Philippines. If the file is executed, the Trojan also creates a hidden window called BAROK and remains resident and active in memory. However, this site was probably overloaded in the early hours of the LoveLetter infection, and was quickly taken down.

A very large number of LoveLetter "cleaners" were made available. Interestingly, most of them were Visual Basic scripts themselves. Unfortunately, at least two variants of the virus pretended to be disinfecting tools and did more damage than the original virus.

Because the virus is an unencrypted script file, it carries its own source code with it. This means that variants started appearing within hours. Over a dozen were reported in the weekend after the virus first struck, and many more have been observed since. One of the more successful of these thanked the recipient for the order of a Mother's Day gift and claimed that the recipient's credit card had been charged $326.92 as per the attached invoice. Obviously, this ruse relied on people being too angry to think about how anybody could charge their credit card when they had not given the number to a vendor. Certainly, the variants showed a certain amount of innovation in the field of social engineering, if not in the actual code. One derivative targets UNIX systems using shell scripts but uses a very similar mechanism.

There have been estimates of damage stemming from LoveLetter in the billions of dollars. It is very difficult to justify those figures. Certainly, a number of e-mail systems were clogged, including those of some very large organizations. Many administrators shut down mail entirely rather than turn to filtering. In addition, the resetting of registry entries is likely to be somewhat time-consuming.

Text in the virus includes the string "Manila, Philippines." There are also the two Philippine e-mail addresses in the code and the Web site's URL. However, all charges against the individual long thought to have been the culprit were eventually dropped by the Manila Department of Justice.

Combinations and Convergence

BadTrans

BadTrans is a Win32 e-mail virus with backdoor functionality. It was found in the wild in April 2001.

The worm uses MAPI functions to access and respond to unread messages. The Trojan component is a version of Hooker, a password-stealing Trojan, and mails system information to ld8dl1@mailandnews.com.

On infection, the worm copies itself to \Windows as inetd.exe and drops the hkk32.exe Trojan, also to the Windows folder. The password stealer is executed and then moved to the system directory as kern32.exe, dropping a keystroke logging DLL (dynamic link library) at the same time. The worm modifies win.ini (Windows 9x) or the registry (Windows NT/2000) so that it is run on start-up.

When infective mail is sent, the worm randomly selects the attachment filename from a number of variants, some of them obviously influenced by previous worms. The subject field in worm messages is the same as in the original message, preceded by "Re:" so that it appears to be a response to that message. The message body also looks like a reply to the original message, which the body quotes in full. At the end of the quote, there is a single line, "Take a look to the attachment." The worm attempts to avoid answering the same mail twice or answering its own messages from other victim systems by adding two spaces to the end of the subject field and not responding to any mail with such a subject line. This mechanism is unreliable, however, because mail servers are likely to discard trailing spaces. In this event, an infective message received on a machine already infected will generate a response from the local instance of the worm, thus initiating a potential loop. A loop can also be initiated if the worm is unable to mark answered messages, as can happen with certain mail clients. Such a loop could result in a mail server meltdown.

Hoaxes

Good Times

Good Times is probably the most famous of all false alerts, and it was certainly the earliest that got widely distributed. Some controversy persists over the identity of the originators of the message, but it is possible that it was a sincere, if misguided, attempt to warn others. The hoax probably started in early December of 1994. In 1995, the FCC variant of the hoax began circulating.

It seems most likely that the Good Times alert was started by a group or an individual who had seen a computer failure without understanding the cause and associated it with an e-mail message that had Good Times in the subject line. (In fact, there are indications that the message started out on the AOL system, and it is known that there are bugs in AOL's mail software that can cause the program to hang.) The announcement states that there was a message identified by the title of Good Times that, when read, would crash a computer. The message was said to be a virus, although there was nothing viral about that sort of activity (even if it were possible).

At the time of the original Good Times message, e-mail was almost universally text based. Suffice it to say that the possibility of a straightforward text message carrying a virus in an infective form is remote. The fact that the warning contained almost no details at all should have been an indication that the message was not quite right. There was no information on how to detect, avoid, or get rid of the virus, except for its warning not to read messages with Good Times in the subject line. (The irony of the fact that many of the warnings contained these words seems to have escaped most people.)

Pathetically (and far from uniquely), a member of the vx community (Virus eXchange, those who write and spread viruses) produced a Good Times virus. Like the virus named after the older Proto-T hoax, the *real* Good Times was an uninteresting specimen, having nothing in common with the original alert. It is generally known as GT-Spoof by the anti-virus community, and was hardly ever found in the field.

Hoaxes are depressingly common and tend to have a number of common characteristics. Here is an annotated version of one:

> There is a virus out now sent to people via e-mail … it is called the A.I.D.S. VIRUS.

There are, in fact, an AIDS virus or two, but they are simple file-infecting viruses that have nothing to do with e-mail.

> It will destroy your memory, sound card and speakers, drive.

Many hoaxes suggest this kind of massive damage, including damage to hardware.

> And it will infect your mouse or pointing device as well as your keyboards.

Hoaxes also tend to state that the new virus has extreme forms of infection. In this case, it would be impossible for a virus to infect pointing devices or keyboards unless those pieces of equipment have memory and processing capabilities. None of these hoax warnings really detail how the virus is supposed to pass itself along.

> Making what you type not able to register on the screen. It self-terminates only after it eats 5MB of hard drive space

More damage claims …

> It will come via e-mail called "OPEN: VERY COOL! :)"

And the virus has no other characteristics, according to this alert.

> PASS IT ON QUICKLY & TO AS MANY PEOPLE AS POSSIBLE!!

This, of course, is the real virus, getting the user to spread it.

Trojan

The AIDS Trojan Extortion Scam

In the fall of 1989, approximately 10,000 copies of an "AIDS Information" package were sent out from a company calling itself PC Cyborg. Some were received at medical establishments; a number were received at other types of businesses. The packages appeared to have been professionally produced. Accompanying letters usually referred to them as sample or review copies. However, the packages also contained a very interesting license agreement:

> In case of breach of license, PC Cyborg Corporation reserves the right to use program mechanisms to ensure termination of the use of these programs. These program mechanisms will adversely affect other program applications on microcomputers. You are hereby advised of the most serious consequences of your failure to abide by the terms of this license agreement.

Further in the license is the sentence: "Warning: Do not use these programs unless you are prepared to pay for them."

The disks contained an installation program and a very simple AIDS information file and risk assessment. The installation program appeared to only copy the AIDS program onto the target hard disk, but in reality did much more. A hidden directory was created with a nonprinting character name, and a hidden program file with a nonprinting character in the name was installed. The autoexec.bat file was renamed and replaced with one that called the hidden program and then the original autoexec. The hidden program kept track of the number of times the computer was rebooted and, after a certain number, encrypted the hard disk. The user was then presented with an invoice and a demand to pay the license fee in return for the encryption key. Two major versions were found to have been shipped. One, which waited for 90 reboots, was thought to be the real attempt; an earlier version, which encrypted after one reboot, alerted authorities and was thought to be an error on the part of the principals of PC Cyborg.

The Panamanian address for PC Cyborg, thought by some to be a fake, turned out to be real. Four principals were identified, as well as an American accomplice who seems to have had plans to send 200,000 copies to American firms if the European test worked. The trial of the American, Joseph Popp, was suspended in Britain because his bizarre behavior in court was seen as an indication that he was unfit to plead. An Italian court, however, found him guilty and sentenced him in *absentia*.

RATs

BackOrifice

BackOrifice was developed by the hacker group Cult of the Dead Cow in order to take control of Windows 95 and 98 systems. A newer version, BackOrifice2000 (BO2K), was created in July 1999 in order to control Windows NT and 2000 systems.

As with all RATs, the BackOrifice2000 backdoor has two major parts: client and server. The server part needs to be installed on a computer system to gain access to it with the client part. The client part connects to the server part via network and is used to perform a wide variety of actions on the remote system. The client part has a dialogue interface that eases the process of hacking the remote computer.

In the same package there is also a configuration utility that is used to configure the server part of BO2K. It asks the user to specify networking type (TCP or UDP); port number (1-65535); connection encryption type, simple (XOR) or strong (3DES); and password for encryption that will be the password for the server access also.

The configuration utility allows flexibility in configuring the server part. It can add or remove plug-ins (DLLs) from the server application, configure file transfer properties, TCP and UDP settings, built-in plug-in activation, encryption key, and start-up properties. The start-up properties setup allows configuration of automatic installation to systems, server file names, process names, process visibility, and also NT-specific properties (NT service and host process names).

The file from which the server part started can be deleted. After that, BO2K will be active in memory each time Windows starts and will provide access to the infected system for hackers who have the client part and the correct password.

The active server part can hide its process or prevent its task from being killed from the Task Manager (on NT). The backdoor uses a smart trick on NT by constantly changing its PID (process ID) and by creating the additional process of itself that will keep the backdoor alive even if one of the processes is killed. The server part adds a random (but large) number of spaces and 'e' at the end of its name; thus, the server part file cannot be deleted from Windows (invalid or long name error). The server file can be only deleted from DOS.

DDoS Zombies

Trin00

Also known as Trinoo, this is a distributed tool used to launch coordinated UDP flood DoS attacks from many sources.

An intruder can actually communicate with a Trinoo master computer by communicating with port 27665, typically by Telnet. The master sends UDP packets to daemons on destination port 27444. The daemons send UDP flood packets to the target.

The binary for the trinoo daemon contains IP addresses for one or more trinoo master systems. When the trinoo daemon is executed, the daemon announces its availability by sending a UDP packet containing the string HELLO to its programmed trinoo master IP addresses on port 31335.

The trinoo master stores a list of known daemons. The trinoo master can be instructed to send a broadcast request to all known daemons to confirm availability. Daemons receiving the broadcast respond to the master with a UDP packet containing the string PONG.

The trinoo master then communicates with the daemons, giving instructions to attack one or more IP addresses for a specified period of time.

All communications to the master on port 27665/tcp require a password, with a default of *betaalmostdone*, which is stored in the daemon binary in encrypted form. All UDP communications with the daemon on port 27444 require the UDP packet to contain the string l44 (that is a lower-case letter L, not a one).

Tribe Flood Network (TFN)

TFN, much like Trinoo, is a distributed tool used to launch coordinated DoS attacks from many sources against one or more targets. In addition to the ability to generate UDP flood attacks, a TFN network can generate TCP SYN flood, ICMP echo request flood, and ICMP directed broadcast (e.g., smurf) DoS attacks. TFN has the capability to generate packets with spoofed source IP addresses.

A TFN master is executed from the command line to send commands to TFN daemons. The master communicates with the daemons using ICMP echo reply packets with 16-bit binary values embedded in the ID field and any arguments embedded in the data portion of the packet. The binary values, which are definable at compile time, represent the various instructions sent between TFN masters and daemons.

Detection/Protection

When dealing with malware, the only safe assumption is that everything that can go wrong will go wrong, and at the worst possible time. Until the need for this level of security diligence is accepted as the general business case, the information security practitioner will have an uphill battle.

However, training and explicit policies can greatly reduce the danger to users. Some guidelines that can really help in the current environment are:

- Do not double-click on attachments.
- When sending attachments, provide a clear and specific description as to the content of the attachment.
- Do not blindly use Microsoft products as a company standard.
- Disable Windows Script Host. Disable ActiveX. Disable VBScript. Disable JavaScript. Do not send HTML-formatted e-mail.
- Use more than one scanner, and scan everything.

Whether these guidelines are acceptable in a specific environment is a business decision based on the level of acceptable risk. But remember: whether risks are evaluated, and whether policies are explicitly developed, every environment has a set of policies (some are explicit, while some are implicit), and every business accepts risk. The distinction is that some companies are aware of the risks that they choose to accept.

Protective tools in the malware area are generally limited to anti-virus software. To this day there are three major types, first discussed by Fred Cohen in his research. These types are known as signature scanning, activity monitoring, and change detection. These basic types of detection systems can be compared with the common intrusion detection system (IDS) types, although the correspondence is not exact. A scanner is like a signature-based IDS. An activity monitor is like a rule-based IDS or an anomaly-based IDS. A change detection system is like a statistical-based IDS. These software types will be examined very briefly.

Scanners

Scanners examine files, boot sectors, and memory for evidence of viral infection, and many may detect other forms of malware. They generally look for viral signatures, sections of program code that are known to be in specific malicious programs but not in most other programs. Because of this, scanning software will generally detect only known malware and must be updated regularly. (Currently, with fast-burner e-mail viruses, this may mean daily or even hourly.) Some scanning software has resident versions that check each file as it is run.

Scanners have generally been the most popular form of anti-viral software, probably because they make a specific identification. In fact, scanners offer somewhat weak protection because they require regular updating. Scanner identification of a virus may not always be dependable: a number of scanner products have been known to identify viruses based on common families rather than definitive signatures. In addition, scanners fail "open;" if a scanner does not trigger an alert when scanning an object, that does not mean the object is not infected or that it is not another type of malware.

It is currently popular to install anti-viral software as a part of filtering firewalls or proxy servers. It should be noted that such automatic scanning is demonstrably less effective than manual scanning and subject to a number of failure conditions.

Activity Monitors

An activity monitor performs a task very similar to an automated form of traditional auditing; it watches for suspicious activity. It may, for example, check for any calls to format a disk or attempts to alter or delete a program file while a program other than the operating system is in control. It may be more sophisticated, and check for any program that performs "direct" activities with hardware, without using the standard system calls.

Activity monitors represent some of the oldest examples of anti-viral software, and are usually effective against more than just viruses. Generally speaking, such programs followed in the footsteps of the earlier anti-Trojan software, such as BOMBSQAD and WORMCHEK in the MS-DOS arena, which used the same "check what the program tries to do" approach. This tactic can be startlingly effective, particularly given the fact that so much malware is slavishly derivative and tends to use the same functions over and over again.

It is, however, very hard to tell the difference between a word processor updating a file and a virus infecting a file. Activity monitoring programs may be more trouble than they are worth because they can continually ask for confirmation of valid activities. The annals of computer virus research are littered with suggestions for virus-proof computers and systems that basically all boil down to the same thing: if the operations that a computer can perform are restricted, viral programs can be eliminated. Unfortunately, so is most of the usefulness of the computer.

Heuristic Scanners

A recent addition to scanners is intelligent analysis of unknown code, currently referred to as heuristic scanning. It should be noted that heuristic scanning does not represent a new type of anti-viral software. More closely akin to activity monitoring functions than traditional signature scanning, this looks for suspicious sections of code that are generally found in viral programs. While it is possible for normal programs to try to "go resident," look for other program files, or even modify their own code, such activities are telltale signs that can help an informed user come to some decision about the advisability of running or installing a given new and unknown program. Heuristics, however, may generate a lot of false alarms, and may either scare novice users or give them a false sense of security after "wolf" has been cried too often.

Change Detection

Change detection software examines system and program files and configurations, stores the information, and compares it against the actual configuration at a later time. Most of these programs perform a checksum or cyclic redundancy check (CRC) that will detect changes to a file even if the length is unchanged. Some programs will even use sophisticated encryption techniques to generate a signature that is, if not absolutely immune to malicious attack, prohibitively expensive, in processing terms, from the point of view of a piece of malware.

Change detection software should also note the addition of completely new entities to a system. It has been noted that some programs have not done this and allowed the addition of virus infections or malware.

Change detection software is also often referred to as integrity-checking software, but this term may be somewhat misleading. The integrity of a system may have been compromised before the establishment of the initial baseline of comparison.

A sufficiently advanced change-detection system, which takes all factors including system areas of the disk and the computer memory into account, has the best chance of detecting all current and future viral strains. However, change detection also has the highest probability of false alarms because it will not know whether a change is viral or valid. The addition of intelligent analysis of the changes detected may assist with this failing.

Gratuitous Summary Opinion

Malware is a problem that is not going away. Unless systems are designed with security as an explicit business requirement, which current businesses are not supporting through their purchasing decisions, malware will be an increasingly significant problem for networked systems.

It is the nature of networks that a problem for a neighboring machine may well become a problem for local systems. To prevent this, it is critical that the information security professional help business leaders recognize the risks incurred by their decisions and help mitigate those risks as effectively and economically as possible. With computer viruses and similar phenomena, each system that is inadequately protected increases the risk to all systems to which it is connected. Each system that is compromised can become a system that infects others. If you are not part of the solution in the world of malware, you are most definitely part of the problem.

Glossary

This glossary is not a complete listing of malware-related terms. Many others can be found in the security glossary posted at http://victoria.tc.ca/ techrev/secgloss.htm and mirrored at http://sun.soci.niu.edu/~rslade/ secgloss.htm.

Activity monitor: A type of anti-viral software that checks for signs of suspicious activity, such as attempts to rewrite program files, format disks, etc. Some versions of activity monitor will generate an alert for such operations, while others will block the behavior.

ANSI bomb: Use of certain codes (escape sequences, usually embedded in text files or e-mail messages) that remap keys on the keyboard to commands such as DELETE or FORMAT. ANSI (the American National Standards Institute) is a short form that refers to the ANSI screen formatting rules. Many early MS-DOS programs relied on these rules and required the use of the ansi.sys file, which also allowed keyboard remapping. The use of ansi.sys is very rare today.

Anti-viral: Although an adjective, frequently used as a noun as a short form for anti-virus software or systems of all types.

AV: An abbreviation used to distinguish the anti-viral research community (AV) from those who call themselves *virus researchers* but who are primarily interested in writing and exchanging viral programs (vx). Also an abbreviation for anti-virus software. *See also* vx.

Backdoor: A hidden software or hardware mechanism that can be triggered to permit system protection mechanisms to be circumvented. The function will generally provide unusually high, or even full, access to the system either without an account or from a normally restricted account. Synonymous with trap door, which was formerly the preferred usage. Usage *back door* is also very common.

BSI: A boot-sector infector; a virus that replaces the original boot sector on a disk, which normally contains executable code.

Change detection: Anti-viral software that looks for changes in the computer system. A virus must change something, and it is assumed that program files, disk system areas, and certain areas of memory should not change. This software is very often referred to as *integrity checking* software, but it does not necessarily protect the integrity of data, nor does it always assess the reasons for a possibly valid change. Change detection using strong encryption is sometimes also known as *authentication software*.

Companion virus: A type of viral program that does not actually attach to another program, but which interposes itself into the chain of command so that the virus is executed before the infected program. Most often, this is done by using a similar name and the rules of program precedence to associate itself with a regular program. Also referred to as a *spawning virus*.

DDoS: Distributed denial of service. A form of network denial-of-service (DoS) attack in which a master computer controls a number of client computers to flood the target (or victim) with traffic, using backdoor agent, client, or zombie software on a number of client machines.

Disinfection: In virus work, the term can mean either the disabling of a virus's ability to operate, the removal of virus code, or the return of the system to a state identical to that prior to infection. Because these definitions can differ substantially in practice, discussions of the ability to disinfect an infected system can be problematic. Disinfection is the means users generally prefer to use in dealing with virus infections, but the safest means of dealing with an infection is to delete all infected objects and replace with safe files from backup.

Dropper: A program, not itself infected, that will install a virus on a computer system. Virus authors sometimes use droppers to seed their creations in the wild, particularly in the case of boot-sector infectors. The term *injector* may refer to a dropper that installs a virus only in memory.

False negative: There are two types of false reports from anti-viral or anti-malware software. A false negative report is when an anti-viral reports no viral activity or presence when there is a virus present. References to false negatives are usually only made in technical reports. Most people simply refer to an anti-viral *missing* a virus. In general security terms, a false negative is called a *false acceptance* or *Type II error*.

False positive: The second kind of false report that an anti-viral can make is to report the activity or presence of a virus when there is, in fact, no virus. False positive has come to be very widely used among those who know about viral and anti-viral programs. Very few use the analogous term, *false alarm*. In general security terms, a false positive is known as a *false rejection* or *Type I error*.

File infector: A virus that attaches itself to, or associates itself with, a file, usually a program file. File infectors most often append or prepend themselves to regular program files, or they overwrite program code. The file infector class is often also used to refer to programs that do not physically attach to files but associate themselves with program filenames. (*See* system infector, companion.)

Heuristic: In general, heuristics refer to trial-and-error or seat-of-the-pants thinking rather than formal rules. In anti-viral jargon, however, the term has developed a specific meaning regarding the examination of program code for functions or opcode strings known to be associated with viral activity. In most cases, this is similar to activity monitoring but without actually executing the program; in other cases, code is run under some type of emulation. Recently, the meaning has expanded to include generic signature scanning meant to catch a group of viruses without making definite identifications.

Infection: In a virus, the process of attaching to or associating with an object in such a way that, when the original object is called, or the system is invoked, the virus will run in addition to or in place of the original object.

Kit: Usually refers to a program used to produce a virus from a menu or a list of characteristics. Use of a virus kit involves no skill on the part of the user. Fortunately, most virus kits produce easily identifiable code.

Packages of anti-viral utilities are sometimes referred to as toolkits, occasionally leading to confusion of the terms.

Logic bomb: A resident computer program that triggers the perpetration of an unauthorized act when particular states of the system are realized.

Macro virus: A macro is a small piece of programming in a simple language, used to perform a simple, repetitive function. Microsoft's Word Basic and VBA macro languages can include macros in data files and have sufficient functionality to write complete viruses.

Malware: A general term used to refer to all forms of malicious or damaging software, including viral programs, Trojans, logic bombs, and the like.

Multipartite: Formerly a viral program that infects both boot sector/MBRs and files. Possibly now a virus that will infect multiple types of objects or reproduces in multiple ways.

Payload: Used to describe the code in a viral program that is not concerned with reproduction or detection avoidance. The payload is often a message but is sometimes code to corrupt or erase data.

Polymorphism: Techniques that use some system of changing the form of the virus on each infection to try to avoid detection by signature-scanning software. Less sophisticated systems are referred to as *self-encrypting*.

RAT (Remote-Access Trojan): A program designed to provide access to, and control over, a network-attached computer from a remote computer or location, in effect providing a backdoor.

Scanner: A program that reads the contents of a file looking for code known to exist in specific viral programs.

Script virus: It is difficult to make a strong distinction between script and macro programming languages, but generally a script virus is a stand-alone object contained in a text file or e-mail message. A macro virus is generally contained in a data file, such as a Microsoft Word document.

Social engineering: Attacking or penetrating a system by tricking or subverting operators or users rather than by means of a technical attack. More generally, the use of fraud, spoofing, or other social or psychological measures to get legitimate users to break security policy.

Stealth: Various technologies used by viral programs to avoid detection on disk. The term properly refers to the technology and not a particular virus.

System infector: A virus that redirects system pointers and information in order to infect a file without actually changing the infected program file. (This is a type of stealth technology.) Or, a virus that infects objects related to the operating system.

Trojan horse: A program that either pretends to have, or is described as having, a (beneficial) set of features but that, either instead or in addition, contains a damaging payload. Most frequently, the usage is shortened to *Trojan*.

Virus, computer: Researchers have not yet agreed on a final definition. A common definition is "a program that modifies other programs to contain a possibly altered version of itself." This definition is generally attributed to Fred Cohen, although Dr. Cohen's actual definition is in mathematical form. Another possible definition is "an entity that uses the resources of the host (system or computer) to reproduce itself and spread, without informed operator action."

vx: An abbreviated reference to the "Virus eXchange" community; those people who consider it proper and right to write, share, and release viral programs, including those with damaging payloads. Probably originated by Sara Gordon, who has done extensive studies of the virus exchange and security-breaking community and who has an aversion to using the Shift key.

Wild, in the: A jargon reference to those viral programs that have been released into, and successfully spread in, the normal computer user community and environment. It is used to distinguish those viral programs that are written and tested in a controlled research environment, without escaping, from those that are uncontrolled *in the wild*.

Worm: A self-reproducing program that is distinguished from a virus by copying itself without being attached to a program file, or that spreads over computer networks, particularly via e-mail. A recent refinement is the definition of a worm as spreading without user action, for example by taking advantage of loopholes and trapdoors in software.

Zombie: A specialized type of backdoor or remote access program designed as the agent, or client (middle layer) component of a DDoS (Distributed Denial of Service) network.

Zoo: Jargon reference to a set of viral programs of known characteristics used to test anti-viral software.

Acknowledgments

The author would like to thank David Harley and Lee Imrey for their valuable contributions to this chapter.

References

1. Cohen, Fred, 1994, *A Short Course on Computer Viruses*, 2nd ed., Wiley, New York.
2. Ferbrache, David, 1992, *A Pathology of Computer Viruses*, Springer-Verlag, London.
3. Gattiker, Urs, Harley, David, and Slade, Robert, 2001, *Viruses Revealed*, McGraw-Hill, New York.
4. Highland, Harold Joseph, 1990, *Computer Virus Handbook*, Elsevier Advanced Technology, New York.
5. Hruska, Jan, 1992, *Computer Viruses and Antivirus Warfare*, 2nd ed., Ellis Horwood, London.
6. Kane, Pamela, 1994, *PC Security and Virus Protection Handbook*, M&T Books, New York.
7. Slade, Robert Michael, 1996, *Robert Slade's Guide to Computer Viruses*, 2nd ed., Springer-Verlag, New York.
8. Slade, Robert Michael, 2002, Computer viruses, *Encyclopedia of Information Systems*, Academic Press, San Diego.
9. Solomon, Alan, 1991, *PC Viruses: Detection, Analysis, and Cure*, Springer-Verlag, London.
10. Solomon, Alan, 1995, *Dr. Solomon's Virus Encyclopedia*, S&S International PLC, Aylesbury, U.K.
11. Vibert, Robert S., 2000, *The Enterprise Anti-Virus Book*, Segura Solutions Inc., Braeside, Canada.
12. Virus Bulletin, 1993, *Survivor's Guide to Computer Viruses*, Abingdon, U.K.

An Introduction to Hostile Code and Its Control

Jay Heiser

VIRUSES AND OTHER FORMS OF HOSTILE CODE, OR "MALWARE," BECAME A UNIVERSALLY EXPERIENCED PROBLEM EARLY IN THE PC ERA, AND THE THREAT CONTINUES TO GROW. *The ICSA Virus Prevalence Survey* reported in 1999 that the infection rate had almost doubled during each of the previous four years. Malware has the potential to subvert firewalls, hijack VPNs, and even defeat digital signature. Hostile code is the most common source of security failure, and it has become so prevalent that its control must be considered a universal, baseline practice. Without an understanding of malware — what it is, what it can do, and how it works — malware cannot be controlled. It is ironic that despite the increasing rate of hostile code infection, the attention given this subject by academics and engineering students is declining. Attack code sophistication and complexity continues to increase, but fortunately, the appropriate response is always good system hygiene and administration.

DEFINITION OF HOSTILE CODE

Hostile code is program data surreptitiously introduced into a computer without the explicit knowledge or consent of the person responsible for the computer. Whatever the purported intent of its creator, code inserted covertly by an outside party can never be considered benign. If it is not approved, it has to be treated as hostile code. Vendors of anti-virus (AV) software have identified approximately 50,000 known viruses. In reality, only about 5 percent of these viruses are ever reported "in the wild," most commonly on Joe Wells's Wild List. (The Wild List is a regularly updated report of malware that has actually been observed infecting real systems. See Exhibit 23-1 for the URL.) Most of these are variations on a few hundred

Exhibit 23-1. Internet resources.

Malware Information Pages	
The Computer Virus Myths Page	http://kumite.com/myths/
IBM's Anti-Virus Online	http://www.av.ibm.com
The WildList Organization International	http://www.wildlist.org/
BackOrifice Resource Center	http://skyscraper.fortunecity.com/cern/600
The BackOrifice Page	http://www.nwi.net/~pchelp/bo/bo.htm
The NetBus Page	http://www.nwi.net/~pchelp/nb/nb.htm
Ports used by Trojans	http://www.simovitz.com/nyheter9902.html
AV Product Test Sites	
Virus Bulletin 100% Awards	http://www.virusbtn.com/100
Virus Test Centre	http://agn-www.informatik.uni-hamburg.de/vtc
Check-Mark certified anti-virus products	http://www.check-mark.com/
ICSA certified anti-virus products	http://www.icsa.net/html/communities/antivirus/certification/certified_products/

well-known examples, and only a small number of viruses account for most attacks. Complicating an understanding of hostile code, simple terms such as "virus" and "Trojan horse" are used imprecisely, blurring a potentially useful distinction between cause and effect. This chapter familiarizes the practitioner with the most common malware terminology, and helps them recognize different contexts in which their meaning changes.

INFECTION AND REPRODUCTION

Analysis of the transmission mechanism for a specific example of hostile code starts by determining two things: (1) if it is self-reproducing, and (2) if it requires the unwitting assistance of a victim. While fear of autonomous attack by self-replicating code is understandable, *manual insertion* is the most reliable way for an attacker to install hostile code. If assailants can gain either physical or remote access to a system, then they have the opportunity to install malware. Many network services, such as FTP, TFTP, and HTTP (and associated poorly written CGI scripts), have been used to upload hostile code onto a victim system. Hacker Web sites contain details on remote buffer overflow exploits for both NT and UNIX, making it possible for script kiddies to install code on many unpatched Web servers. Manual insertion is not very glamorous, but it works.

Cyberplagues, code designed to reproduce and spread itself, takes one of two different forms. A **virus** is hostile code that parasitically attaches to some other code, and is dependent on that code for its transmission. This is completely analogous to a biological virus, which alters the genetic

content of its victim, using its victim for reproduction. Unfortunately, the word "virus" has also taken on a secondary meaning as a generic moniker for all forms of hostile code. This meaning is perpetuated by using the term "anti-virus" to describe commercial software products that actually search for a number of forms of malware. A true virus can only spread with the participation of its victim. Host infection occurs when a contaminated file is executed, or when a floppy with an infected boot sector is read. Because users do not log directly into them, servers are less likely to contract viruses than are workstations. However, users who have write access to data on servers, either group configuration files or shared data, will infect files on the server that can spread to all users of the server. If they are write protected, server executables can only be infected when someone with write privilege, such as an administrator, runs a file containing a virus.

A **worm** is self-reproducing hostile code that has its own discrete existence. It is a stand-alone executable that uses remote services to reproduce itself and spread to other systems through a network. This is analogous to a biological bacterium (within the parasite pantheon, some experts distinguish a bacterium from a worm). Worms do not require the victim's participation — they rely on technical vulnerabilities that are the result of bugs or poor configuration. Because they spread by exploiting network vulnerabilities, and do not require a victim's participation, servers are just as vulnerable to worms as workstations are. They are probably more vulnerable, because they typically have more network services running.

A **Trojan horse** is an artifact with an ulterior hostile effect that appears desirable to the victim, tricking the victim into transferring it through a security perimeter so that its hostile intent can be manifested within the protected area. Examples of Trojan horses on computers are e-mailed greeting cards and games that include a hostile payload. The term "Trojan horse" is often applied to any hostile code that is nonreproducing. This secondary usage is imprecise and misleading; it does not explain the infection process, the trigger event, or the effect. This meaning is used in two different contexts. Most recently, it refers to nonreproducing hostile remote control applications, such as Back Orifice and NetBus, which often — but not always — spread as the payload of an e-mailed Trojan horse. NetBus, for example, is often surreptitiously bundled with the Whack-a-Mole video game. More traditionally, and especially on UNIX hosts, the term refers to a manually inserted hostile executable that has the same name as a legitimate program, or as a hostile program that mimics the appearance of a legitimate program, such as a login screen. An effective security practitioner is sensitive to these different meanings for commonly used terms.

Logic bombs are manually inserted, nonreproducing code created by system insiders, usually for revenge. For example, a system administrator

might create a utility that is designed to delete all the files on a computer two weeks after that employee leaves a job. There have been cases where software vendors have included logic bombs in their product to encourage prompt payment. These software vendors have usually lost in court.

The term **"backdoor"** most accurately applies to a capability. A backdoor is a hidden mechanism that circumvents existing access controls to provide unauthorized access. Historically, some software developers have left backdoors into their application to facilitate troubleshooting. Backdoors can also be provided through system configuration. If a UNIX system administrator or intruder creates a copy of the shell and sets it to be SUID root, it could also be considered a form of backdoor.

Executable Content

Every new technology brings new risks. The convenience of executable content, data files that have some sort of programming and execution capability, is undeniable, but executable content is also a marvelously convenient mechanism for hostile capability.

Macro viruses are hostile code applications written in the macro language of application software. The ability to automatically launch a macro when a file is opened, and the power to access virtually any system function from within that macro, make some applications particularly vulnerable. Microsoft Word documents are the most widely shared form of executable content, making them an efficient malware vector. In late 1995, Concept was the first macro virus observed in the wild. Within two years, Microsoft Word macro viruses had become the most frequently reported form of malware.

Self-extracting archives include executable zip files, Windows setup files, and UNIX shell archives (shar files). As a convenience to the recipient, a set of files (usually compressed) is bundled into a single executable file, along with a script to extract files into the appropriate directories, and make any necessary changes to system configuration files. When the archive is executed, it extracts its components into a temporary directory; and if an installation script is included, it is automatically launched. A user executing such a self-extracting object must trust the intentions and abilities of the archive's creator. It is a simple matter to modify an existing archive to include a piece of malware, transmogrifying a legitimate file into a Trojan horse.

Mobile code is a form of executable content becoming increasingly prevalent on the Internet. Java, JavaScript, ActiveX, and Shockwave are used to create Web-based objects that are automatically downloaded and locally executed when they are browsed. The environments used to run mobile code reside either within the browser or in downloadable browser plug-ins. The level of system access afforded mobile code interpreters varies, but

the user's browser has full access to the user's system privileges — use of mobile code requires a trust in the technical capabilities and configuration of the mobile code execution environment. At the time of this writing, no hostile mobile code exploits have ever been documented in the wild.

Complex Life Cycles

Replicating hostile code has a life cycle, just like biological pathogens, and the increasing sophistication of malware life cycles is enabling malware to take greater advantage of infrastructure opportunities both to evade controls and to maximize infection rate. **Propagation** is the life stage in which malware reproduces itself in a form suitable for the actual **infection**. As described in several examples below, some replicating code has multiple propagation methods. After infection, hostile programs often enter a **dormancy** period, which is ended by a **triggering event**. The trigger may be a specific date and time, a specific user action, the existence of some specific data on the computer, or possibly some combination of any of the above. When the trigger event occurs, an action is taken. The virus code that performs this action is referred to as the **payload**. When the action is performed, it is sometimes referred to as **payload delivery**. The payload may delete data, steal data and send it out, attempt to fool the user with bogus messages, or possibly do nothing at all. Self-replicating hostile code completes its life cycle by propagating itself.

A 1999 attack called Explorezip provides a good example of malware with a complex life cycle. Explorezip is a worm; it does not infect files. It also has a hostile payload that attacks and deletes certain kinds of data files, such as Word documents. It first spreads as a Trojan horse, masquerading as a legitimate message from a known correspondent. Explorezip actually mails itself — the owner of an infected PC does not send the message personally. If the recipient clicks and launches the Explorezip code attached to the phony mail message, their PC becomes infected. The next time their computer starts, the hostile code is activated. It immediately begins to reproduce using a secondary mechanism, spreading across the intranet looking for vulnerable Windows shares so it can copy itself to other PCs. The triggering event for the primary infection mechanism is the reception of mail. Whenever an infected victim receives a new mail message, Explorezip replies with a bogus message containing a copy of itself, potentially spreading itself to another organization. Once an infection has occurred, shutting off the e-mail server may not halt its spread because it can also reproduce through the network using file sharing. This combination of two infection mechanisms complicated the response to Explorezip. Explorezip increases the chance of a successful infection by appropriating its victim's e-mail identity; it is a spoof that takes advantage of correspondent trust to lull recipients into accepting and executing an e-mail attachment that they may otherwise avoid.

EXPLOITS

Autonomous Attacks

Viruses and worms are autonomous. They are self-guided robots that attack victims without direction from their creator. Happy99 spreads as a Trojan horse, purportedly a fun program to display fireworks in a window (sort of a New Year's celebration). While it is displaying fireworks, it also patches WSOCK.DLL. This Winsock modification hooks any attempts to connect or send e-mail. When the victim posts to a newsgroup or sends e-mail, Happy99 invokes its own executable, SKA.DLL, to send a UUENCODED copy of itself to the news group or the mail recipients. As illustrated in Exhibit 23-2, Caligula is a Word macro virus that when triggered searches for PGP key rings (a PGP key ring is the data file that includes a user's encrypted private key for Pretty Good Privacy mail and file encryption). If it finds a PGP key ring, it FTPs it to a Web site known to be used for the exchange and distribution of viruses. Caligula is an example of autonomous code that steals data.

Melissa is designed to covertly e-mail infected documents to the first 50 e-mail addresses in a victim's address book. The document — which might contain sensitive or embarrassing information intended only for internal

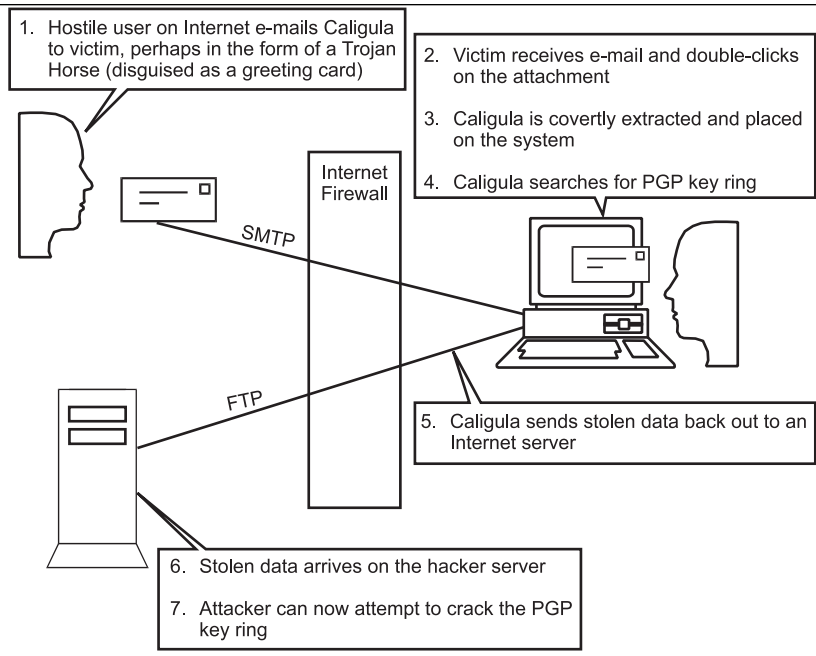


Exhibit 23-2. How Caligula steals data.

use — will be unwittingly sent to those 50 addresses. While it probably was not designed to steal data, a Melissa infection can easily result in the loss of privacy. At the time of this writing, there are at least 20 different families of mail-enabled hostile code. Caligula, Happy99, Explorezip, and Melissa are all examples of malware that take advantage of network capabilities. As shown in Steps 1 and 5 of Exhibit 23-2, most firewalls are configured to allow all incoming SMTP traffic, and all outgoing FTP connections. Attack code can easily use these protocols to circumvent a firewall and perform its intended task without human direction. However, autonomous attacks lack flexibility. Attackers desiring a more flexible and personal mechanism must use some form of interactive attack.

Interactive Attacks

Fancifully named programs, such as BackOrifice and NetBus, represent a significant change in the use of hostile code to attack computers. If installed on a victim's PC, these programs can serve as backdoors, allowing the establishment of a surreptitious channel that provides an attacker virtually total access to the PC across the Internet. A pun on Microsoft's BackOffice, BackOrifice (or BO) is the first Windows backdoor to be widely spread. Once a BO server has been inserted on a PC, either manually or as a Trojan horse, it can be remotely accessed using either a text or graphical client. The BO server allows intruders to execute commands, list files, silently start network services, share directories, upload and download files, manipulate the registry, list processes, and kill processes. Reminiscent of UNIX attacks, it allows a Windows machine to be a springboard for attacks on other systems. BO supports the use of accessory plug-ins, and several have been developed (continuing the naming convention with catchy puns like Butt Trumpet). Plug-ins allow BO to be wrapped into a self-extracting executable or to ride piggyback on another program. Once it has been installed, another plug-in announces itself on an IRC group. Several dozen surreptitious channel remote control applications are available. These programs can be used as legitimate system administration tools, and their creators steadfastly maintain that this is their purpose. Certainly, attackers also exploit commercial remote control applications, such as pcAnywhere. However, hostile backdoor exploits have special features designed to make them invisible to their victims, and they have other capabilities that facilitate data theft.

Once accidentally installed by the hapless victim, these programs listen for connection attempts on specific ports. Starting in late 1998, CERT reported a high rate of connection attempts on these ports across the Internet. Systems connected to the Internet full-time, such as those using cable modems or DSL, can expect to be scanned several times a week. The appeal of these programs to an attacker should be obvious. Someone motivated to

steal or alter specific data can easily do so if they can install a remote control application on a suitable target host and access it over a network. Kiddie scripts are available to piggyback NetBus or BackOrifice onto any executable that an attacker feels a victim would be willing to execute, creating a customized Trojan horse. Attackers too lazy to "trojanize" the remote control server program themselves can just send potential victims a video game that is already prepared as a NetBus Trojan, such as Whack-a-Mole. Once a vulnerable system is created or found, the keystroke recording feature can be used to compromise the passwords, which control access to secret keys, threatening a variety of password-protected security services, including S-MIME, PGP, and SSL. Most VPN clients are only protected by a password. If this password were to be compromised through a surreptitious backdoor's keystroke recording function, someone who later used the backdoor to remotely connect to the infected PC would be able to appropriate the victim's identity and corresponding VPN privileges. This could lead to the compromise of a corporate network.

MEME VIRUSES

Just the threat of a virus is sufficient to impact productivity — **virus hoaxes** can cause more disruption than actual viruses. Typically, a naïve user receives an e-mail message warning them of the dire consequences of some new form of computer virus. This message, full of exclamation points and capital letters, instructs the user to warn as many people as possible of some imminent danger (an example is shown in Exhibit 23-3). Viral hoax creators take advantage of a human need to feel important, and enough users are willing to forward these messages to their friends and co-workers that the deception spreads quickly and widely. Just like actual viruses, some virus hoaxes can live for years, flaring up every six to twelve months in a flurry of unproductive e-mail. When thousands of corporate users receive a bogus warning simultaneously, the effect on corporate productivity can be significant. No e-mail warning from an individual about a new virus should be taken seriously before doing research. Every vendor of anti-virus software has a Web page cataloging known hostile code, and several excellent Web sites are dedicated to the exposure and discussion of viral hoaxes. Virus hoax response can only be accomplished procedurally, and must be addressed in the organizational policy on hostile code or e-mail usage. Users must be instructed to report concerns to the IS department, and not take it upon themselves to inform the entire world. The Internet has proven to be an extraordinarily efficient mechanism for misinformation dissemination, and virus scares are not the only form of disruptive hoax. Well-meaning users are also prone to spreading a variety of similar practical joke messages. Classic e-mail pranks include chain letters, "make a wish" requests for a dying boy who collects business cards, and petitions to the government for some upcoming fictitious decision.

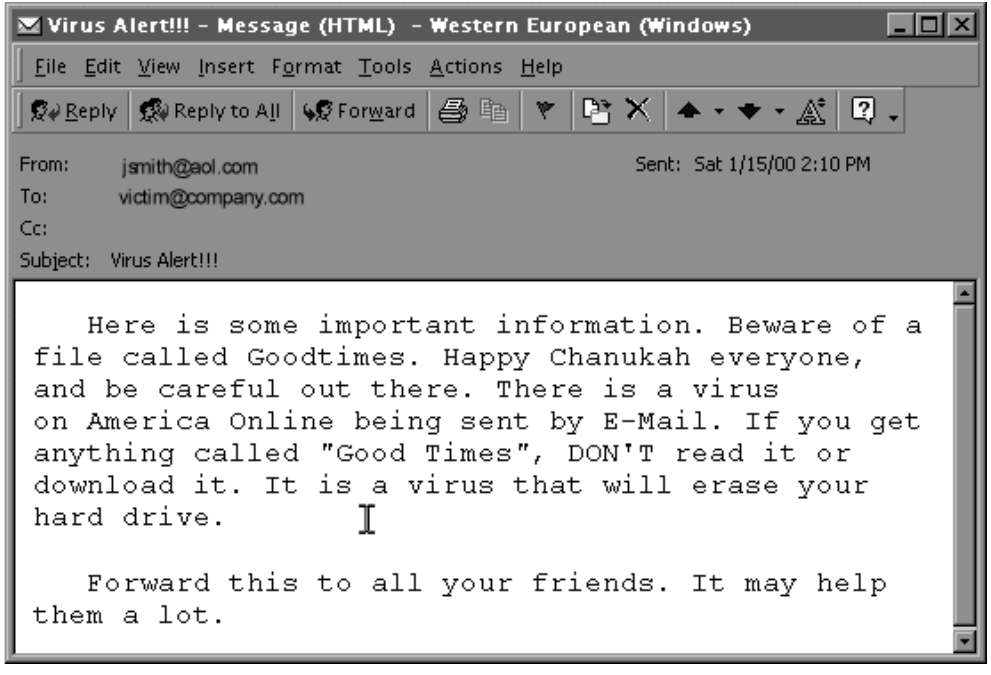


Exhibit 23-3. Hoaxes are easy to recognize.

COUNTERMEASURES

Hostile code control requires a comprehensive program simultaneously addressing both technical and human issues. It is safe to assume that some infections will occur, so prepare a recovery strategy before it is needed.

Policy and Procedure

The first step in computer security is always well-conceived policy establishing organizational priorities and basic security rules. Hostile code infections are prevented through procedural and technical countermeasures; policy must address both. Users need to be aware of the danger associated with the acceptance and use of files from external sources, and trained not to automatically double-click on e-mail attachments. Anti-virus (AV) software must be installed on every desktop and updated regularly. Corporate policy needs to set rules for e-mail. In addition to hostile code, organizational e-mail policy should address chain letters, hoaxes, and other forms of harmful internal communication. Policy must address response and cleanup. A malware response team should be appointed and charged with creating procedures for the rapid response to and recovery from an infection.

Creating policy is just the first step; it must be implemented through guidelines and procedures. Effective implementation involves more than just the publication of a set of rules. If corporate staff understands the nature of the threat, is aware of their role in spreading infection, and is provided with simple but effective behavioral guidelines, they will become the allies of the IS department. Awareness is not a one-time event provided to new hires. Existing staff must be periodically reminded of their responsibility to protect their organization. Media scares about new forms of hostile code can be an opportunity to educate the users and help them understand that security is an ongoing process. Clamp down on hoaxes and chain mail immediately. Remember the fable about the little boy who cried wolf. Users subjected to continuous warnings about dangers that never appear will become inured, and will not respond appropriately when an actual event occurs.

Good Hygiene

Maintaining optimal configuration and following best practices for administration results in robust systems that are resistant to security threats. Effective configuration management ensures that all systems will be appropriately configured for both performance and security, and it facilitates their recovery in case of a disaster or failure. System security should be as tight as practical, protecting sensitive system configuration and corporate data from unauthorized or accidental deletion or change. Excessive use of administrative privileges increases the risk of a failure. Every time

someone is logged in with full administrative privileges, the negative consequences of a mistake are increased. Accidentally executing hostile code while logged in as an administrator can be disastrous. UNIX users should not login as root, but should use the **sudo** command to temporarily access root privileges when needed. When NT users read mail or work on documents, they should be logged into an account that is not a member of the administrative group. Human attackers and worms need access to network services in order to compromise systems remotely, so both servers and workstations should avoid running unnecessary network applications.

System and Data Backups. The performance of regular and complete system data backups is the most effective security countermeasure. No security administrator can ever guarantee that a system will not fail, or that some unforeseen attack will not succeed. Having complete restore capability is a last-ditch defense, but it is a reliable defense. Organizational policy should mandate system backups and provide standards for backup storage. Depending on the volume and significance of the information, this policy might direct that some data be backed up on a daily basis or even in real-time, while other data be backed up on a weekly basis. Backups must be stored off-site. While redundant systems, such as RAID, provide a high level of reliability, if a site becomes unusable, the data would probably be inaccessible. Always test restoration capability. While it is helpful to perform a read test whenever data is backed up, this test does not guarantee that the tapes can be used to perform a restore. Develop procedures for periodically testing file restoration. It is inconvenient to back up laptops, but increasingly they contain large amounts of critical corporate data. Requirements for portable computers should be included as part of the data backup policy.

Anti-virus Software

When properly managed, AV software can be highly effective. It uses a variety of mechanisms to identify potentially hostile code; scanning is the most effective. Anti-virus scanning engines methodically search through system executables and other susceptible files for evidence of known malware. A file called the *virus definition* file contains signatures of known hostile code. The signature is a sequence of bits that researchers have identified as being unique to a specific example of hostile code. Searching every executable, Word document, and boot record for each of 50,000 signatures would take an unacceptably long time. Viral code can only be inserted at certain spots within an existing executable, so scanning engines increase performance by only searching specific parts of an executable. Over the years, virus writers have devised several methods to defeat virus scanners. Polymorphic viruses mutate, changing their appearance every time they reproduce; but when they execute, they revert to their original form

within system memory. Modern anti-virus software actually runs executables within a CPU simulator first, so that polymorphic viruses can decrypt themselves safely in a controlled environment where their signatures can be recognized. Unfortunately, anti-virus scanners are blissfully unaware of new hostile code until the AV vendors have the opportunity to analyze it and update their definition files. AV software vendors share newly discovered examples of hostile code, allowing each vendor the opportunity to update their own definition files. Most AV vendors update their definitions every four weeks, unless they become aware of some especially harmful virus and provide an interim update. This latency prevents scanning from ever being 100 percent effective, and is the reason why users must be trained to protect themselves. Several techniques have been developed to detect previously unknown hostile code, such as heuristics and behavior blocking, but results have been mixed. It is relatively easy to anticipate certain behaviors that file and boot sector viruses will follow. For example, most AV products can be configured to prevent writing to the master boot record. Monitoring more complex behaviors increases the potential for user disruption from false positives.

AV vendors offer several choices as to when scanning occurs. Scanning can be performed manually, which is a good idea when the virus definition files have been updated, especially if there is reason to believe that a previously undetectable form of hostile code might be present. Scanning can also be scheduled to occur periodically. The most reliable way to prevent the introduction of malware to a PC is to automatically scan files that potentially contain hostile code before accepting them. Before e-mail became a universal means for file exchange, floppy disks with infected boot sectors were the most common infection vector. During the past few years, hostile code has been more likely to spread via e-mail. AV software with real-time capabilities can be configured to scan files for the presence of hostile code whenever a floppy disk is inserted, a file is copied or read, or an e-mail attachment is opened. PC anti-virus software real-time detection capabilities have proven effective at stopping the spread of recognized hostile code attached to e-mail messages. Running AV software in this mode does raise performance concerns, but the cost of faster hardware can be offset against the cost of downtime, cleanup, and loss of system integrity after a significant viral infection.

In addition to running AV software on the desktop, scanners can be server based. The automatic periodic scanning of file servers for hostile code will ensure that even if an individual desktop is misconfigured, malware stored on the server will eventually be discovered. An increasing number of products are available to scan e-mail attachments before the mail is placed in a user's incoming mailbox. It is a common misconception that a firewall is a total solution to Internet security. Firewalls are network perimeter security

devices that control access to specific network services — a limited task that they perform well. They are not designed to examine incoming data and determine whether it is executable or what it is likely to do. **Virus walls** are application-level countermeasures designed to screen out hostile code from e-mail. These products can often be run on the firewall or the mail server, but it is usually most practical to use a stand-alone machine for mail filtering, locating it between the firewall and the organizational mail server. Operating as an e-mail proxy, virus walls open each message, check for attachments, unarchive them, and scan them for recognizable hostile code using a commercial AV product. They are efficient at unzipping attachments, but they cannot open encrypted messages. If organizational policy allows incoming encrypted attachments, they can only be scanned at the desktop. E-mail scanners should be considered as an augmentation to desktop control — not as a replacement. Use different AV products on the virus wall and the desktop; the combination of two different products in series provides a better detection rate than either product alone. E-mail scanners can protect both incoming and outgoing mail; unfortunately, most organizations only scan incoming mail. Scanning outgoing mail can double the cost of a virus wall, but this should be balanced against the loss of goodwill or bad publicity that will occur when a customer or partner is sent a virus.

Exhibit 23-4 shows the different locations within an enterprise where hostile code can be controlled. An obvious location for the scanning of incoming content is the firewall, which is already a dedicated security device. However, the primary mission of a firewall is to provide access control at the transport level. From the purist point of view, it is inappropriate to perform application layer functions on a network security device. However, it is becoming common to provide this service on a firewall, and many organizations are doing it successfully. If the firewall has enough processing power to perform scanning in addition to its other duties, adding a scanning upgrade is an easy way to scan mail attachments and downloads. Be aware that the addition of new services to a firewall increases the risk of failure — it is easy to overload a firewall with add-ons. Mail scanning can also be performed on the mail server. This has the minor disadvantage of not protecting HTTP or FTP. The bigger disadvantage is the increased complexity and decreased performance of the mail server. Mail servers are often finicky, and adding additional functionality does not increase dependability. Organizations already using high-end firewall and mail servers should consider one or more dedicated proxy machines, which is the most scalable solution. It can easily be inserted immediately behind the firewall and in front of the mail server. Organizations that want immediate protection but do not have the desire or wherewithal to provide it in-house can contract with an outside provider. Increasingly, ISPs are offering a scanning option for incoming e-mail. Managed security service providers will remotely manage a firewall, including the maintenance of virus wall capabilities. The desktop is the

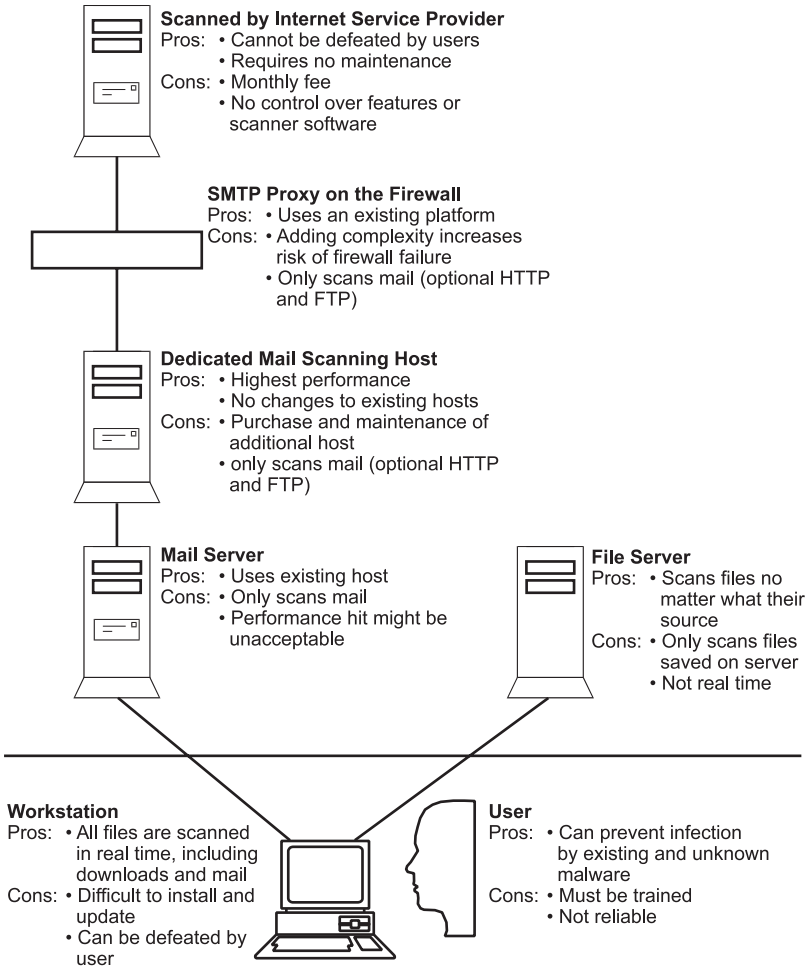


Exhibit 23-4. Hostile code control options.

most crucial place to control hostile code. If desktop systems could be reliably updated and users prevented from tampering with the configuration, there would be no need to scan anywhere else in the organization, but desktop scanning is difficult to maintain and the users cannot be trusted.

Cleaning. AV software not only detects hostile code, but also can be used to remove it. Removal of a virus from an executable is not always practical, but the AV vendors work very hard to provide automated cleaning of the hostile code most likely to be encountered in the wild. Although most users are capable of running a wizard and cleaning up their own system, organizational policy should provide them with guidance on what to

do when their AV software informs them an infection has been found. Even if users are allowed to clean up their own systems, their AV software should be configured to place a copy of all infected files in a quarantine area to facilitate later diagnosis. When infected, most organizations use their AV software to perform a cleanup; and if the cleanup is successful, the system is returned to production use. Any applications that cannot be repaired must be reinstalled or restored from a backup.

AV Software Configuration Should be Based on Policy. Several policy decisions have to be made in order to control hostile code with an anti-virus product. The most significant decision is whether the desktops will be individually administered or centrally administered. Even the worst product, when properly maintained and updated, outperforms the most effective product available if that product is improperly configured. It is not reasonable to expect users to make appropriate decisions concerning the configuration of their security software; and even when they are provided guidance, they cannot be trusted to reliably configure their own systems. Clearly, the trend is toward central administration, and the AV vendors are trying to accommodate this. Most AV products can be configured to periodically download definition files, automatically updating themselves. Software distribution tools available from Novell, Microsoft, and a number of independent software vendors can be used to push updates to user desktops. Use of a virus wall is also a form of central control. The choice of whether or not incoming files should be scanned before reaching the desktop is a policy decision. Likewise, policy should also address the scanning of outgoing files. Once policy exists that requires the centralized scanning of ingoing or outgoing mail, the choice of whether to scan on the firewall, on a dedicated mail scanning host, or on the existing mail server, is an implementation issue — not a policy decision.

Unless individual users experience problems, such as an unacceptably high number of false positives or a high rate of infection, desktop AV software will probably be configured to use the manufacturer's defaults on every internal system. Anti-virus software typically does not scan every file on the system — this would be a waste of time. On a Windows machine, the choice of files to be scanned is determined by their suffix. The vendor's recommendations for appropriate file types should not be changed unless hostile code has been consistently missed because of its file type. The software should be configured to automatically scan e-mail attachments and files at read time. Hostile code is rarely contracted through FTP or HTTP, but the overhead of scanning individual files is not noticeable, so the cost of scanning all Internet downloads is low.

Choosing a Product. The trade press is ill-equipped to evaluate anti-virus products. Reviews in popular computer magazines are more likely to

be misleading than helpful. At best, they tend to dwell on meaningless number games, comparing vendor's inflated claims for the number of recognized viruses. At worst, they concentrate on the attractiveness of the user interface. Only dedicated periodicals, such as *Virus Bulletin*, have the expertise to make valid comparisons between heavily marketed anti-virus products. The industry analyst firms usually have the expertise and objectivity to make useful recommendations on choice of virus control software. Organizations that subscribe to desktop or security bulletins from companies like this should see if they are eligible to receive reports on anti-virus products. Most AV vendors offer free mailing lists. These mailing lists serve a marketing function and tend to exaggerate the danger of newly discovered hostile code examples. Although AV vendor Web sites provide useful reference data on hostile code, their mailing lists are usually not helpful to the security practitioner. Several organizations test AV software and place their results on the Web. See Exhibit 23-1 for the URLs. *Virus Bulletin* (which is owned by an AV vendor), and the ICSA and Check-Mark (which are for-profit organizations) certify AV products and place the results on the Web. The Computer Science Department at the University of Hamburg is the only non-profit organization to methodically test AV products and regularly publish results.

Microsoft Word Macro Virus Control

Word macro viruses are the most prevalent form of hostile code. Microsoft Word documents support a programming language that is a variant of Visual Basic. Virtually anything that can be done on a PC can be done within a Word document using this language, including low-level machine language calls. They can be configured to execute automatically when a document is opened. Macros are a powerful tool for office automation, but they also place Word users at risk. There are several ways to reduce the macro virus risk, but none of them is foolproof. Word can be configured so that it does not automatically execute macros. When so configured, it will prompt the user for a decision whenever encountering a file with an autoexecute macro. Without training, users cannot be expected to make an appropriate decision, and even experienced users can accidentally push the wrong button. Just like executable viruses, macro viruses have become increasingly stealthy over time, and virus writers have developed several techniques for evading this automatic check. Word has the capability of storing documents in several formats. Only Word's native DOC format supports macros, so some organizations have chosen to distribute files in Rich Text Format (RTF). Windows 2000 can also store files in HTML with no loss of formatting, an even more portable format. Unfortunately, it is possible to change the extension on a DOC file to RTF or HTML and Word will still recognize it as a DOC file. Opening a DOC with an autoexecute macro — even when it is disguised with a different extension — will cause the macro to be executed. The safest choice is to

use an application that cannot run macros, such as Microsoft's free DOC file viewer. Downloadable from Microsoft's Web site, this utility can be safely used to view a file suspected of containing a macro virus.

Most macro viruses will infect NORMAL.DOT, the Word file containing default styles and macros. If the infection cannot be removed with AV software, remove NORMAL.DOT and Word will recreate it the next time it is started. Note that deleting NORMAL.DOT will result in a loss of all user-defined Word hot keys, macros, and changes to default styles. If this file is shared across the network, all users will contract the virus the next time they start Word. For this reason, a shared NORMAL.DOT file should always be configured as read-only.

Mobile Code Control: Java and ActiveX

Java is an interpreted programming language, while ActiveX is a Microsoft binary format. The two technologies are different, but from the point of view of a Web browser, they are alternate mechanisms for distributing code from a Web page to a user desktop for local execution. Mobile code is a security concern because it allows Web site operators control over what is executed on a user's desktop. It is important to remember that Java and ActiveX have never been exploited in a security-relevant way. The only known mobile code exploits have been demonstrations — there is no recorded example of an actual security failure involving mobile code on a production system. Unlike other more prevalent forms of malware, mobile code security remains a popular area of academic research, ensuring that security-relevant software bugs are identified and reported to the browser vendors.

Because it is a strongly typed language, Java is less susceptible to buffer overruns than C, making Java code more reliable and difficult to exploit. Java executes within a controlled environment called the Java virtual machine (JVM). When running within a browser, the 1.0 version of the JVM enforces its security policy using three different mechanisms. First, the applet class loader assigns a private namespace associated with the network origin of each downloaded applet, maintaining a separate and unique namespace for Java code loaded locally. Second, all applets pass through the applet code verifier, which checks for illegal code constructions. Finally, the Java security manager, a reference monitor, prevents the local reading and writing of files, and prevents applets associated with one host from accessing a different one. Sometimes referred to as the Java sandbox, the security manager only allows applets to do four things: they can run, they can access the screen, they can accept input, and they can connect back to their originating host. Several Java security bugs have been demonstrated in the laboratory by tricking the JVM into allowing applets access to hosts other than the originating host, or allowing them access to the local file system. Both Microsoft and Netscape quickly patched these

vulnerabilities. The limitations of Java 1.0 functionality should be clear. While it is an intrinsically safe environment, the lack of file system access limits its utility for transactions. Java 2.0 provides the capability for authorized applets to break out of the Java sandbox. The newer version of Java allows applets to be digitally signed. Compatible browsers will be able to allow controlled access to system resources on behalf of applets signed by approved parties.

ActiveX is Microsoft's trade name for compiled Windows executables that can be automatically distributed across the Internet as part of a Web page. It uses Microsoft's Component Object Module standard (COM). It does not have any form of sandbox, but uses a trust model similar to Java version 2.0. The Microsoft browser, Internet Explorer, checks the digital signature of any ActiveX objects presented to it by a Web server. Microsoft browsers support a hierarchy of security zones, each allowing greater access to system resources. Specific signers can be configured within the browser environment as being authorized for the access level of specific zones. A typical configuration might allow ActiveX originating from within an organization to have full access to a user's resources, but code originating from the Internet would have no special privileges. Unfortunately, if a user encounters an ActiveX object from an unrecognized signer, the default behavior is to ask the user what to do. Because the onus is on the user to determine what code is appropriate to operate, ActiveX has been widely criticized in the security community. For this model to work, users must be trained — which is relatively difficult. Microsoft provides a centralized configuration management tool for Internet Explorer, enabling an organization to centrally configure behavior on all desktops. Effective use of this capability should allow an organization to take full advantage of ActiveX internally without placing users at unnecessary risk.

Although neither Java nor ActiveX has ever been successfully exploited, several commercial products are available that protect desktops for both. Organizations wishing to ensure that mobile code is never activated on employee PCs can also control it at the perimeter. Many Web proxies, including those running directly on a firewall, can be configured to trap Java and ActiveX objects, shielding users from mobile code on the Internet. Security practitioners should be aware of the potential for mobile code failures, and know the countermeasures available in case a problem ever manifests itself. At the time of this writing, only the most sensitive organizations need to be concerned about mobile code risk.

WHY DOES HOSTILE CODE EXIST?

Why is so much malware floating around? The motivations behind the writing of hostile code are complex. In most cases, it is not necessarily an explicit desire to hurt other people, but it is often a form of self-actualization.

It is a hobby — carried to obsessive levels by some of the most successful virus writers. The quest for knowledge and the joy of parenthood are fun and satisfying. Virus creators are driven by Dr. Frankenstein's relentless curiosity on the nature of life itself. Once having created something that appears able to reproduce, it can be difficult to resist the temptation of experimenting in the ultimate laboratory, the Internet. Robert Morris, Jr., the creator of the Internet Worm, is probably not the only programmer who has experienced a sorcerer's apprentice moment and realized that their handiwork has succeeded beyond their wildest dreams — and beyond their sphere of control.

Many writers of self-replicating code belong to an extended virtual community where they socialize, exchanging ideas and code. Virus writers were early users of bulletin board systems, forums that have been transplanted to the Internet. The most desirable virus meeting places are closed, requiring the submission of a functioning virus as an initiation requirement. Created by neophytes, these initial efforts are typically simple variations of existing malware, which explains why such a high percentage of the thousands of identified hostile programs are closely related. Social status within the virus writing community is similar to other hacker subcultures. It is derived from technical prowess, which must be proven repeatedly as community members compete for superiority. Fame and respect derive from recognition within their social group of their superior skills, as demonstrated by clever coding and successfully propagating creations. There is undoubtedly a need on the part of some coders to overcome their inferiority feelings by exerting power of others as digital bullies, but studies of virus writers indicate that the challenge and social aspects are most significant. By attempting to evade AV software, virus writers demonstrate an awareness of the AV industry. Only the most socially obtuse programmer could fail to realize that AV software exists because people fear viruses and wish to avoid them.

Why Windows?

UNIX viruses are possible, but in practice, they are essentially nonexistent. There continue to be a few Macintosh viruses, but the overwhelming majority of malware attacks are aimed at Microsoft Windows systems. That which makes Windows most useful is also its greatest weakness — a characteristic not unique to computer security. Windows represents a monoculture — the majority of user workstations utilize the same operating environment, run the same application (Microsoft Word), and many use Microsoft Outlook for e-mail. As modern agriculture has shown that monoculture provides an opportunity for insects and disease, huge numbers of similar PCs are susceptible to a common infection. Exacerbating the low level of diversity is the high level of both internal and external connectivity, and the privileges granted to their unsophisticated operators make

Windows systems vulnerable. Users of Windows 98 are effectively the system administrator. NT is an operating system designed to meet the C2 requirements for access control, but normal users are often granted administrator privileges, effectively bypassing the system's built-in protection. Finally, the widespread use of a macro-enabled word processor means that executable content is pervasive. The combination of ubiquitous e-mail, a powerful word processor, weak access control, and unsophisticated users has resulted in macro viruses quickly becoming a universal threat.

CONCLUSION

While documented cases are low, a risk analyst needs to be aware that remotely inserted hostile code is an ideal way for motivated and skillful attackers to commit computer-based fraud or vandalize information. Malware already exists that steals passwords, and other forms of directed data theft are just as easy to accomplish. As easily customizable hostile code continues to proliferate, and as motivated external attackers become increasingly sophisticated, directed attacks will be carried out through e-mail. Organizations that are subject to either espionage or especially strong and unethical competitive pressure need to be on the lookout for customized attack code. Malware has been present throughout the PC era. While the cost of viral infections remains a matter of debate, despite the millions of dollars spent fighting malware, the rate of hostile code incidents continues to increase. The latest forms of Internet security countermeasures, such as firewalls, VPNs, and PKI, are vulnerable to software attack. Fortunately, control is relatively simple. A well-orchestrated combination of human effort and technical countermeasures has proven effective in maintaining an acceptably low rate of hostile code infection.

Bibliography

1. Cohen, Frederick B., *A Short Course on Computer Viruses*, Wiley, New York, 1994.
2. Denning, Dorothy E., *Information Warfare and Security*, Addison-Wesley, New York, 1999.
3. Gordon, Sarah, The Generic Virus Writer, presented at *The 4th International Virus Bulletin Conference,* Jersey, U.K., September 1994.
4. Gordon, Sarah, Technologically Enabled Crime: Shifting Paradigms for the Year 2000, *Computers and Security*, 1994.
5. Gordon, Sarah, Ford, Richard, and Wells, Joe, Hoaxes & Hypes, presented at the *7th Virus Bulletin International Conference,* San Francisco, CA, October 1997.
   (Sarah Gordon papers can be found at http://www.av.ibm.com/ScientificPapers/Gordon/)
6. Heiser, Jay, Java Security Mechanisms: A Three-sided Approach for the Protection of Your System, *Java Developer's Journal*, 2(3), 1997.
7. Kabay, Michel E., Tippett, Peter, and Bridwell, Lawrence M., *Fifth Annual ICSA Computer Virus Prevalence Survey*, ICSA, 1999.
8. Kephart, Jeffrey O., Sorkin, Gregory B., Chess, David M., and White, Steve R., Fighting Computer Viruses, *Scientific American*, 277(5), 88–93, November 1997.
9. McClure, Stuart, Scambray, Joel, and Kurtz, George, *Hacking Exposed*, Osborne, 1999.
10. Nachenberg, C., Computer Virus-Antivirus Coevolution, *Communications of the ACM*, January 1997.

11. National Institute of Standards and Technology, *Glossary of Computer Security Terminology*, NISTIR4659, 1991.
12. Schneier, Bruce, Inside Risks: The Trojan Horse Race, *Communications of the ACM*, 42(9), September 1999.
13. Slade, Robert, *Robert Slade's Guide to Computer Viruses*, Springer, 1996.
14. Smith, George C., *The Virus Creation Labs*, American Eagle Publications, 1994.
15. Solomon, Alan and Kay, Tim, *Dr. Solomon's PC Antivirus Book*, New Tech, 1994.
16. Spafford, Eugene H., Computer Viruses, *Internet Besieged*, Denning, Dorothy E., Ed., ACM Press, 1998.
17. Whalley, Ian, Testing Times for Trojans, presented at the *Virus Bulletin Conference,* October 1999, http://www.av.ibm.com/ScientificPapers/Whalley/inwVB99.html.

105

A Look at Java Security

Ben Rothke, CISSP

Introduction

Why should Java security concern you? Many push-based applications are being ported to Java. In addition, Java is one of the cornerstones of active content and an understanding of Java security basics is necessary for understanding the implications of push security issues.

A lot of people ask: "Why do I need Java security? I thought it was safe." Java as a language is basically safe and is built on top of a robust security architecture. But security breaches related to bugs in the browser, poorly written Java code, malicious Java programs, poorly written CGI scripts and JavaScript code, and others often occur. Moreover, placing the enforcement of a security policy in the browser, and thus in the hands of end users, opens up many opportunities for security measures to be defeated. In addition, many push vendors are relatively new start-ups that do not always understand mission-critical software and security needs. Such circumstances only exacerbate the security predicament.

While some people might opine that Java is too insecure to be used in production environments and that it should be completely avoided, doing so creates the situation where a tremendous computing opportunity is lost. While the company that decides to bypass Java relieves itself of Java security worries, that means that they also relinquish the myriad benefits that Java affords. In addition, a significant number of cutting-edge Internet-based activities, such as E-commerce, online trading, banking, and more, are all written in Java. Also, many firewall and router vendors are writing their management front-end applications in Java. When a company cuts itself off from Java, it may likely cut itself off from the next generation of computing technology.

Push-based programs are powerful and flexible Web tools, and where the Web is directed, but these programs, by their nature, are inherently buggy and untrustworthy. Now take a look at the Java security model.

A Quick Introduction to the Java Programming Language

The essence of Java is to be a portable and robust programming language for development of write-once programs. Java was created to alleviate the quandary of writing the same applications for numerous platforms that many large organizations faced in developing applications for large heterogeneous networks. To achieve this, the Java compiler generates class files, which have an architecturally neutral, binary intermediate format. Within the class file are Java bytecodes, which are implementations for each of the class' methods, written in the instruction set of a virtual machine. The class file format has no dependencies on byte-ordering, pointer size, or the underlying operating system, which allows it to be platform independent. The bytecodes are run via the runtime system, which is an emulator for the virtual machine's instruction set. It is these same bytecodes that enable Java to be run on any platform. Finally, two significant advantages that increase Java's security is that it is a well-defined and openly specified language.

While many systems subscribe to the security through obscurity model, Java achieves a significant level of security through being published. Anyone can download the complete set of Java source code and examine it for themselves. In addition, numerous technical security groups and universities have done their own audits of Java security.

The second area where Java security is increased is through its architectural definitions. Java requires that all primitive types in the language are guaranteed to be a specific size and that all operations defined must be performed in a specified order. This ensures that two correct Java compilers will never give different results for execution of a program, as opposed to other programming languages in which the sizes of the primitive types are machine- and compiler-dependent, and the order of execution is undefined except in a few specific cases.

Overview of the Java Security Model

The Java applet[1] security model introduced with the 1.0 release of Java SDK considers any Java code running in a browser from a remote source to be untrusted. The model anticipates many potential attacks, such as producing Java code with a malicious compiler (one that ignores any protection boundaries), tampering with the code in transit, etc. The goal of the Java security model is to run an applet under a set of constraints (typically referred to as a sandbox) that ensures the following:

- No information on the user's machine, whether on a hard disk or stored in a network service, is accessible to the applet.
- The applet can only communicate with machines that are considered to be as trusted as itself. Typically, this is implemented by only allowing the applet to connect back to its source.
- The applet cannot permanently affect the system in any way, such as writing any information to the user's machine or erasing any information.

From a technical perspective, this sandbox is implemented by a layer of modules that operate at different levels.

Language Layer

The language layer operates at the lowest layer of the Java language model and has certain features that facilitate the implementation of the security model at the higher levels.

Memory Protection

Java code cannot write beyond array boundaries or otherwise corrupt memory.

Access Protection

Unlike C++, Java enforces language-level access controls such as private classes or methods.

Bytecode Verifier

When a Java applet is compiled, it is compiled all the way down to the platform-independent Java bytecode where the code is verified before it is allowed to run. The function of bytecode verification is to ensure that the applet operates according to the rules set down by Java and ensures that untrusted code is snared before it can be executed.

While the language restrictions are implemented by any legal Java compiler, there is still the possibility that a malicious entity could craft its own bytecode or use a compromised compiler. To deal with this possibility, Sun Microsystems architected the Java interpreter to run any applet bytecode against a verifier program that scans the bytecode for illegal sequences. Some of the checks performed by the verifier are done statically before the applet is started. However, because the applet can dynamically load more code as it is running, the verifier also implements some checks at runtime.

The bytecode verifier is the mechanism that ensures that Java class files conform to the rules of the Java application. Although not all files are subject to bytecode verification, those that are have their memory boundaries enforced by the bytecode verifier.

Security Manager

The function of the Java security manager is to restrict the ways in which an applet uses the available interfaces, and the bulk of Java's security resources are implemented via the security manager.

At the highest level, the security manager implements an additional set of checks. The security manager is the primary interface between the core Java API and the operating system and has the responsibility for allowing or denying access to the system resources it controls.

This security manager can be customized or subclassed, which allows it to refine or change the default security policy. Changing the security manager at runtime is disallowed because an applet could possibly discover a way to install its own bogus security manager. All of the Java class libraries that deal with the file system or the network call the security manager to ensure that accesses are controlled.

From a technology perspective, the security manager is a single interface module that performs the runtime checks on potentially dangerous methods that an applet could attempt to execute.

Security Package

The security package is the mechanism that allows for the authentication of signed Java classes. Those are the classes that are specified in the java.security package.

Signed applets were introduced in version 1.1 of the Java SDK and specifically are collections of class files and their supporting files that are signed with a digital signature.

The way in which a signed applet operates is that a software developer obtains a certificate from a certificate authority (CA) and uses that certificate to sign their applications. When an end user browses a Web page the developer has signed, the browser informs the end user who signed the applet and allows the user to determine if he wants to run that applet.

Key Database

The key database works with the security manager to manage the keys used by the security manager to control access via digital signatures.

The Java Standard Applet Security Policy

The exact set of policies that are enforced by Java in a specific environment can be modified by creating a custom version of the security manager class. However, there is a standard policy that has been defined by Sun and is implemented by all Web browsers that implement Java applets. The standard policy basically states:[2]

- An applet can only connect back to its source. This means, for example, that if the applet source is outside a company firewall, the applet is only allowed to talk to a machine that is also outside the firewall.
- An applet cannot query system properties because these properties could hold important information that could be used to compromise the system or invade the user's privacy.
- An applet cannot load native libraries because native code cannot be restricted by the Java security model.
- An applet cannot add classes to system packages because it might violate some access-control restrictions.
- An applet cannot listen on socket connections. This means that an applet can connect to a network service (on its source machine), but it cannot accept connections from other machines.
- An applet cannot start another program on the client workstation. This way, an applet cannot then spawn some other program or rogue process on the workstation. From a programming perspective, an applet is not allowed to manipulate threads outside its own thread group.
- An applet cannot read or write to any files on the user's machine.
- An applet can only add threads to its own thread group.

Java Language Security

This is not the place to detail the security features of the Java programming language, but a few of its most significant security-based features include the following.

Lack of Pointer Arithmetic

Java security is extended through lack of pointer arithmetic because Java programs do not use explicit pointers. Pointers are simply memory locations in applications. Consequently, no one can program (either maliciously or accidentally) a forged pointer to memory. The mishandling of pointers is probably one of the largest sources of bugs in most programming languages. To get around the lack of pointers, all references to methods and instance variables in the Java class file are via symbolic names.

Garbage Collection

Java garbage collection is the process by which Java deallocates memory that it no longer needs. Most languages such as C and C++ simply allocate and deallocate memory on the fly. The use of garbage collection requires Java to keep track of its memory usage and to ensure that all objects are properly referenced. When objects in memory are no longer needed, the memory they use is automatically freed by the garbage collector so that it can be used for other applets. The Java garbage collection engine is a multithreaded application that runs in the background and complements the lack of memory pointers in that they prevent problems associated with bad pointers.

Compiler Checks

The Java compiler checks that all programming calls are legitimate.

E-Commerce and Java

Sun Microsystems has entered the E-commerce arena in a big way and envisions having Java at the forefront of E-commerce. To assist in that attempt, Sun has created a Java E-commerce architecture to promote it.

Components of the architecture are the Java Wallet, Commerce Client, Commerce API, and Commerce JavaBeans.

The Java Wallet is a family of products written in Java that enable secure electronic commerce operations. The Java Wallet combines the Java Commerce Client, Commerce JavaBeans components, the gateway security model, and the Java Commerce Messages to create a single platform for E-commerce. It should be noted that the components can be used independently of one another. The Java wallet is written in Java; thus, it can run in any Java-capable browser.

Threats

In *Java Security: Hostile Applets, Holes and Antidotes,* McGraw and Felten describe four classes of threats that Java is susceptible to:

1. *System modification*. This is the most severe class of threats where an applet can significantly damage the system on which it runs. Although this threat is the most severe, the defenses Java has to defend against it are extremely strong.
2. *Invasion of privacy*. This is the type of attack where private information about host, file, or user is disclosed. Java defends against this type of attack rather well because it monitors file access and applets can only write back to the channel in which they were originally opened.
3. *Denial of service*. Denial-of-service attacks are written to deny users legitimate access to system resources. Denial-of-service attacks take many forms, but are primarily applications or malicious applets that take more processes or memory allocation area than they should use, such as filling up a file system or allocating all of a system's memory. Denial-of-service attacks are the most commonly encountered Java security concern and, unfortunately, Java has a weak defense against them.
4. *Antagonism*. An antagonistic threat is one in which the applet simply annoys the user, such as by playing an unwanted sound file or displaying an undesired image. Many antagonistic attacks are simply programming errors. Most denial-of-service attacks can be classified as antagonistic threats, but the ones defined here are less annoying than their denial-of-service counterpart. Like their counterpart, Java has a weak defense against them.

Using Java Securely

By following some generic guidelines, and then customizing those guidelines for an environment's unique needs, Java can be safely used in most environments. Java security, like most computer security, is built on a lot of common sense. A few of the major issues are:

- *Make sure that your browser is up to date*. Many Java vulnerabilities have originated in browser design flaws. Staying with a relatively new release of a browser hopefully ensures that discovered security flaws have been ameliorated.

- *Stay on top of security alerts.* Keep track of advisories from CERT (www.cert.org), CIAC (www.ciac.llnl.gov), and the appropriate browser vendor.
- *Think before you visit a Web site.* If visiting www.whitehouse.gov, chances of downloading a hostile Java applet are much less than if visiting www.hackers.subterfuge.org. The bottom line, use your head when surfing the Web.
- *Know your risks.* Every company must assess its risks before it can really understand how to deal with the security risks involved with Java. If the risk of Java is too great (i.e., nuclear control centers), do not use Java; if the risks are more minimal (i.e., home), one can pretty much use Java with ease.

Third-Party Software Protection

There are numerous third-party software tools available to further secure Java and add protection against the potential security threats that Java can produce. Such products are a necessity for running push and active content applications.

- Finjan — SurfinGate & SurfinGate (www.finjan.com)
- Safe Technologies — eSafe Protect (www.esafe.com)
- Digitivity — Cage (www.digitivity.com)
- Security7 — SafeGate (www.security7.com)

Conclusions About Java Security

Java has an impressive security architecture and foundation, but one cannot rely on the sandbox model exclusively. Combined with poorly written PERL and CGI scripts, browser vulnerabilities, operating system holes, Web server holes, and more, there are plenty of potential openings in which a malicious or poorly written application could wreak havoc.

Knowing what one's risks are, combined with an understanding of Java's vulnerabilities and active protection of content, will prove that *Java security* is not an oxymoron.

Notes

1. An applet is defined as a Java program that is run from inside a Web browser. The html page loaded into the Web browser contains an <applet> tag, which tells the browser where to find the Java .class files. For example, the URL http://cnn.com/TECH/computing/JavaNews.html starts a Java applet in the browser window because the source code contains the entry <applet code=Ticker.class>.
2. This article cannot list all of the details of the standard policy. For a thorough listing, view the Java SDK documentation set.

References

Frequently Asked Questions — Java Security, http://java.sun.com/sfaq/index.html.

Under Lock and Key: Java Security for the Networked Enterprise, http://java.sun.com/features/1998/01/security.html.

The Java Commerce FAQ, http://java.sun.com/products/commerce/faq.html.

The Gateway Security Model in the Java Commerce Client, http://java.sun.com/products/commerce/docs/whitepapers/security/gateway.pdf.

Low Level Security in Java by Frank Yellin, http://www.javasoft.com/sfaq/verifier.html

THE RAID ADVANTAGE

Tyson Heyn

INSIDE

The Solution to Server Gridlock and Data Integrity, RAID Elements, The Array of RAID Levels, Interface Options

INTRODUCTION

Electronic data processing evolved from virtually nothing 50 years ago to its virtual omnipresence in the industrialized societies of the world today. The technologies that have been harnessed to manipulate data converted to its lowest common denominators (zeros and ones) have made a huge impact on the lives of people throughout the world. Digitized information, or data, is being used to enable everything from live conversations between continents via satellite, to the advancement of scientific discoveries and research, to controlling the temperatures of different rooms in a home. The recently emerged raft of online services provides not only the links to communicate with personal computers, but provides access to oceans of information to navigate, capture, and use by anyone with a computer. Businesses like banks and credit card companies use massive computing systems to provide everyday conveniences like easier and faster access to money, in turn making it easier to bill or manage accounts. Even supermarkets and retail department stores are using powerful, data-intensive information systems to do everything from managing inventories to monitoring consumer spending habits. The applications list goes on and on; everyone in virtually every walk of life is exposed in some manner or form to the impact of the ongoing revolution called the Information Age.

The engines behind this revolution, of course, are computers. Today's Pentium-class personal computers, RISC workstations, minicomputers, supercomputers, and even (still!) mainframes provide the power that drives this infinite mass of data that is relied on to make everything from bank transactions to the purchase of groceries as easy as possible. The flow of data between computers,

PAYOFF IDEA

Redundant arrays of independent disks (RAID) presents a solution to the problem of providing access to gigabytes of data to users quickly and reliably.

whether networked or linked via online services or the Internet, has become nothing less than a raging flood.

This astounding volume of data being transmitted between systems today has created an obvious need for data management. As a result, more and more servers — whether they are PCs, UNIX workstations, minicomputers, or supercomputers — have assumed the role of information or data traffic cops. The number of networked or connectable systems is increasing by leaps and bounds as well, thanks to the widespread adoption of the client/server computing model, the boom in home computer use, and the rise of Internet access service providers.

Hard disk storage plays an important role in enabling improvements to networked systems, because the vast and growing ocean of data has to reside somewhere. It also has to be readily accessible, placing a demand on storage system manufacturers to not only provide high-capacity products, but also products that can access data as fast as possible and to as many people at the same time as possible. Such storage also has to be secure, placing an importance on reliability features that best ensure that data will never be lost or otherwise rendered inaccessible to network system users.

RAID: THE SOLUTION TO SERVER GRIDLOCK AND DATA INTEGRITY

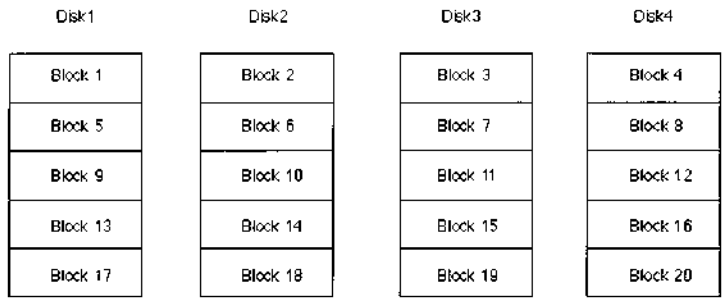
The solution to providing access to many gigabytes of data to users fast and reliably has been to assemble a number of drives together in a gang or array of disks. These are known as RAID subsystems, which stands for redundant arrays of independent disks. Simple RAID subsystems (Exhibit 1) are basically a clutch of up to five or six disk drives assembled in a cabinet and connected to a single controller board. The RAID controller orchestrates read and write activities in the same way a controller for a single disk drive does, and treats the array as if it were in fact a single or virtual drive. RAID management software that resides in the host system provides the means to manage data to be stored on the RAID subsystem.

RAID ELEMENTS

Despite its multidrive configuration, RAID subsystems disk drives remain hidden from users. The subsystem itself is the virtual drive, although it can be as large as 1000 Gbytes. The phantom virtual drive is created at a lower level within the host operating system through the RAID management software. Not only does the software set up the system to address the RAID unit as if it were a single drive, but it allows the subsystem to be configured in ways that best suit the general needs of the host system.

RAID subsystems can be optimized for performance, the highest capacity, fault tolerance, or a combination of two or three of these. Different so-called RAID levels have been defined and standardized in accordance with those general optimization parameters. There are six

EXHIBIT 1 — A Simple RAID Subsystem



such standardized levels of RAID, called RAID 0, 1, 2, 3, 4, or 5, depending on performance, redundancy, and other attributes required by the host system. The RAID software that is used to configure the desired RAID level of features in an array is described in more detail in the following paragraphs.

The RAID controller board is the hardware element that serves as the backbone for the array of disks. It not only relays the input/output (I/O) commands to specific drives in the array, but provides the physical link to each of the independent drives so they may easily be removed or replaced. The controller also serves to monitor the health or integrity of each drive in the array to anticipate the need to move data should it be placed in jeopardy by a faulty or failing disk drive. This feature is known as fault tolerance.

THE ARRAY OF RAID LEVELS

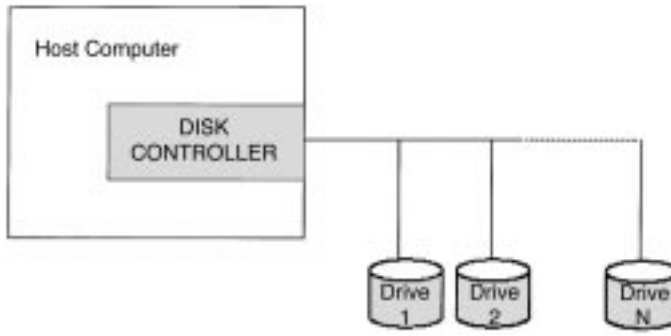
The RAID 1 through 5 standards offer users and system administrators a host of configuration options. These options allow the arrays to be tailored to their application environments. Each of the various configurations listed in the following paragraphs focuses on maximizing the abilities of an array in one or more of the following areas: capacity, data availability, performance, and fault tolerance.

RAID Level 0

An array configured to RAID Level 0 is an array optimized for performance, but at the expense of fault tolerance or data integrity.

RAID Level 0 is achieved through a method known as striping. The collection of drives (or virtual drive) in a RAID Level 0 array has data laid down in such a way that it is organized in stripes across the multiple drives. A typical array can contain any number of stripes, usually in mul-

EXHIBIT 2 — In a RAID Level 0 configuration, a virtual drive comprises several stripes of information. Each consecutive stripe is located on the next drive in the chain, evenly distributed over the number of drives in the array.



tiples of the number of drives present in the array. As an example, imagine a four-drive array configured with 12 stripes (four stripes of designated space per drive). Stripes 0, 1, 2, and 3 would be located on corresponding hard drives 0, 1, 2, and 3. Stripe 4, however, appears on a segment of drive 0 in a different location than Stripe 0; Stripes 5 through 7 appear accordingly on drives 1, 2, and 3. The remaining four stripes are allocated in the same even fashion across the same drives, such that data would be organized in the manner depicted in Exhibit 2. Practically any number of stripes can be created on a given RAID subsystem for any number of drives; 200 stripes on two disk drives is just as feasible as 50 stripes across 50 hard drives. Most RAID subsystems, however, tend to have between 3 and 10 stripes.

The reason RAID Level 0 is a performance-enhancing configuration is that striping enables the array to access data from multiple drives at the same time. In other words, because the data is spread out across a number of drives in the array, it can be accessed faster because its not bottled up on a single drive. This is especially beneficial for retrieving very large files, because they can be spread out effectively across multiple drives and accessed as if they were the size of any of the fragments they are organized into on the data stripes.

The downside to RAID Level 0 configurations is that it sacrifices fault tolerance, raising the risk of data loss because no room is made available to store redundant data. If one of the drives in the RAID 0 fails for any reason, there is no way of retrieving the lost data, as can be done in the following RAID implementations.

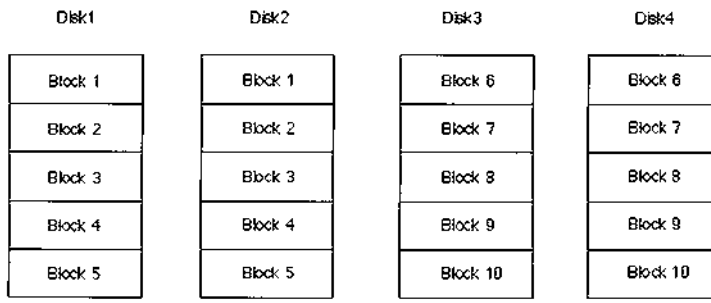
RAID Level 1

The RAID Level 1 configuration employs what is known as disk mirroring, which is done to ensure data reliability or a high degree of fault tolerance. RAID Level 1 also enhances read performance, but the improved performance and fault tolerance come at the expense of available capacity in the drives used.

In a RAID Level 1 configuration, the RAID management software instructs the subsystems controller to store data redundantly across a number of the drives (mirrored set) in the array. In other words, the same data is copied and stored on different disks (or mirrored) to ensure that, should a drive fail, the data is available somewhere else within the array. In fact, all but one of the drives in a mirrored set could fail and the data stored to the RAID Level 1 subsystem would remain intact. A RAID Level 1 configuration can consist of multiple mirrored sets, whereby each mirrored set can be a different capacity. Usually the drives making up a mirrored set are of the same capacity. If drives within a mirrored set are of different capacities, the capacity of a mirrored set within the RAID Level 1 subsystem is limited to the capacity of the smallest-capacity drive in the set; hence, the sacrifice of available capacity across multiple drives.

The read performance gain can be realized if the redundant data is distributed evenly on all of the drives of a mirrored set within the subsystem. The number of read requests and the total wait state times both drop significantly, in inverse proportion to the number of hard drives in the RAID, in fact. To illustrate, suppose three read requests are made to the RAID Level 1 subsystem (see Exhibit 3). The first request looks for data in the first block of the virtual drive; the second request goes to block 2, and the third seeks from block 3. The host-resident RAID man-

EXHIBIT 3 — A RAID Level 1 subsystem provides high data reliability by replicating (or mirroring) data between physical hard drives. In addition, I/O performance is boosted as the RAID management software allocates simultaneous read requests between several drives.



agement software can assign each read request to an individual drive. Each request is then sent to the various drives, and now — rather than having to handle the flow of each data stream one at a time — the controller can send three data streams almost simultaneously, which in turn reduces system overhead.

RAID Level 2

RAID Level 2 is rarely used in commercial applications, but is another means of ensuring data is protected in the event drives in the subsystem incur problems or otherwise fail. This level builds fault tolerance around Hamming error correction code (ECC), which is often used in modems and solid-state memory devices as a means of maintaining data integrity. ECC tabulates the numerical values of data stored on specific blocks in the virtual drive using a special formula that yields what is known as a checksum. The checksum is then appended to the end of the data block for verification of data integrity when needed.

As data gets read back from the drive, ECC tabulations are again computed, and specific data block checksums are read and compared against the most recent tabulations. If the numbers match, the data is intact; if there is a discrepancy, the lost data can be recalculated using the first or earlier checksum as a reference point.

The following example shows one method of ECC. Suppose the phrase being stored is HELLOTHERE. The checksum is computed for every 10 bytes of data.

Data being stored	H	E	L	L	O	T	H	E	R	E	
Numerical representation	72	69	76	76	79	84	72	69	82	69	
Checksum formula	× 1	× 2	× 3	× 4	× 5	× 6	× 7	× 8	x 9	x 10	
Multiplied out	72	138	228	304	395	504	504	414	738	690	
[Check]sum of all values	72	+138	+228	+304	+395	+504	+504	+414	+738	+690	= 3987

So, the data is stored on the drive as 72 69 76 76 79 84 72 69 82 69 3987.

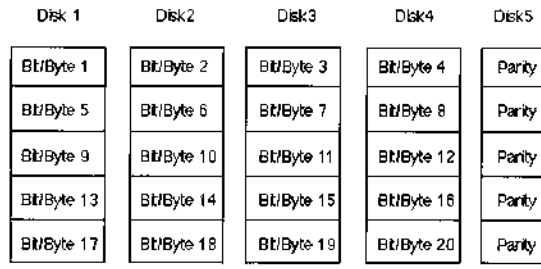
As the data is read back from the drive, the same calculations with the data segment are made. The newly computed checksum is compared against the previously stored checksum, thus verifying data integrity.

This form of ECC is actually different from the ECC technologies employed within the drives themselves. The topological formats for storing data in a RAID Level 2 array is somewhat limited, however, compared with the capabilities of other RAID implementations, which is the reason it is not often used in commercial applications.

RAID Level 3

This RAID level is really an adaptation of RAID Level 0 that sacrifices some capacity, for the same number of drives, but achieves a high level

EXHIBIT 4 — A RAID Level 3 configuration is very similar to a RAID Level 0 configuration in its utilization of data stripes dispersed over a series of hard drives to store data. In addition to these data stripes, a special drive is configured to hold parity information used to maintain data integrity throughout the RAID subsystem.



of data integrity or fault tolerance. It takes advantage of RAID Level 0 data-striping methods, except that data is striped across all but one of the drives in the array. This drive is used to store parity information that is used to maintain data integrity across all drives in the subsystem. The parity drive itself is divided up into stripes, and each parity drive stripe is used to store parity information for the corresponding data stripes dispersed throughout the array. This method achieves very high data transfer performance by reading from or writing to all of the drives in parallel or simultaneously, but retains the means to reconstruct data if a given drive fails, maintaining data integrity for the system (see Exhibit 4). RAID Level 3 is an excellent configuration for moving very large sequential files in a timely manner.

The stripes of parity information stored on the dedicated drive are calculated using the Exclusive OR (XOR) function. XOR is a logical function between the two series that carries most of the same attributes as the conventional OR function. The difference occurs when the two bits in the function are both nonzero: in XOR, the result of the function is zero, whereas with conventional OR it would be one, as described in Table 1.

By using XOR with a series of data stripes in the RAID, any lost data can easily be recovered. Should a drive in the array fail, the missing information can be determined in a manner similar to solving for a single variable in an equation (for example, solving for x in the equation, $4 + x = 7$). Similarly, in an XOR operation, it would be an equation like $1\ x = 1$. Thanks to XOR, there is always only one possible solution (in this case, 0), which provides a complete error recovery algorithm in a minimum amount of storage space.

TABLE 1 — Standard OR Function: Group A Group B

Group A	Group B	Result
0	0	0
1	0	1
0	1	1
1	1	1

RAID Level 4

This level of RAID is similar in concept to RAID Level 3, but emphasizes performance for different applications, e.g., database transaction processing versus large sequential files. Another difference between the two is that RAID Level 4 has a larger stripe depth, usually of two blocks, which allows the RAID management software to operate the disks much more independently than RAID Level 3, which controls the disks in unison. This essentially replaces the high data throughput capability of RAID Level 3 with faster data access in read-intensive applications.

A shortcoming of RAID Level 4 is rooted in an inherent bottleneck on the parity drive. As data gets written to the array, the parity-encoding scheme tends to be more tedious in write activities than with other RAID topologies. This more or less relegates RAID Level 4 to read-intensive applications with little need for similar write performance. As a consequence, like its Level 3 cousin, it does not see much common use in commercial applications.

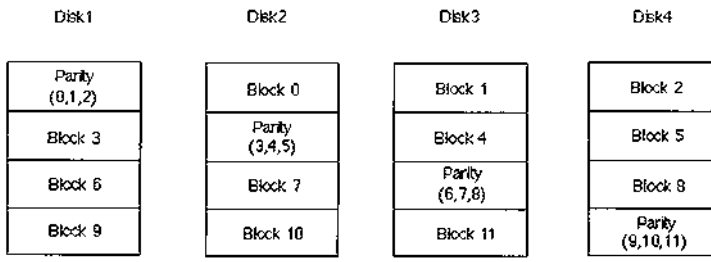
RAID Level 5

This is the last of the most commonly used RAID levels, and is probably the most frequently implemented. RAID Level 5 minimizes the write bottlenecks of RAID Level 4 by distributing parity stripes over a series of hard drives. In doing so it provides relief to the concentration of write activity on a single drive, which in turn enhances overall system performance (see Exhibit 5).

The way RAID Level 5 reduces parity write bottlenecks is relatively simple. Instead of allowing any one drive in the array to assume the risk of a bottleneck, all of the drives in the array assume write activity responsibilities. The distribution frees up the concentration on a single drive, improving overall subsystem throughput.

The RAID Level 5 parity-encoding scheme is the same as Levels 3 and 4. It maintains the ability of the system to recover any lost data should a single drive fail. This can happen as long as no parity stripe on an individual drive stores the information of a data stripe on the same drive. In

EXHIBIT 5 — RAID Level 5 overcomes the RAID Level 4 write bottleneck by distributing parity stripes over two or more drives within the system. This better allocates write activity over the RAID drive members, thus enhancing system performance.



other words, the parity information for any data stripe must always be located on a drive other than the one on which the data resides.

Other RAID Levels

Other, less-common RAID levels have been developed as custom solutions by independent vendors (they are not established standards):

- RAID Level 6, which emphasizes ultrahigh data integrity;
- RAID Level 10 (also known as RAID Levels 0 and 1), which focuses on high I/O performance and very high data integrity; and
- RAID Level 53, which combines RAID Levels 0 and 3 for uniform read and write performance.

Tailormade RAID

Perhaps the biggest advantage of RAID technology is the sheer number of possible adaptations available to users and systems designers. RAID offers the ability to customize an array subsystem to the requirements of its environment and the applications demanded of it. The inherent variety of configuration options of RAID provides several ways in which to satisfy specific application requirements (see Table 2). Customization, however, does not stop with a RAID level. Drive models, capacities, and performance levels have to be factored in, as well as what connectivity options are available.

INTERFACE OPTIONS

Differential SCSI (small computer systems interface), for example, allows a subsystem to be cabled as far as 18 feet from a host with no degrada-

TABLE 2 — RAID Configuration Options

RAID Level	Capacity	Data Availability	Data Throughput	Data Integrity
0	High	Read/write high	High I/O transfer rate	
1		Read/write high		Mirrored
2	High		High I/O transfer rate	ECC
3	High		High I/O transfer rate	Parity
4	High	Read high		Parity
5	High	Read/write high		Parity
6		Read/write high		Double parity
10		Read/write high	High I/O transfer rate	mirrored
53			High I/O transfer rate	Parity

tion to the data signal. Fast/Wide SCSI, another interface option, can be combined with differential SCSI or employed by itself; it essentially doubles the 10 Mbyte/s throughput of Fast SCSI, enabling data rates of up to 20 Mbytes/s. The newest parallel SCSI interface option is UltraSCSI, a 40 Mbyte/s interface standard.

An emerging new serial interface standard known as Fibre Channel-Arbitrated Loop (FC-AL) is yet another interface option for RAID subsystems, and is the most powerful of them all. FC-AL is capable of up to 200 Mbyte/s data throughputs (dual-loop configurations) while allowing RAID subsystems or other connected peripherals to be placed as far as 10 km from the host. It also enables easy connection of up to 126 disk drives on a single controller (compared with seven devices with conventional SCSI). The potential impact of FC-AL alone will undoubtedly be enormous on the evolution of RAID subsystems. FC-AL can be operated in either single- or dual-loop configurations. The dual loop allows another level of redundancy by allowing two separate data paths for all attached devices.

SCA: CLEANING UP THE CABLE MESS

Many of these interface options, including serial FC-AL and parallel UltraSCSI, support the SCSI Single Connector Attachment (SCA) standard. SCA is an elegant means of eliminating the miles of wiring involved with connecting several drives via conventional backplane architectures. Before SCA, conventional connections involved two cables per drive: one for power and the other for data transmission. Arrays with more than a few drives would amass a lot of spaghetti at the rear of the rack, and especially large arrays would have an unwieldy mess of wire to connect the drives. SCA, however, allows for drives to be plugged directly into a backplane without cables. It not only rids subsystems of the mass of cabling previously required, but facilitates hot plugging (removal or inser-

tion of a drive while the subsystem is online) and improves the reliability of the system as a whole because of the substantially reduced number of connections.

Tyson Heyn is Product Communications Manager at Seagate Technology, Inc. He specializes in high-end disk drives and technologies.

Malicious Code: The Threat, Detection, and Protection

Ralph Hoefelmeyer, CISSP
Theresa E. Phillips, CISSP

Malicious code is logically very similar to known biological attack mechanisms. This analogy is critical; like the evolution of biological mechanisms, malicious code attack mechanisms depend on the accretion of information over time. The speed of information flow in the Internet is phenomenally faster than biological methods, so the security threat changes on a daily if not hourly basis.

One glaring issue in the security world is the unwillingness of security professionals to discuss malicious code in open forums. This leads to the hacker/cracker, law enforcement, and the anti-virus vendor communities having knowledge of attack vectors, targets, and methods of prevention; but it leaves the security professional ignorant of the threat. Trusting vendors or law enforcement to provide information on the threats is problematic and is certainly not due diligence. Having observed this, one must stress that, while there is an ethical obligation to publicize the potential threat, especially to the vendor, and observe an embargo to allow for fixes to be made, exploit code should *never* be promulgated in open forums.

Macro and script attacks are occurring at the rate of 500 to 600 a month. In 2001, Code Red and Nimda caused billions of dollars of damage globally in remediation costs. The anti-virus firm McAfee.com claims that the effectiveness of the new wave of malicious codes was due to a one-two punch of traditional virus attributes combined with hacking techniques. Industry has dubbed this new wave of attacks *the hybrid threat*.

Exhibit 32-1. Viruses, 1986–2001.

Virus	First Observed	Type
Brain	1986	.com infector
Lehigh	1987	Command.com infector
Dark Avenger	1989	.exe infector
Michelangelo	1991	Boot sector
Tequila	1991	Polymorphic, multipartite file infector
Virus Creation Laboratory	1992	A virus builder kit; allowed non-programmers to create viruses from standard templates
Sme.g.,pathogen	1994	Hard drive deletion
Wm.concept	1995	Macro virus
Chernobyl	1998	Flash BIOS rewrite
Explore.zip	1999	File erasure
Magistr	2001	E-mail worm; randomly selects files to attach and mail

The goals in this chapter are to educate the information security practitioner on the current threat environment, future threats, and preventive measures.

CURRENT THREATS

Viruses

The classic definition of a virus is a program that can infect other programs with a copy of the virus. These are binary analogues of biological viruses. When these viruses insert themselves into a program — the program being analogous to a biological cell — they subvert the control mechanisms of the program to create copies of themselves. Viruses are not distinct programs — they cannot run on their own and need to have some host program, of which they are a part, executed to activate them. Fred Cohen clarified the meaning of *virus* in 1987 when he defined a virus as "a program that can 'infect' other programs by modifying them to include a possibly evolved copy of itself." Cohen earned a Ph.D. proving that it was impossible to create an accurate virus-checking program.

One item to note on viruses is the difference between damage as opposed to infection. A system may be infected with a virus, but this infection may not necessarily cause damage. Infected e-mail that has viral attachments that have not been run are referred to as *latent viruses*.

Exhibit 32-1 describes some examples of viruses released over the years. (Note: This is not an exhaustive list — there are arguably 60,000 known viruses.)

Worms

Worms are independent, self-replicating programs that spread from machine to machine across network connections, leveraging some network medium — e-mail, network shares, etc. Worms may have portions of themselves running on many different machines. Worms do not change other programs, although they may carry other code that does (e.g., a virus). Worms illustrate attacks against availability, where other weapons may attack integrity of data or compromise confidentiality. They can deny legitimate users access to systems by overwhelming those systems. With the advent of the *blended threat* worm, worm developers are building distributed attack and remote-control tools into the worms. Worms are currently the greatest threat to the Internet.

Morris Worm. Created by Robert T. Morris, Jr. in 1988, the Morris worm was the first active Internet worm that required no human intervention to spread. It attacked multiple types of machines, exploited several vulnerabilities (including a buffer overflow in fingered and debugging routines in *sendmail*), and used multiple streams of execution to improve its speed of propagation. The worm was intended to be a proof of concept; however, due to a bug in the code, it kept reinfecting already infected machines, eventually overloading them. The heavy load crashed the infected systems, resulting in the worm's detection. It managed to infect some 6200 computers — 10 percent of the Internet at that time — in a matter of hours. As a result of creating and unleashing this disruptive worm, Morris became the first person convicted under the Computer Fraud and Abuse Act.

Code Red Worm. The Code Red worm infected more than 360,000 computers across the globe on July 19, 2001. This action took less than 14 hours. The intention of the author of Code Red was to flood the White House with a DDoS attack. The attack failed, but it still managed to cause significant outages for other parties with infected systems. This worm used the ida and idq IIS vulnerabilities. The patch to correct this known vulnerability had been out for weeks prior to the release of the worm.

Nimda. Nimda also exploited multimode operations: it was an e-mail worm, it attacked old bugs in Explorer and Outlook, and it spread through Windows shares and an old buffer overflow in IIS. It also imitated Code Red 2 by scanning logically adjacent IP addresses. The net result was a highly virulent, highly effective worm that revealed that exploiting several old bugs can be effective, even if each hole is patched on most machines: all patches must be installed and vulnerabilities closed to stop a Nimda-like worm. Such a worm is also somewhat easier to write because one can use many well-known exploits to get wide distribution instead of discovering new attacks.

Exhibit 32-2. Trojan horses and payloads.

Trojan Horse	"Legitimate" Program	Trojan
PrettyPark	Screen Saver	Auto e-mailer; tries to connect to specific IRC channel to receive commands from attacker
Back Orifice	Program	Allows intruders to gain full access to the system
Goner	Screen Saver	Deletes AV files; installs DDoS client
W32.DIDer	Lottery game "ClickTilUWin"	Transmits personal data to a Web address

Trojan Horses

A Trojan horse, like the eponymous statue, is a program that masquerades as a legitimate application while containing another program or block of undesired, malicious, destructive code, deliberately disguised and intentionally hidden in the block of desirable code. The Trojan Horse program is not a virus but a vehicle within which viruses may be concealed. Exhibit 32-2 lists some Trojan horses, their distribution means, and payloads.

Operating System-Specific Viruses

DOS. DOS viruses are checked for by current anti-virus software. They are a threat to older machines and systems that are still DOS capable. DOS viruses typically affect either the command.com file, other executable files, or the boot sector. These viruses spread by floppy disks as well as e-mail. They are a negligible threat in today's environment.

Windows. Macro viruses take advantage of macros — commands that are embedded in files and run automatically. Word-processing and spreadsheet programs use small executables called macros; a macro virus is a macro program that can copy itself and spread from one file to another. If you open a file that contains a macro virus, the virus copies itself into the application's start-up files. The computer is now infected. When you next open a file using the same application, the virus infects that file. If your computer is on a network, the infection can spread rapidly; when you send an infected file to someone else, they can also become infected.

Visual Basic Script (VBS) is often referred to as *Virus Builder Script*. It was a primary method of infection via e-mail attachments. Now, many network or system administrators block these attachments at the firewall or mail server.

UNIX/Linux/BSD. UNIX, Linux, and BSD were not frequently targeted by malicious code writers. This changed in 2001, with new Linux worms target-

ing systems by exploiting flaws in daemons that automatically perform network operations. Examples are the Linux/Lion, which exploits an error in the bind program code and allows for a buffer overflow. Another example of a UNIX worm is SadMind. This worm uses a buffer overflow in Sun Solaris to infect the target system. It searches the local network for other Solaris servers, and it also searches for Microsoft IIS servers to infect and deface. Many of the UNIX variant exploits also attempt to download more malicious code from an FTP server to further corrupt the target system. The goal of UNIX attacks involves placing a root kit on the target system; these are typically social engineering attacks, where a user is induced to run a Trojan, which subverts system programs such as *login*.

Macintosh. Main attack avenues are bootable Macintosh disks, HyperCard stacks, and scripts. An example is the Scores virus, first detected in early 1988. This virus targeted EDS and contained code to search for the code words *ERIC* and *VULT*. It was later ascertained that these were references to internal EDS projects. This is notable in that this is the first example of a virus targeting a particular company. Scores infected applications and then scanned for the code words on the target system. Resources that were so identified were terminated or crashed when they were run. As cross-platform attacks become more common, Macintosh platforms will become increasingly vulnerable.

Cross-Platform. An example of cross-platform malicious code is the Lindose/Winux virus. This virus can infect both Linux Elf and Windows PE executables. Many installations of Linux are installed on dual-boot systems, where the system has a Linux partition and a Windows partition, making this a particularly effective attack mechanism.

Other attacks target applications that span multiple platforms, such as browsers. A good source of information on cross-platform vulnerabilities is http://www.sans.org/newlook/digests/SAC/cross.htm.

Polymorphic Viruses

Virus creators keep up with the state-of-the-art in antiviral technology and improve their malicious technology to avoid detection. Because the order in which instructions are executed can sometimes be changed without changing the ultimate result, the order of instructions in a virus may be changed to bypass the anti-virus signature. Another method is to randomly insert null operations instructions to the computer, mutating the sequence of instructions the anti-virus software recognizes as malicious. Such changes result in viruses that are polymorphic — they constantly change the structural characteristics that would have enabled their detection.

Script Attacks

Java and JavaScript. Java-based attacks exploit flaws in the implementation of Java classes in an application. A known early attack was the BrownOrifice applet. This applet exploited flaws in Netscape's Java class libraries.

JavaScript has been used in the Coolnow-A worm to exploit vulnerabilities in Microsoft Internet Explorer.

ActiveX. ActiveX controls have more capabilities than tools that run strictly in a sandbox. Because ActiveX controls are native code that run directly on a physical machine, they are capable of accessing services and resources that are not available to code that runs in a restricted environment. There are a few examples of ActiveX attack code as of this writing. There is example code called Exploder, which crashed Windows 95 systems. There is also a virus, the HTML.bother.3180, that uses ActiveX controls to perform malicious activity on the target system.

FUTURE THREATS: WHO WILL WRITE THEM?

The Script Kiddie Threat

There are automated hacking tools on the Internet, readily available at many hacker sites. These tools are of the point-and-click genre, requiring little to no programming knowledge. The security practitioner must visit these hacker sites to understand the current threat environment. Fair warning: these sites often have attack scripts, and many hackers use pornography to prevent or limit official perusal of their sites by legitimate authorities. The script kiddies are a serious threat due to their numbers. The recent *goner* worm was the work of three teenagers in Israel; other malicious code has been created by untrained people in Brazil, Finland, and China.

Criminal Enterprises

The amount of commerce moving to the Internet is phenomenal, in the multibillion-dollar range. Wherever there are large transactions, or high transaction volumes, the criminal element will attempt to gain financial advantage. Malicious code introduced by criminals may attempt to gain corporate financial information, intellectual property, passwords, access to critical systems, and personnel information. Their goals may be industrial espionage, simple theft by causing goods and services to be misdelivered, fraud, or identity theft.

Ideologues

Small groups of ideologues may use the Internet and malicious code to punish, hinder, or destroy the operations of groups or governments they find objectionable. Examples are the anti-WTO groups, which have

engaged in hacking WTO systems in Europe, and various anti-abortion groups in the United States. Also, individual citizens may take action, as recently seen in the Chinese fighter striking the American surveillance plane; many Chinese citizens, with tacit government approval, have launched attacks on American sites.

Terrorist Groups

Terrorist groups differ from ideologues in that they are generally better funded, better trained, and want to destroy some target. Since September 11, the seriousness of the terrorist threat cannot be stressed enough. The goals of a terrorist group may be to use malicious code to place root kits on systems responsible for dam control, electrical utilities' load balancing, or nuclear power plants. A speedy propagating worm, such as the Warhol, would be devastating if not quickly contained. Additionally, terrorist groups may use malicious code to manipulate financial markets in their favor; attacked companies may lose stock value over a short time, allowing for puts and calls to be made with foreknowledge of events.

Terrorists generally fall into two categories: (1) well-educated and dedicated, and (2) highly motivated Third- or Fourth-World peasants. An example of the first would be the Bader Meinhoff group; for the second, the Tamil Tigers of Sri Lanka.

Government Agencies

The Internet has allowed many government and corporate entities to place their functions and information to be readily accessible from the network. The flip side of this is that, logically, one can "touch" a site from anywhere in the world. This also means that one can launch attacks using malicious code from anywhere on the planet.

Intelligence agencies and military forces have already recognized that the Internet is another battlefield. The U.S. National Security Agency, FBI, and U.K. MI5 and MI6 all evince strong interest in Internet security issues. The U.S. Air Force has in place a cyber-warfare center at Peterson Air Force Base, Colorado Springs, Colorado. Its Web site is http://www.spacecom.af.mil/usspacecom/jtf-cno.htm. Note that their stated mission is:

> Subject to the authority and direction of USCINCSPACE, JTF-CNO will, in conjunction with the unified commands, services and DoD agencies, coordinate and direct the defense of DoD computer systems and networks; coordinate and, when directed, conduct computer network attack in support of CINCs and national objectives.

The intelligence and military attackers will be well-educated professionals with the financial and technical backing of nation-states. Their attacks will not fail because of bad coding.

Warhol

Nimda was the start of multiple avenues and methods of attack. After Code Red, researchers began to investigate more efficient propagation or infection methods. One hypothetical method is described in a paper by Nicholas Weaver of the University of California, Berkeley; the paper can be obtained at http://www.cs.berkeley.edu/~nweaver/warhol.html. Weaver named this attack methodology the *Warhol Worm*. There are several factors affecting malicious code propagation: the efficiency of target selection, the speed of infection, and the availability of targets. The Warhol method first builds a list of potentially vulnerable systems with high-speed Internet connections. It then infects these target systems because they are in the best position to propagate the malicious code to other systems. The newly infected system then receives a portion of the target list from the infecting system. Computer simulations by Weaver indicate that propagation rates across the Internet could reach one million computers in eight minutes. His initial assumptions were to start with a 10,000-member list of potentially vulnerable systems; the infecting system could perform 100 scans per second; and infecting a target system required one second.

Cross-Platform Attacks: Common Cross-Platform Applications

A very real danger is the monoculture of applications and operating systems (OS) across the Internet. Identified flaws in MS Windows are the targets of malicious code writers. Applications that span platforms, such as MS Word, are subject to macro attacks that will execute regardless of the underlying platform; such scripts may contain logic to allow for cross-platform virulence.

Intelligent Scripts

These scripts detect the hardware and software on the target platform, and they have different attack methods scripted specifically for a given platform/OS combination. Such scripts can be coded in Java, Perl, and HTML. We have not seen an XML malicious code attack method to date; it is really only a matter of time.

Self-Evolving Malicious Code

Self-evolving malicious code will use artificial neural networks (ANNs) and genetic algorithms (GA) in malicious code reconstruction. These platforms will change their core structures and attack methods in response to the environment and the defenses encountered. We see some of this in Nimda, where multiple attack venues are used. Now add an intelligence capability to the malicious code, where the code actively seeks information on new vulnerabilities; an example would be scanning the Microsoft patch site for patches, creation of exploits that take advantage of these

patch fixes, and release of the exploit. These will have far larger payloads than current attacks and may require a home server site for evolution. As networks evolve, these exploits may *live* in the network.

The development of distributed computing has led to the idea of *parasitic computing*. This model would allow the intelligent code to use the resources of several systems to analyze the threat environment using the distributed computing model. The parasitic model also allows exploits to steal cycles from the system owner for whatever purpose the exploit builder desires to use them for, such as breaking encryption keys.

Router Viruses or Worms

Attack of routers and switches is of great concern; successful cross-platform attacks on these devices could propagate across the Internet in a manner akin to the aforementioned Warhol worm.

Analysis of Formal Protocol Description. This attack method requires a formal analysis of the protocol standard and the various algorithms used to implement the protocol. We have seen an example of this with the SNMP v1 vulnerability, released publicly in February 2002. The flaw is not in the protocol but in the implementation of the protocol in various applications.

Further research of protocols such as the Border Gateway Protocol (BGP), Enhanced Interior Gateway Routing Protocol (EIGRP), testing the implementation versus the specification, may lead to other vulnerabilities.

Test against Target Implementations. The malicious code builders simply gain access to the target routing platform and the most prevalent version of the routing software and proceed to test various attack methods until they succeed. Also, with privileged access to a system, attackers may reverse-engineer the implementation of the target protocols underlying software instance. An analysis of the resulting code may show flaws in the logic or data paths of the code.

The primary target of router attacks will be the BGP. This protocol translates routing tables from different vendors' routing platforms for interoperability. It is ubiquitous across the Internet. By targeting ISPs' routers, the attackers can potentially take down significant portions of the Internet, effectively dropping traffic into a black hole. Other methods use packet-flooding attacks to effect denial-of-service to the network serviced by the router. Router or switch operating system vulnerabilities are also targeted, especially because these network devices tend not to be monitored as closely as firewalls, Web servers, or critical application servers.

Wireless Viruses

Phage is the first virus to be discovered that infects hand-held devices running the PalmOS. There were no confirmed reports of users being

affected by the virus, and it is considered a very low threat. It overwrites all installed applications on a PalmOS handheld device.

Wireless phones are another high-risk platform. An example is the Short Messaging Service (SMS) exploit, where one sends malformed data headers to the target GSM phone from an SMS client on a PC, which can crash the phone.

In June of 2001, the Japanese I-mode phones were the targets of an e-mail that caused all I-mode phones to dial 110, the Japanese equivalent of 911. Flaws in the software allowed embedded code in the e-mail to be executed.

The growing wireless market is sure to be a target for malicious code writers. Additionally, the software in these mobile devices is not implemented with security foremost in the minds of the developers, and the actual infrastructures are less than robust.

Active Content

Active content, such as self-extracting files that then execute, will be a great danger in the future. The security and Internet communities have come to regard some files as safe, unlike executable files. Many organizations used Adobe PDF files instead of Microsoft Word, because Adobe was perceived as safe. We now see exploits in PDF files. Additionally, there is now a virus, SWF/LFM-926, which infects Macromedia Flash files.

PROTECTION

Defense-in-Depth

A comprehensive strategy to combat malicious code encompasses protection from, and response to, the variety of attacks, avenues of attack, and attackers enumerated above. Many companies cocoon themselves in secure shells, mistakenly believing that a perimeter firewall and anti-virus software provide adequate protection against malicious code. Only when their systems are brought to a halt by a blended threat such as the Code Red worm do they recognize that, once malicious code penetrates the first line of defense, there is nothing to stop its spread throughout the internal network and back out to the Internet. Malicious code has multiple ways to enter the corporate network: e-mail, Web traffic, instant messenger services, Internet chat (IRC), FTP, handheld devices, cell phones, file sharing programs such as Napster, peer-to-peer programs such as NetMeeting, and unprotected file shares through any method by which files can be transferred. Therefore, a sound protection strategy against malicious code infiltration requires multiple overlapping approaches that address the people, policies, technologies, and operational processes of information systems.

Exhibit 32-3. Safe computing practices for the Windows user community.

1. Install anti-virus software. Make sure the software is set to run automatically when the system is started, and do not disable real-time protection.
2. Keep anti-virus software up-to-date. Configure systems to automatically download updated signature files from the company-approved server or vendor site on a regular basis.
3. Install the latest operating system and application security patches.
4. Do not share folders or volumes with other users. If drive sharing is necessary, do not share the full drive and do password-protect the share with a strong password.
5. Make file extensions visible. Windows runs with the default option to "hide file extensions for known file types." Multiple e-mail viruses have exploited hidden file extensions; the VBS/LoveLetter worm contained an e-mail attachment, a malicious VBS script, named "LOVE-LETTER-FOR-YOU.TXT.vbs; the .vbs extension was hidden.
6. Do not forward or distribute non-job-related material (jokes, animations, screen savers, greeting cards).
7. Do not activate unsolicited e-mail attachments and do not follow the Web links quoted in advertisements.
8. Do not accept unsolicited file transfers from strangers in online peer-to-peer computing programs such as Instant Messaging or IRC.
9. Beware of virus hoaxes. Do not forward these messages, and do not follow the instructions contained therein.
10. Protect against infection from macro viruses:
    If Microsoft Word is used, write-protect the global template.
    Consider disabling macros in MS Office applications through document security settings.
    Consider using alternate document formats such as rtf (Rich Text Format) that do not incorporate executable content such as macros.
11. Check ALL attachments with anti-virus software before launching them. Scan floppy disks, CDs, DVDs, Zip disks, and any other removable media before using them.
12. Turn off automatic opening of e-mail attachments or use another mail client. BadTrans spread through Microsoft Internet Explorer-based clients by exploiting a vulnerability in auto-execution of embedded MIME types.
13. Establish a regular backup schedule for important data and programs and adhere to it.

Policy

An organization's first step in the battle against malicious code is the development and implementation of a security policy addressing the threat to information systems and resources (see Exhibit 32-3). The policy describes proactive measures the organization has taken to prevent infection; safe computing rules and prevention procedures that users must follow; tools and techniques to implement and enforce the rules; how to recognize and report incidents; who will deal with an outbreak; and the consequences of noncompliance. The policy should make employees assume responsibility and accountability for the maintenance of their computers.

When users understand why procedures and policies are implemented, and what can happen if they are not followed, there tends to be a higher level of compliance.

Suggested Policy Areas. Require the use of company-provided, up-to-date anti-virus software on all computing devices that access the corporate network, including handheld and wireless devices. Inform users that removing or disabling protection is a policy violation. Address remote and mobile Windows users by specifying that they must have up-to-date protection in order to connect to the network. Consider establishing virus protection policies for guest users, such as vendors and consultants, and for protecting Linux, UNIX, and Macintosh operating systems as well.

Weaknesses in software programs are routinely discovered and exploited; therefore, a sound anti-virus policy must address how and when patching will be done, as well as the means and frequency for conducting backups.

The information security practitioner needs to recognize that users with Web-based e-mail accounts can circumvent the carefully constructed layers of protection at the firewall, e-mail gateway, and desktop by browsing to a Web-based e-mail server. Policy against using external e-mail systems is one way to prevent this vector, but it must be backed up with an HTTP content filter and firewall rules to block e-mail traffic from all but approved servers or sources.

Finally, include a section in the policy about virus warnings. Example: "Do not forward virus warnings of any kind to *anyone* other than the *incident handling/response team.* A virus warning that comes from any other source should be ignored."

Education and Awareness

Security policy must be backed up with awareness and education programs that teach users about existing threats, explain how to recognize suspicious activity, and how to protect the organization and their systems from infection. The information security practitioner must provide the user community with safe computing practices to follow, and supply both the tools (e.g., anti-virus software) and techniques (e.g., automatic updates) to protect their systems.

Awareness training must include the social engineering aspects of viruses. The AnnaKournikova and NakedWife viruses, for example, took advantage of human curiosity to propagate; and communications-enabled worms spread via screen savers or attachments from known correspondents whose systems had been infected.

The awareness program should reiterate policy on how to recognize and deal with virus hoaxes. E-mail hoaxes are common and can be as costly in terms of time and money as the real thing. Tell users that if they do forward the "notify everyone you know" warnings to all their colleagues, it can create a strain on mail servers and make them crash — having the same effect as the real thing.

Protection from Malicious Active Code

Protect against potentially malicious scripts by teaching users how to configure their Internet browsers for security by disabling or limiting automatic activation of Java or ActiveX applets. Teach users how to disable Windows Scripting Host and to disable scripting features in e-mail programs — many e-mail programs use the same code as Web browsers to display HTML; therefore, vulnerabilities that affect ActiveX, Java, and JavaScript are often applicable to e-mail as well as Web pages.

System and Application Protection. Consider using alternative applications and operating systems that are less vulnerable to common attacks. The use of the same operating system at the desktop or in servers allows one exploit to compromise an entire enterprise. Similarly, because virus writers often develop and test code on their home computers, corporate use of technologies and applications that are also popular with home users increases the threat to the corporation from malicious code designed to exploit those applications. If trained support staff is available in-house, the organization may decide to run services such as DNS, e-mail, and Web servers on different operating systems or on virtual systems. With this approach, an attack on one operating system will have less chance of affecting the entire network.

Regardless of which operating system or application is used, it is critical to keep them up-to-date with the latest security patches. Worms use known vulnerabilities in the OS or application to take over systems. Frequently, vendors have released patches months in advance of the first exploitation of a weakness. Rather than being in the reactive mode of many system administrators who were caught by the Code Red worm, be proactive about testing and applying patches as soon as possible after receiving notification from the vendor. Use scripts or other tools to harden the operating system and disable all unnecessary services. Worms have taken advantage of default installations of Web server and OS software.

Layered Anti-virus Protection. Because malicious code can enter the enterprise through multiple avenues, it is imperative that protective controls be applied at multiple levels throughout the enterprise. In the time prior to macro viruses, there was little benefit to be gained by using anti-virus controls anywhere but the desktop. However, when macro viruses

became prevalent, placing controls at the file server helped reduce infection. In today's environment of communication-enabled worms and viruses, a thorough protection strategy involves integrated anti-virus solutions at the desktop, file and application servers, groupware servers, and Internet e-mail gateway and firewall; and inspection of all traffic flowing between the external gateway and internal network.

Protect the Desktop. Desktop protection remains a crucial component of an effective protection strategy. The information security practitioner must ensure that the organization has an enterprise license for anti-virus software, along with a procedure to automate installation and updates. Anti-virus software should be part of the standard build for desktops, laptops, and workstations, backed up by policy that makes it a violation to disable or uninstall the real-time scanning. It is prudent to give remote users a license for company-approved anti-virus software to enable them to run it on their end systems, regardless of whether the company owns those nodes.

Because current viruses and worms can spread worldwide in 12 hours or less (and new ones may propagate much faster), the ability to quickly update systems during an outbreak can limit the infection. However, the heavy traffic caused by thousands or millions of users trying to simultaneously update their definition files will hamper the ability to obtain an update from the vendor's site during an outbreak. Instead, the enterprise anti-virus administrator can provide a local site for updating. The anti-virus administrator can download once from the vendor site, allowing the entire network to be updated locally. This approach avoids network congestion and reduces the risk of infection from users who are unable to obtain a timely update from the vendor.

Server Protection. Although infection via macro viruses is no longer widespread, protection for network files and print servers can prevent infection from old or infrequently used files. Regardless of policies or training, there are always some users without up-to-date anti-virus protection — whether from naïveté, deliberately disabling the software, or because of system problems that prevent the anti-virus software from starting. One unprotected system can infect many files on the network server if server-side protection is not installed.

Fortify the Gateway. The speed of infection and the multiple vectors through which malicious code can enter the enterprise provide the impetus to protect the network at the perimeter. Rather than trying to keep current on the list of ports known to be used by malicious programs, configure firewalls to use the default *deny all* approach of closing all ports and only opening those ports that are known to be needed by the business. Virus writers are aware of this approach, so they attack ports that are usually open such as HTTP, e-mail, and FTP. Because e-mail is the current method

of choice for malicious code propagation, the information security practitioner must implement gateway or network-edge protection. This protection is available as anti-virus software for a particular brand of e-mail server, as gateway SMTP systems dedicated to scanning mail before passing the messages to the corporate e-mail servers, or anti-virus and malicious code services provided by an e-mail service provider. To protect against infection via Web and FTP, gateway virus protection is available for multiple platforms. The software can scan both incoming and outgoing FTP traffic, and it scans HTTP traffic for hostile Java, JavaScript, or ActiveX applets.

Protect the Routing Infrastructure. As companies learn to patch their systems, block certain attachments, and deploy malicious code-detection software at the gateway, attackers will turn to other vectors. As mentioned earlier, routers are attractive targets because they are more a part of the network infrastructure than computer systems; and they are often less protected by security policy and monitoring technology than computer systems, enabling intruders to operate with less chance of discovery.

To protect these devices, practice common-sense security: change the default passwords, set up logging to an external log server, use AAA with a remote server, or require access through SSH or VPNs.

Vulnerability Scans. A proactive security program includes running periodic vulnerability scans on systems; results of the scans can alert the information security practitioner to uninstalled patches or security updates, suddenly opened ports, and other vulnerabilities. System administrators can proactively apply patches and other system changes to close identified vulnerabilities before they are exploited by attackers using the same tools. There are a number of commercial and open-source scanning tools, such as SATAN, SAINT, and Nessus.

Handhelds. As IP-enabled handhelds such as PDAs, palmtops, and smart phones become more popular, they will be targeted by attackers. To keep these computing devices from infecting the network, provide a standard anti-virus software package for mobile devices and instruct users on how to download updates and how to run anti-virus software when synching their handheld with their PC.

Personal Firewalls. Personal firewalls offer another layer of protection for users, especially for remote users. Properly configured personal firewalls can monitor both incoming and outgoing traffic, detect intrusions, block ports, and provide application (e-mail, Web, chat) controls to stop malicious code. The firewalls function as an agent on the desktop, intercepting and inspecting all data going into or out of the system. To facilitate enterprise management, the personal firewall software must be centrally managed so that the administrator can push policy to users, limit the ability

of users to configure the software, and check for the presence of correctly configured and active firewalls when the remote user connects to the network. The firewall logging feature should be turned on to log security — relevant events such as scans, probes, viruses detected, and to send the logs to a central server.

Research

> If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.
>
> — Sun Tzu,
> 6th-century BC Chinese general,
> Author of *The Art of War*

Knowing what direction virus development is taking, and knowing and eliminating potential vulnerabilities before they can be exploited, is one of the most positive steps an organization can take toward defense. Virus creators keep up with the state-of-the-art in antiviral technology and improve their malicious technology to avoid detection. The information security practitioner must do likewise. Monitor hacker and black-hat sites (follow precautions listed earlier) to keep abreast of the threat environment. Visit anti-virus vendor sites: EICAR (European Institute of Computer Anti-virus Researchers), *The Virus Bulletin*, and the Wild List of viruses at www.wildlist.org. Other sources to monitor are the Honeynet Project and SecurityFocus' ARIS (Attack Registry and Intelligence Services) predictor service (fee based). These sites monitor exploits and develop statistical models that can predict attacks.

DETECTION AND RESPONSE

Virus and Vulnerability Notification

Monitor sites such as BugTraq and SecurityFocus that publish vulnerability and malicious code information. Subscribe to mailing lists, alert services, and newsgroups to be notified of security patches. Subscribe to alerts from anti-virus vendors, organizations such as SANS, Carnegie Mellon's CERT, NIPC (National Infrastructure Protection Center), Mitre's CVE (Common Vulnerabilities and Exposures), and BugTraq. Monitor the anti-virus vendor sites and alerts for information about hoaxes as well, and proactively notify end users about hoaxes before they start flooding the corporate e-mail server.

Anti-virus (AV) software vendors rely on customers and rival AV companies for information on the latest threats. Typically, if a corporation thinks that an as-yet unidentified virus is loose on its network, it sends a sample

to the AV vendor to be analyzed. This sample is then passed on to other AV vendors so that all work in concert to identify the virus and develop signature updates. This cooperative effort ensures that end users receive timely protection, regardless of which AV vendor is used.

Virus researchers also spend time visiting underground virus writing sites where some authors choose to post their latest code. This allows AV companies to work to develop methods to detect any new techniques or potential threats before they are released.

Current Methods for Detecting Malicious Code

The propagation rate of malware attacks is rapidly reaching the point of exceeding human ability for meaningful reaction. The Code Red and Nimda worms were virulent indicators of the speed with which simple active worms can spread. By the time humans detected their presence, through firewall probes or monitoring of IP ranges, the worms had spread almost worldwide.

Signature Scanning. Signature scanning, the most common technique for virus detection, relies on pattern-matching methods. This technique searches for an identifiable sequence or string in suspect files or traffic samples and uses this virus fingerprint or *signature* to detect infection. While this method is acceptable for detecting file and macro viruses or scripts that require activation to spread, it is not very effective against worms or polymorphic viruses. This reactive method also allows a new virus a window of opportunity between the initial appearance of the virus and the time it takes for the industry to analyze the threat, determine the virus signature, and rush to deploy updates to detect the signature.

The response time to worm outbreaks is shrinking to a few hours. Worms can spread faster than virus updates can be created. Even faster infection strategies have been postulated, such as the Warhol worm and Flash worms, which theoretically may allow a worm to infect all vulnerable machines in minutes. Firewall and anti-virus development must move in the direction of detecting and automatically responding to new attacks.

Client or Desktop AV to Detect and Remove Viral Code. Client AV programs can detect and often disinfect viruses, and they must provide both on-access and static virus checking. Static file scanning checks a file or file volume for viruses; on-access, real-time virus checking scans files before they are fully opened. Suspect files are treated according to configurable rules — they may be repaired, disinfected, quarantined for later treatment, or deleted.

Anti-virus software generally uses virus signatures to recognize virus threats. Most viruses that arrive via e-mail have been released within the

previous year or more recently; therefore, virus software containing old signatures is essentially useless. It is vital to ensure that virus software is updated on a regular basis — weekly at a minimum for desktops. To ensure that desktop protection is up-to-date, the information security practitioner should provide an automated update mechanism. The client software can be configured to periodically check for new AV signatures and automatically install them on the desktop. Desktop anti-virus software must be able to scan compressed and encoded formats to detect viruses buried in multiple levels of compression.

Because laptops and notebooks are frequently used without being connected to the network, when an unprotected machine attaches to the network, some mechanism needs to be in place to detect the connection and force either the installation or update of anti-virus software, or force the computer to disconnect. Another way to check a laptop system is to run a vulnerability scan each time a remote desktop authenticates to the network in order to ensure it has not already been compromised. Many of the enterprise Code Red infections occurred not through Internet-facing MS Internet Information Services (IIS) servers but through infected notebook computers or systems connecting via VPNs. Once Code Red enters the internal network, it infects unpatched systems running IIS, although those systems were inaccessible from the Internet.

Recently, anti-virus vendors have recommended that companies update their virus software every day instead of weekly. With the arrival of viruses such as Nimda, some customers pull software updates every hour.

Besides detection through technology, user observation is another way to detect worm activity. The "goner" worm disabled personal firewall and anti-virus software; users should recognize this, if through no other means than by missing icons in their Windows system tray, and notify the incident handling team.

Server Detection. Server administrators must regularly review their system and application logs for evidence of viral or Trojan activity, such as new user accounts and new files, (rootkits or root.exe in the scripts directory), and remove these files and accounts. Remove worm files and Trojans using updated virus scanners to detect their presence. Discovery of *warez* directories on FTP servers is proof that systems have been compromised. Performance of real-time anti-virus scanners may impact servers; not all files need to be scanned, but at a minimum critical files should be scanned. Server performance monitoring will also provide evidence of infection, either through reduced performance or denial of service.

File Integrity Checkers. File integrity tools are useful for determining if any files have been modified on a system. These tools help protect systems against computer viruses and do not require updated signature files. When

an integrity checker is installed, it creates a database of checksums for a set of files. The integrity checker can determine if files have been modified by comparing the current checksum to the checksum it recorded when it was last run. If the checksums do not match, then the file has been modified in some manner. Some integrity checkers may be able to identify the virus that modified a file, but others may only alert that a change exists.

Real-Time Content Filtering. To prevent the entry of malicious code into the corporate network, implement content filtering at the gateways for Web, mail, and FTP traffic. Set the filters to block known vulnerable attachments at the gateway. Filter attachments that have been delivery vehicles for malicious code, such as .exe, .com, .vbs, .scr, .shs, .bat, .cmd, .com, .dll, .hlp, .pif, .hta, .js, .lnk, .reg, .vbe, .vbs, .wsf, .wsh, and .wsc. Inform users that if they are trying to receive one of these files for legitimate purposes, they can have the sender rename the extension when they send the attachment. Many worms use double extensions, so block attachments with double extensions (e.g., .doc.vbs or .bmp.exe.) at the gateway or firewall.

At the initial stages of an infection, when new signatures are not available, block attachments or quarantine e-mails that contain certain words in the subject line or text until the anti-virus vendor has a signature update.

E-mail and HTML filtering products can examine file attachments and HTML pages. Objects such as executable files or code can be stripped out before passing them on, or they can be quarantined for later inspection. Deploy software that performs real-time virus detection and cleanup for all SMTP, HTTP, and FTP Internet traffic at the gateway. SMTP protection complements the mail server to scan all inbound and outbound SMTP traffic for viruses.

Set up scanning rules on the gateway SMTP system to optimize scanning of incoming e-mail. Some systems scan attachments only, and others scan both attachments and e-mail text — this distinction is important because some viruses, such as BubbleBoy, can infect without existing as an attachment. Be aware of the capabilities of the system selected. As with desktop software, gateway systems provide options to scan all attachments or only selected attachments. Handling viruses is tunable as well — the attachment can be deleted, repair can be attempted, or it can be logged and forwarded. Files with suspect viruses can be quarantined until new updates are received, and repair can be attempted at that time.

HTTP protection keeps infected files from being downloaded and allows the information security practitioner to set uniform, system-wide security standards for Java and Authenticode. It also affords protection against malicious Java and ActiveX programs for users. FTP protection works to ensure that infected files are not downloaded from unsecured remote sites.

Proactive Detection

Detecting Anomalous Activity: Sandboxing and Heuristics. Sandboxing is a proactive technique that works by monitoring the behavior of certain attachments in real-time, blocking malicious content from running before it can negatively impact a system. It essentially places a barrier in front of the operating system resources and lets the barrier determine which access programs and applications have to operating system resources. Programs are classified as low, medium, or high restricted, and system resources' access controls are assigned accordingly. An anti-virus package is still required to identify and disinfect known malicious code, but the threat is removed regardless of whether the anti-virus system reacts.

Heuristic scanning uses an algorithm to determine whether a file is performing unauthorized activities, such as writing to the system registry or activating its own built-in e-mail program. Both sandboxing and heuristic techniques at the desktop can be useful as the final layer of defense. Both examine the behavior of executed code to attempt to identify potentially harmful actions, and they flag the user for action should such behavior be identified. Because behavior-blocking tools do not need to be updated with signatures, layering traditional anti-virus solutions with these proactive solutions can create an effective approach to block both known and new malicious code. The drawback to both methods is the tendency to generate false positives; to get their work done, users often end up saying *yes* to everything, thus defeating the protection.

Worm Detection: Firewalls and Intrusion Detection Systems (IDSs). Hybrid firewalls (those that combine application proxies with stateful inspection technologies) can be used effectively to repel blended threats such as Code Red and Nimda. Application inspection technology analyzes HTTP and other protocol requests and responses to ensure they adhere to RFC standards.

Worms can also be detected by their excessive scanning activity — network monitoring on the LAN should send alerts to the network operations staff when unusual scanning activity is detected, whether the activity is generated externally or internally. Monitoring the network for normal activity will allow operators to set thresholds and trip alarms when those thresholds are exceeded. A number of machines suddenly scanning all its neighbors should send an alarm in fairly short order.

A network IDS that combines heuristics and signature technologies can provide monitoring staff with the first indication of a worm infection by identifying anomalous network traffic with known worm signatures or unusual traffic patterns. The alert still requires analysis by humans to determine if it is malicious, but such systems can provide early warning of potential infection. Many modern firewalls and IDS systems have the ability

to detect certain types of virus and worm attacks such as Code Red and Nimda, alert network support personnel, and immediately drop the connection. Some intelligent routing and switching equipment also comes with the ability to foil certain types of attacks.

Deploy IDS at the network level to detect malicious code that passes the firewall on allowed ports. The information security practitioner should also consider deploying IDS on subnets that house critical servers and services to detect malicious code activity, such as unusual scanning activity or mailing patterns. Have alerts sent when unusual traffic is logged to or from your e-mail server; the LoveLetter e-mail virus, for example, sent out 100 infected e-mails per minute from one user. Possible responses to these communication-enabled viruses include blocking e-mail with the suspect subject line, automatically (based on thresholds) blocking the victim's outbound mail queue, and contacting both the victim and the sender to notify them of the infection.

Tarpits. Tarpits such as LaBrea are a proactive method used to prevent worms from spreading. A tarpit installed on a network seeks blocks of unused IP addresses and uses them to create virtual machines. When a worm hits one of the virtual machines, LaBrea responds and keeps the worm connected indefinitely, preventing it from continuing to scan and infect other systems.

RESPONSE AND CLEANUP

If it appears that a system or network is under attack by a worm, it is prudent to sever the network connection immediately in order to isolate the local network. If the worm is already loose in the system, this act may limit its spread and may also prevent important data from being sent outside of the local area network. It may be appropriate to take the system offline until the breach has been repaired and any necessary patches installed. Critical servers should have backup systems that can be installed while the infected machine is rebuilt with fresh media.

Worms seldom attack single systems, so the incident response team will need to inspect all systems on the network to determine if they have been affected. With expanding use of extranets for customers and partners, and as Web services proliferate, responding to an intrusion or worm may involve contacting partners or customers who could lose their access to services or be compromised themselves. Such notification should be detailed in escalation procedures and incident response plans.

Incident Response and Disaster Recovery Plans

It is imperative that the information security practitioner create and test a rapid-response plan for malicious code emergencies. Infections will happen

despite defense measures, so be prepared to wipe them out quickly. The recovery plan must include escalation levels, malicious code investigators, and repair teams equipped with the tools and techniques to recover lost data. A consistent, strong backup policy, for both users and systems administrators, is essential for restoring lost or damaged data. Ensure that backup operators or system administrators have backups of all data and software, including operating systems. If the organization is affected by a virus, infected files and programs can be replaced with clean copies. For particularly nasty viruses, worms, and remote-access Trojans, the administrator may have no choice but to reformat and rebuild — this process can be simplified using a disk-imaging program such as GHOST.

SUMMARY

Practice defense-in-depth — deploy firewalls, proxy servers, intrusion detection systems, on-demand and on-access scanners at the network gateway, mail, file and application servers, and on the desktop. Employ proactive techniques such as integrity checkers, vulnerability scans, e-mail filters, behavior blockers, and tarpits to protect against incursions by malicious code. All of these tools and techniques must enforce a security policy and be clearly laid out and explained in procedures. The enterprise is complex, with many operating systems and applications running simultaneously. To address this complexity, protection must be multi-layered — controlling all nodes, data transmission channels, and data storage areas. Expect that new vulnerabilities will emerge at least as fast as old ones are repaired, and that attackers will take advantage of any that are not yet repaired.

To fight malicious code, enterprises must take a holistic approach to protection. Every aspect of the enterprise should be examined for ways to reduce the impact of malicious code and allow the organization to fight infection in a coordinated fashion. Once effective measures are in place, the information security practitioner should maintain vigilance by researching new attack methodologies and devising strategies to deal with them. By doing this, the enterprise can remain relatively virus-free, and the end users can concentrate on the business.

References

1. F. Cohen, Trends in Computer Virus Research, http://all.net/books/integ/japan.html.
2. A. Chuvakin, Basic Security Checklist for Home and Office Users, November, 2001, http://www.securityfocus.com.
3. P. Schmehl, Holistic Enterprise Anti-Virus Protection, January, 2002, http://online.securityfocus.com/infocus/.
4. J. Martin, A Practical Guide to Enterprise Anti-Virus and Malware Prevention, August, 2001, http://www.sans.org.
5. D. Banes, How to Stay Virus, Worm, and Trojan Free — Without Anti-Virus Software, May, 2001, http://www.sans.org.

G. Hulme, Going the distance, Nov. 2001, *Information Week*.
7. R. Nichols, D. Ryan, and J. Ryan, *Defending Your Digital Assets*, McGraw-Hill, 2000.
8. G. Spafford and S. Garfinkel, *Practical UNIX and Internet Security*, 2nd ed., O'Reilly & Associates, Inc., 1996.
9. Responding to the Nimda Worm: Recommendations for Addressing Blended Threats, Symantec Enterprise Security, http://securityresponse.symantec.com.

ABOUT THE AUTHORS

Ralph S. Hoefelmeyer, CISSP, began his career as a U.S. Air Force officer and went on to defense work. He has more than 20 years of experience in operations, systems design, analysis, security, software development, and network design. Hoefelmeyer has earned a B.S. and M.S. in computer science and has one patent with several patents pending. He is currently a senior engineer with WorldCom in Colorado Springs, Colorado.

Theresa E. Phillips, CISSP, is a senior engineer with WorldCom. She has five years' experience in information security engineering, architecture, design, and policy development. Prior to that, she held management positions in not-for-profit membership organizations dealing with open systems and quality engineering. Phillips earned a B.S. in social work, which provides her with the background to deal with people and policy issues related to information security.

Domain 5
Cryptography

The Cryptography Domain addresses the principles, means, and methods of disguising information to ensure its integrity, confidentiality, and authenticity. Unlike the other domains, Cryptography does not support the standard of availability.

The professional should fully understand the basic concepts within cryptography, including public and private key algorithms in terms of their applications and uses. Cryptography algorithm construction, key distribution, key management, and methods of attack are also important for the successful candidate to understand. The applications, construction, and use of digital signatures are discussed and compared to the elements of cryptography. The principles of authenticity of electronic transactions and non-repudiation are also included in this domain.

Contents

Section 5.3 Private Key Algorithms

Principles and Applications of Cryptographic Key Management
William Hugh Murray, CISSP

Section 5.4 Public Key Infrastructure (PKI)

Getting Started with PKI
Harry DeMaio

Mitigating E-Business Security Risks: Public Key Infrastructures in the Real World
Douglas C. Merrill and Eran Feigenbaum

Preserving Public Key Hierarchy
Geoffrey C. Grabow, CISSP

PKI Registration
Alex Golod, CISSP

Section 5.5 System Architecture for Implementing Cryptographic Functions

Implementing Kerberos in Distributed Systems
Joe Kovara, CTP and Ray Kaplan, CISSP, CISA, CISM

Section 5.6 Methods of Attack

Methods of Attacking and Defending Cryptosystems
Joost Houwen, CISSP

108

Three New Models for the Application of Cryptography

Jay Heiser, CISSP

Applying encryption is not easy. False confidence placed in improperly applied security mechanisms can leave an organization at greater risk than before the flawed encryption project was started. It is also possible to err in the opposite direction. Overbuilt security systems cost too much money upfront, and the ongoing expense from unneeded maintenance and lost productivity continues forever. To help avoid costly misapplications of security technology, this chapter provides guidance in matching encryption implementations to security requirements. It assumes a basic understanding of cryptological concepts, and is intended for security officers, programmers, network integrators, system managers, Web architects, and other technology decisionmakers involved with the creation of secure systems.

Introduction

The growing reliance on the Internet is increasing the demand for well-informed staff capable of building and managing security architectures. It is not just E-commerce that is generating a demand for encryption expertise. Personal e-mail needs to be protected, and employees demand secure remote access to their offices. Corporations hope to increase their productivity — without increasing risk — by electronically linking themselves to their business partners. Unfortunately, eager technologists have a tendency to purchase and install security products without fully understanding how those products address their security exposures. Because of the high cost of a security failure, requirements analysis and careful planning are crucial to the success of a system that relies on cryptological services. This chapter presents four models, each providing a different understanding of the effective application of encryption technology. Descriptive models like these are devices that isolate salient aspects of the systems being analyzed. By artificially simplifying complex reality, the insight they provide helps match security and application requirements to appropriate encryption-based security architectures.

The first model analyzes how encryption implementations accommodate the needs of the encrypted data's recipient. The relationship between the encrypter and the decrypter has significant ramifications, both for the choice of technology and for the cryptographic services used. The second model describes how encryption applications differ based on their logical network layer. The choice of available encryption services varies from network layer to network layer. Somewhat less obviously, choice of network layer also affects who within the organization controls the encryption process. The third encryption application model is topological. It illustrates concepts usually described with terms such as end-to-end, host-to-host, and link-to-link. It provides an understanding of the scope of protection that can be provided when different devices within the network topology perform encryption. The final model is based on the operational state of data. The number of operational states in which data is cryptographically protected varies, depending on the form of encryption service chosen.

Business Analysis

Before these descriptive models can be successfully applied, the data security requirements must be analyzed and defined. One of the classic disputes within the information security community is over the accuracy of quantitative risk analysis. Neither side disagrees that some form of numeric risk analysis providing an absolute measure of security posture would be desirable, and neither side disputes that choice of security countermeasures would be facilitated by an accounting analysis that could provide return on investment or at least a break-even analysis. The issue of contention is whether it is actually possible to quantify security implementations, given that human behavior can be quite random and very little data is available. Whether or not an individual prefers a quantitative or a qualitative analysis, some form of analysis must be performed to provide guidance on the appropriate resource expenditure for security countermeasures. It is not the purpose of this chapter to introduce the subject of risk management; however, a security architecture can only be optimized when the developer has a clear understanding of the security requirements of the data that requires protection.

The Data Criticality Matrix is helpful in comprehending and prioritizing an organization's information asset security categories. Exhibit 108.1 shows an example analysis for a corporation. This matrix includes five security requirements. The widely used CIA requirements of confidentiality, integrity, and availability are supplemented with two additional requirements: non-repudiation and time. The term "non-repudiation" refers to the ability to prevent the denial of a transaction. If a firm submits a purchase order but then refuses to honor the purchase, claiming no knowledge of the original transaction, then the firm has repudiated it. In addition to privacy services, cryptography may be required to provide non-repudiation services. The models in this chapter illustrate encryption options that include both services. The time requirements for data protection are important both in choosing appropriately strong encryption, and in ensuring that data is never left unprotected while it has value. This particular Data Criticality Matrix presents a simplified view of the lifetime requirement; in some cases, it may be useful to assign a specific lifetime to each of the first four security requirements, instead of assuming that confidentiality, integrity, availability, and non-repudiation all must be supported to the same degree over the same period of time. Note that availability is usually not a service provided by encryption, although encryption applications have the potential to negatively affect availability. Encryption rarely improves availability, but if mission-critical encryption services fail, then availability requirements probably will not be met. (Use of a cryptographically based strong authentication system to prevent denial-of-service attacks is an example of using encryption to increase availability.)

An economic analysis cannot be complete without an understanding of the available resources. Insufficient funds, lack of internal support, or poor staff skills can prevent the successful achievement of any project. While the four models can be used to develop an ideal security architecture, they can also facilitate an understanding of the security ramifications of a resource-constrained project.

Recipient Model

The choice of cryptographic function and implementation is driven by the relationship between the originator and the recipient. The recipient is the party — an individual, multiple individuals, or an organizational entity — consuming data that has cryptological services applied to it. The simplified diagram in Exhibit 108.2

EXHIBIT 108.1 Data Criticality Matrix

	Confidentiality	Integrity	Availability	Non-Repudiation	Lifetime
Public Web page	Low	High	High	Low	NA
Unreleased earnings data	High	High	Medium	Low	2 weeks
Accounts receivable	High	High	Medium	High	5 years
Employee medical records	High	High	Low	Low	80 years

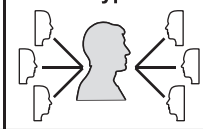
	Personal Encryption	Workgroup Encryption	Transaction Encryption
			
Recipient	Data Owner	Co-workers	Strangers
Concern	Privacy	Privacy	Establishment and Maintenance of Trust
Technical Concerns	Speed and Transparency	Speed and Transparency	Interoperability

EXHIBIT 108.2 Recipient model.

presents three possible choices of recipient. In reality, the recipient model is a spectrum, encompassing everything from a well-known recipient to a recipient with no prior relationship to the originator. The recipient model provides guidance when choosing between an open-standard product and a closed, proprietary product, and it provides insight into the specific cryptographic services that will be needed.

Personal encryption is the use of cryptographic services on an individual basis, without the expectation of maintaining cryptographic protection when sharing data. Someone encrypting data for his own personal use has completely different priorities than someone encrypting data for others. In most cases, that someone is using personal encryption to maintain the confidentiality of data that is at risk of being physically accessed — especially on a laptop. In a corporate setting, personal encryption is legitimately used on workstations to provide an additional level of privacy beyond what can be provided by an operating environment's access controls — which can always be circumvented by an administrator. Personal encryption might also be used by corporate employees trying to hide data they are not authorized to have, and criminals concerned about law enforcement searches also use personal encryption. Although individuals might want to use digital signature to provide assurance that they did indeed sign their own documents — especially if they have a high number of them — this use is rare. Increasingly, digital signature is used as an integrity control, even for files originated and stored locally. While few individuals currently use digital signature to protect files on their own workstation or laptop, the increasing proliferation of hostile code could make this use of personal encryption routine. Maintaining the confidentiality of personal information is the usual intent of individual encryption, but the technical concerns remain the same for any use of personal encryption. Individuals encrypting data for themselves place a high priority on speed and ease of use. Laptop users typically encrypt the entire hard drive, using an encryption product transparent to applications and almost transparent to users, requiring only the entry of a password at the start of a session. Standards and interoperability are not important for personal encryption, although use of unproven proprietary encryption algorithms should be avoided.

Workgroup encryption is the use of cryptological services to meet the confidentiality needs of a group of people who know each other personally and share data. As in the case of the individual, the group might be concerned that sensitive data is at risk of inappropriate viewing by system administrators. Encryption can even be used to implement a form of access control — everyone given a password has access to the encrypted data, but nobody else does. If the workgroup shares data but does not have a common server, encryption can help them share that data without having to rely on distributed access controls. The most significant issue with workgroup encryption is managing it. If the data has a long life, it is likely that the membership of the workgroup will change. New members need access to existing data, and members leaving the group may no longer be authorized for access after leaving. Constantly decrypting and reencrypting large amounts of data and then passing out new keys is inefficient. For a short-term project, it is feasible for group members to agree on a common secret key and use it to access sensitive data for the project duration. Groups and data with a longer life might find it easier to use an encryption system built on a session key that can be encrypted in turn with each group member's public key. Whether it is based on secret or public key encryption, workgroup

encryption is similar to personal encryption, having the advantage of not being concerned with open standards or multivendor compatibility. Interoperability is provided by choosing a single product for all group members, either a stand-alone encryption utility, an application with encryption capabilities, or an operating environment with security services. Trust is a function of organizational and group membership and personal relationships. Because all the members of a workgroup are personally acquainted, no special digital efforts need be provided to enhance the level of trust.

Transactional encryption describes the use of cryptological services to protect data between originators and recipients who do not have a personal relationship capable of providing trust. It facilitates electronic transactions between unknown parties; E-commerce and Web storefronts are completely dependent on it. While confidentiality may be important, in many transactions the ability to establish identity and prevent repudiation is even more significant. To accept a transaction, the recipient must have an appropriate level of trust that the purported sender is the actual sender. The recipient must also have an appropriate level of confidence that the sender will not deny having initiated the transaction. Likewise, the sender often requires a level of assurance that the recipient cannot later deny having accepted it. If the value of the transaction is high, some form of non-repudiation service may be necessary. Other cryptographic services that can be provided to increase the level of confidence include time stamp and digital notary service. Authentication mechanisms and non-repudiation controls are all electronic attempts to replace human assurance mechanisms that are impossible, impractical, or easily subverted in a digital world. The technical characteristic distinguishing transactional encryption from workgroup or personal encryption is the significance of interoperability. Because the parties of a transaction often belong to different organizations and may not be controlled by the same authority, proprietary products cannot be used. The parties of the transaction might be using different platforms, and might have different applications to generate and process their transactions. Transactional encryption depends on the use of standards to provide interoperability. Not only must standard encryption algorithms be used, but they must be supported with standard data formats such as PKCS #7 and X.509.

Network Layer Model

The OSI seven-layer reference model is widely used to explain the hierarchical nature of network implementations. Services operating at a specific network layer communicate with corresponding services at the same layer through a network protocol. Services within a network stack communicate with higher- and lower-level services through interprocess communication mechanisms exposed to programmers as APIs. No actual network represents a pure implementation of the OSI seven-layer model, but every network has a hierarchical set of services that are effectively a subset of that model. Encryption services can be provided in any of the seven network layers, each with its own advantages and disadvantages. Use of the seven-layer model to describe existing network protocol stacks that grew up organically is more than a little subjective. Over the years, the mapping of the Internet protocol set into the model has slowly but surely changed. Today, it is accepted that the IP layer maps to the OSI network layer, and the TCP protocol maps to the OSI transport layer, although this understanding of exact correspondence is not universal. Likewise, the assignment of specific encryption protocols and services to specific network layers is somewhat arbitrary. The importance of this model to security practitioners is in understanding of how relative position within the network hierarchy affects the characteristics of cryptographic services. As illustrated in Exhibit 108.3, the higher up within the network hierarchy encryption is applied, the more granular its ability to access objects can be. The lower down encryption is provided, the greater the number of upper-layer services that can transparently take advantage of it. Greater granularity means that upper-layer encryption can offer more cryptographic functions. Services based on digital signature can only be provided in the upper layers. A simplified version of this layered model can be used to analyze a nonnetwork environment. For the purposes of this discussion, stand-alone hosts are considered to have four layers: physical, session, presentation, and application.

The physical layer is the lowest layer, the silicon foundation upon which the entire network stack rests. Actually, providing encryption services at the physical layer is quite rare. In a network environment, several secure LANs have been developed using specialized Ethernet cards that perform encryption. Most of these systems actually operate at the data-link layer. Several specialized systems have been built for the defense and intelligence market that could possibly be considered to operate at the physical layer, but these systems are not found in the commercial market. The only common form of physical network layer encryption is spread spectrum, which scrambles transmissions across a wide range of constantly changing frequencies. Physical layer encryption products have been developed for stand-alone systems to protect the hard drive. The advantage

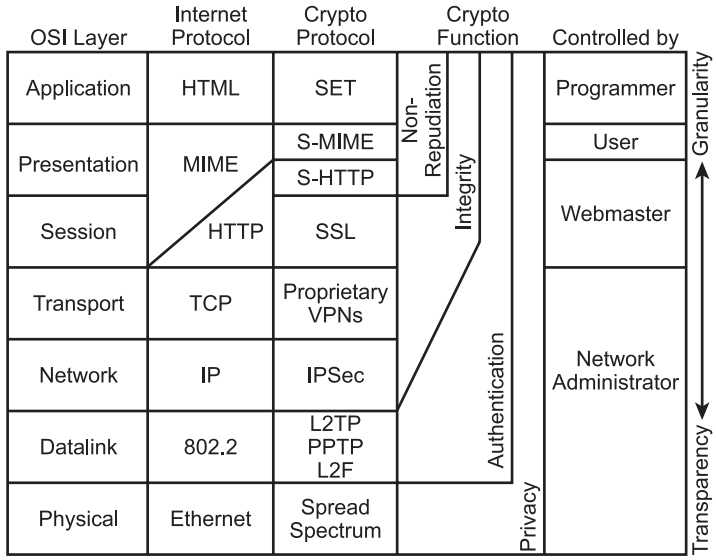


EXHIBIT 108.3 OSI model.

of such a system is that it provides very high performance and is very difficult to circumvent. Furthermore, because it mimics the standard hardware interfaces, it is completely transparent to all system software and applications. Physical layer security is under control of the hardware administrator.

A great deal of development work is being done today at both the data-link and the network layer. Because they provide interoperability between hosts and network elements, and are completely transparent to network-based applications, these two layers are used for the implementation of VPNs. The data-link layer is used to support L2TP, PPTP, and L2F. Although many popular implementations of these link layer security services actually take advantage of the IPSec transport mode, the model still treats them as link layer services because they are providing interfaces at that layer. A major advantage of the link layer is that a single encryption service can support multiple transport protocols. For example, VPNs providing an interface at this layer can support TCP/IP and SPX/IPX traffic simultaneously. Organizations running both Novell and Internet protocols can use a single VPN without having to build and maintain a protocol gateway. The IPSec security protocol resides at the network layer. It is less flexible than the link layer security services, and only supports Internet protocols, but IPSec is still transparent to applications that use TCP or UDP. Whether implemented at the network or the link layer, to be considered a VPN, the interface must be completely transparent to network services, applications, and users. The disadvantage of this complete transparency is a low level of granularity. VPNs can provide identification and authentication either at the host level, or in the case of a remote access client, at the user level. In other words, remote access users can authenticate themselves to whatever host is at the other end of their VPN. Individual files are effectively invisible to a VPN security service, and no transactional services are provided by a VPN. Many proprietary VPN and remote access products are arguably implemented at the transport layer, although no standard defines a security service at this layer. Most transport layer encryption products are actually built as a shim on top of the existing transport layer. However, because they still support the existing transport interface — usually the socket interface — they should be treated as transport layer services. These VPN products are often implemented using protocols operating at higher network layers. Upper-layer security services such as SSH and SOCKS are robust and proven, making them useful mechanisms for VPN implementations. The characteristic that determines if a security service is operating as a true VPN is not the layer at which the encryption service itself runs, but the interface layer at which security services are provided to existing upper-layer applications; whatever is running under the hood is hidden from applications and not relevant to this model. VPNs are under the administrative control of the network administrator.

The session layer is not considered relevant for standard implementations of the Internet protocols; however, the Secure Sockets Layer (SSL) service neatly fits into the definition of a session-layer service. Applications capable of using SSL must be compiled with special SSL versions of the normal socket libraries. Network services compiled with support for SSL, such as S-HTTP, listen on specific ports for connection requests by

compatible clients, such as Web browsers. SSL is still too low in the network stack to provide transactional services. In common with a VPN, it provides session-level privacy, and host-to-user or host-to-host authentication at the session start. SSL does offer a higher level of control granularity than does a VPN. Applications capable of using SSL, such as Web browsers or mail clients, usually have the ability to use it as needed by alternating between connections to standard ports and connections to secured daemons running on SSL ports. This amount of granularity is adequate for electronic commerce applications that do not require digital signature. Note that the HTML designer does have the option of specifying URLs that invoke SSL, providing that person with indirect influence over the use of SSL. Whoever has write access to the Web server is the person who has the final say over which pages are protected with SSL. Sometimes, the user is provided with a choice between SSL-enabled Web pages or unsecured pages, but giving them this option is ultimately the prerogative of the Webmaster. A stand-alone system analogy to a session-layer security service is a security service based on a file system. An encrypting file system is effectively a session-layer service. It requires initial session identification and authentication, and then performs transparently in the background as a normal file system, transparent to applications. An encrypting file system is under the control of the system administrator.

Several commonly-used Internet applications, such as Web browsers and mail clients, provide data representation services, which are presentation-layer services. Presentation-layer services operate at the granularity of an individual file. Presentation-layer file operators are not aware of application-specific data formats, but are aware of more generalized data standards, especially those for text representation. Another example is FTP, which copies individual files while simultaneously providing text conversion such as EBCDIC to ASCII. In a nonnetworked environment, any generic application that operates on files can be considered a presentation-layer service. This includes compression and file encryption utilities. Because it allows access to individual files, the presentation layer is the lowest layer that can provide transactional services, such as integrity verification and non-repudiation. Generic network services that provide digital signature of files, such as PGP, S-MIME, and file system utilities, are not operating at the application level; they are at the presentation level. Presentation services are under control of the end user. Secure HTTP (S-HTTP) is another example of a presentation-layer security service. It was intended to be used both for privacy and the digital signature of individual file objects. Secure HTTP was at one time in competition with SSL as the Web security mechanism of choice. SSL gained critical mass first, and is the only one of the two now being used. If it were available, S-HTTP would be under the control of the Webmaster, so Exhibit 108.3 represents it as being lower in the crypto protocol hierarchy than S-MIME.

Application-layer services have access to the highest level of data granularity. Accessible objects may include application-specific file formats such as word processors or spreadsheets records, or even fields within a database. Application-layer encryption is provided within an application and can only be applied to data compatible with that application. This makes application-layer encryption completely nontransparent, but it also means that application encryption can provide all cryptographic services at any needed granularity. Application-layer encryption services are normally proprietary to a specific application, although standard programming libraries are available. These include CAPI, the Java security libs, and BSAFE. Although it could arguably be considered a session-layer protocol, SET (Secure Electronic Transaction) data formats are quite specific, so it more closely resembles an application-layer protocol. It is intended to provide a complete system for electronic transaction processing, especially for credit card transactions, between merchants and financial institutions. Application-layer encryption is under the control of the programmer. In many cases, the programmer allows the user the option of selectively taking advantage of encryption services, but it is always the programmer's prerogative to make security services mandatory.

Topological Model

The topological model addresses the physical scope of a network cryptological implementation. It highlights the segments of the transmission path over which encryption is applied. Exhibit 108.4 illustrates the six most common spans of network encryption. The top half of the diagram, labeled "a," depicts an individual user on the Internet interacting with organizational servers. This user may be dialed into an ISP, be fully connected through a cable modem or DSL, or may be located within another organization's network. The bottom half of the diagram, labeled "b," depicts a user located at a partner organization or affiliated office. In case "b," the security perimeter on the left side of the diagram is a firewall. In case "a," it is the user's own PC. Note that the endpoints are always vulnerable because encryption services are always limited in their scope.

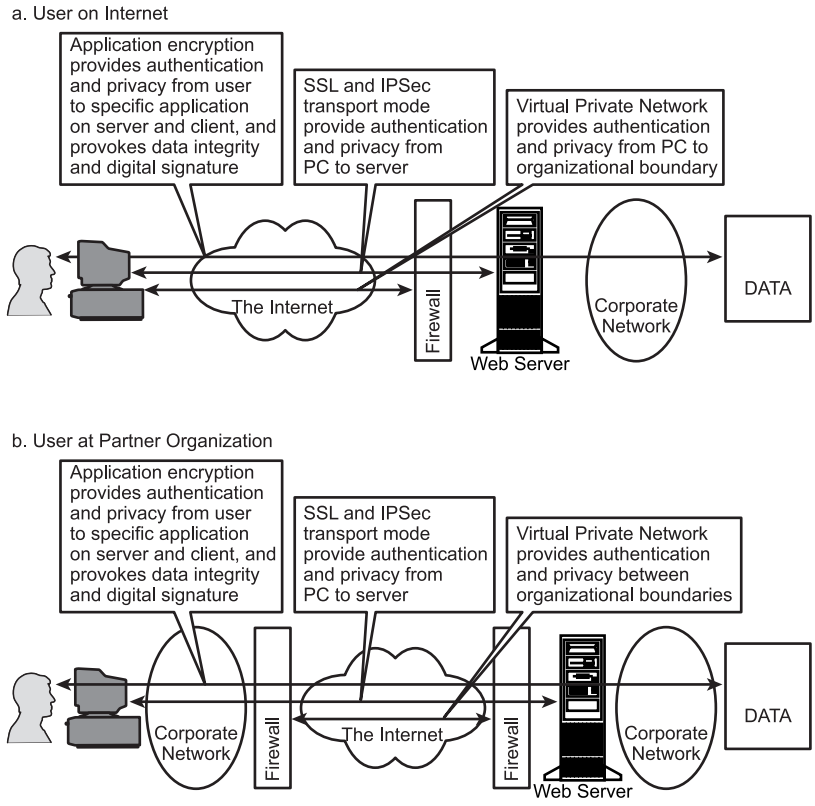


EXHIBIT 108.4 Topological model.

The term "end-to-end encryption" refers to the protection of data from the originating host all the way to the final destination host, with no unprotected transmission points. In a complex environment, end-to-end encryption is usually provided at the presentation or application-layer. The top boxes in both "a" and "b" in Exhibit 18-4 illustrate a client taking advantage of encryption services to protect data directly between the user and the data server. As shown, the data might not be located on the Web server, but might be located on another server located several hops further interior from the firewall and Web server. SSL cannot provide protection beyond the Web server, but application- or presentation-layer encryption can. Full end-to-end protection could still be provided — even in the Web environment illustrated — if both the client and data server have applications supporting the same encryption protocols. This could even take the form of a Java applet. Although the applet would be served by the Web server, it would actually run within the Java virtual machine on the Web browser, providing cryptographic services for data shared between the client and the server, protecting it over all the interior and exterior network segments.

SSL and IPSec transport mode provide authentication and privacy between a workstation and a remote server. This is a sometimes referred to as host-to-host, or node-to-node. The two middle boxes in Exhibit 108.4 represent virtually identical situations. In "b," the outgoing SSL session must transit a firewall, but it is common practice to allow this. On the server side, if the Web server is located inside a firewall, traffic on the port conventionally used by S-HTTP is allowed through the firewall to the IP address of the Web server. The SSL session provides privacy between the Web browser on the client machine and the Web server on the remote host. As in this example, if the Web server uses a back-end database instead of a local datastore, SSL cannot provide protection between the Web server and the database server (hopefully, this connection would be well protected using noncryptographic countermeasures). SSL is somewhat limited in what it can provide, but its convenience makes it the most widely implemented form of network encryption. SSL is easy to implement; virtually all Web servers provide it as a standard capability, and it requires no programming skills. It does not necessarily provide end-to-end protection, but it does protect the transmission segment that is most vulnerable to outside attack.

Another form of host-to-host encryption is the virtual private network (VPN). As shown in both cases "a" and "b" in Exhibit 108.4 at least one of the hosts in a VPN is located on an organizational security boundary. This is usually, but not always, an Internet firewall. Unlike the previous example, where "a" and "b" were functionally identical in terms of security services, in the case of a VPN, "b" is distinct from "a" in that security services are not applied over the entire transmission path. A VPN is used to create an extension of the existing organizational security perimeter beyond that which is physically controlled by the organization. VPNs do not protect transmissions within the security perimeter that they extend. The VPN architecture is probably the more common use of the term "host-to-host." The term implies that cryptographic services are provided between two hosts, at least one of which is not an endpoint. Like SSL, a VPN provides host authentication and privacy. Depending on the implementation, it may or may not include additional integrity services. As shown in case "a," VPN software is often used to support remote access users. In this scenario, the VPN represents only a temporary extension of the security perimeter. Case "b" shows an example of two remotely separated sites that are connected permanently or temporarily using a VPN. In this case, the user side represents a more complex configuration, with the user located on a network not necessarily directly contiguous with the security perimeter. In both "a" and "b," the VPN is only providing services between security perimeters.

Link-to-link encryption is not illustrated. The term refers to the use of encryption to protect a single segment between two physically contiguous nodes. It is usually a hardware device operating at layer two. Such devices are used by financial firms to protect automatic teller machine transactions. Another common form of link-to-link encryption is the secure telephone unit (STU) used by the military. The most common use of link layer encryption services on the Internet is the protection of ATM or Frame Relay circuits using high-speed hardware devices.

Information State Model

It should be clear by now that no encryption architecture provides total protection. When data undergoes transition through processing, copying, or transmission, cryptographic security may be lost. When developing a security architecture, the complete data flow must be taken into account to ensure an appropriate level of protection whenever operations are performed on critical data. As shown in Exhibit 18-5, the number of states in which data is protected varies widely, depending on the choice of encryption service. The table is sorted from top to bottom in increasing order of the number of states in which protection is provided. SSL and VPNs are at the top of the chart because they only protect during one phase: the transmission phase. In contrast, application encryption can be designed to protect data during every state but one. Although systems have been researched, no commercially available product encrypts data while it is being processed. Data undergoing processing is vulnerable in a number of ways, including:

- The human entering or reading the data can remember it or write it down.
- Information on the screen is visible to shoulder surfers.
- Virtual memory can store the cleartext data on the hard drive's swap space.
- If the process crashes, the operating system may store a core dump file containing the cleartext data.

Outside of the processing phase, which can only be addressed through administrative and physical security countermeasures, encryption options are available to protect data wherever necessary.

Several different styles of automated encryption have been developed, relieving users of the responsibility of remembering to protect their data. Some products encrypt an entire hard drive or file system, while others allow configuration of specific directories and encrypt all files placed into them. After users correctly authenticate themselves to such an encryption system, files are automatically decrypted when accessed. The downside of this automated decryption is that whenever authenticated users access a file, the encryption protection is potentially lost. Critical data can be inadvertently stored in cleartext by transmitting it, backing it up, or copying it to another system. Protection can be maintained for an encrypted file system by treating the entire file system as a single object, dumping the raw data to backup storage. Depending on the implementation, it may be impossible to perform incremental backups or restore single files. Products that do not encrypt the directory listing, leaving the names of encrypted files in cleartext, offer more flexibility, but increase the risk that an intruder can gain information about the encrypted data. If the data can be copied without automatically decrypting it, then it can be backed up or transmitted without losing cryptographic protection. Although such data would be safely encrypted on a recipient's system, it probably would not be usable because the recipient would not have a key for it. It would rarely be appropriate for someone automatically encrypting data to share

EXHIBIT 108.5 Information State Model

	Encrypted during Processing	Automatically Encrypted on First Save	Sent Data Encrypted on Originating Host	Automatically Reencrypted after Use	Encrypted when Backed Up	Encrypted during Transmission	Data Encrypted on Receiving Host
SSL						Π	
VPN						Π	
Encrypting file system that automatically decrypts		Π	Π	Π			
Encryption utility			Π		Π	Π	Π
Encrypting file system without automatic decryption		Π	Π	Π	Π	Π	No key
E-mail encryption			Π	Π	Π	Π	Π
Application with built-in data encryption		Optional	Π	Π	Π	Π	Π

the key with someone else if that key provided access to one's entire personal information store. Automatic encryption is difficult in a workgroup scenario — at best, moving data between personal storage and group storage requires decryption and reencryption with a different key.

File encryption utilities, and this includes compression utilities with an encryption option (beware of proprietary encryption algorithms), are highly flexible, allowing a file or set of files to be encrypted with a unique key and maintaining protection of that data throughout copy and transmission phases. The disadvantage of encrypting a file with a utility is that the data owner must remember to manually encrypt the data, increasing the risk that sensitive information remains unencrypted. Unless the encryption utility has the ability to invoke the appropriate application, a plaintext version of the encrypted data file will have to be stored on disk before encrypted data can be accessed. The user who decrypts it will have to remember to reencrypt it. Even if the encryption utility directly invokes an application, nothing prevents the user from bypassing automated reencryption by saving the data from within the application, leaving a decrypted copy on disk. E-mail encryption services, such as PGP and S-MIME, leave data vulnerable at the ends. Unless the data is created completely within the e-mail application and is never used outside of a mail browser, cleartext can be left on the hard drive. Mail clients that support encryption protect both the message and any attachments. If cryptographic protection is applied to an outgoing message (usually a choice of signature, encryption, or both), and outgoing messages are stored, the stored copy will be encrypted. The recipient's copy will remain encrypted too, as long as it is stored within the mail system. As soon as an attachment is saved to the file system, it is automatically decrypted and stored in cleartext. Encrypting within an application is the most reliable way to prevent inappropriate storage of cleartext data. Depending on the application, the user may have no choice but to always use encryption. When optional encryption has been applied to data, normal practice is to always maintain that encryption until a keyholder explicitly removes it. On a modern windowing workstation, a user can still defeat automated reencryption by copying the data and pasting it into another application, and application encryption is often weakened by the tendency of application vendors to choose easily breakable proprietary encryption algorithms.

Several manufacturers have created complex encryption systems that attempt to provide encryption in every state and facilitate the secure sharing of that data. Analysis of these systems will show that they use combinations of the encryption types listed in the first column of Exhibit 108.5. Such a hybrid solution potentially overcomes the disadvantages of any single encryption type while providing the advantages of several. The Information State Model is useful in analyzing both the utility and the security of such a product.

Putting the Models to Work

The successful use of encryption consists of applying it appropriately so that it provides the anticipated data protection. The models presented in this chapter are tools, helpful in the technical analysis of an encryption implementation. As shown in Exhibit 108.4, good implement choices are made by following a process. Analyzing the information security requirements is the first step. This consists of understanding what data must be protected, and its time and sensitivity requirements for confidentiality, integrity, availability, and nonrepudiation. Once the information security requirements are well documented, technical analysis can take place. Some combination of the four encryption application models should be used to develop potential implementation options. These models overlap, and they may not all be useful in every situation — choose whichever one offers the most useful insight into any particular situation. After technical analysis is complete, a final implementation choice is made by returning to business analysis. Not every implementation option may be economically feasible. The most rigorous encryption solution will probably be the most expensive. The available resources will dictate which of the implementation options are possible. Risk analysis should be applied to those choices to ensure that they are appropriately secure. If insufficient resources are available to adequately offset risk, the conclusion of the analysis should be that it is inappropriate to undertake the project. Fortunately, given the wide range of encryption solutions available for Internet implementations today, most security practitioners should be able to find a solution that meets their information security requirements and fits within their budget.

109

Auditing Cryptography: Assessing System Security

Steve Stanek

After a start-up data security firm applied for a patent for its newly developed encryption algorithm, the company issued a public challenge: it promised to pay $5000 to anyone who could break the algorithm and another $5000 to the person's favorite charity.

William Russell, an Andersen technology risk manager, accepted the challenge. He is now $5000 richer, his charity is waiting for its money, and the data security firm has run out of business because Russell cracked the supposedly uncrackable code. It took him about 60 hours of work, during which time he developed a program to predict the correct encryption key. His program cracked the code after trying 6120 out of a possible 1,208,925,819,614,629,174,706,176 electronic keys. Clearly, it should not have been as easy as that!

Assessing Risk

In the course of performing a security risk assessment, auditors or security professionals may learn that cryptographic systems were used to address business risks. However, sometimes the cryptographic systems themselves are not reviewed or assessed — potentially overlooking an area of business risk to the organization.

Russell believes there is a lesson in this for information technology auditors: when it comes to encryption technology, rely on the tried and true. "You want the company to be using well-known, well-tested algorithms," Russell says. "Never use private encryption. That goes under the assumption that someone can create something that's as good as what's on the market. The reality is that there are only a few hundred people in the world who can do it well. Everyone else is hoping nobody knows their algorithm. That's a bad assumption."

Russell recently worked with a client who asked him to look at one of the company's data systems, which was secured with encryption technology developed in-house. Russell cracked that system's security application in 11 hours. "If it had been a well-known, well-tested algorithm, something like that would not have been at all likely," Russell says.

Encryption's Number-One Problem: Keeping Keys Secret

Security professionals who use cryptography rely on two factors for the security of the information protected by the cryptographic systems: (1) the rigor of the algorithm against attack and (2) the secrecy of the key that is used to encrypt the sensitive information. Because security professionals advocate well-documented and scrutinized algorithms, they assume that the algorithm used by the cryptographic system has been compromised by an attacker; thus the security professional ultimately relies on the protection of the keys used in the algorithm.

The more information encrypted with a key, the greater the harm if that key is compromised. So it stands to reason that keys must be changed from time to time to mitigate the risk of information compromise. The

length of time a key is valid in a crypto-system is referred to as the cryptographic key period and is determined by factors such as the sensitivity of the information, the relative difficulty to "guess" the keys by a known crypto-analysis technique, and the environment in which the crypto-system functions and operates. While changing keys is important, it can be very costly, depending on the type of cryptography used, the storage media of the keying material, and the distribution mechanism of the keying material. It is a business decision on how to effectively balance security risk with cost, performance, and functionality within the business context.

Keys that can be accessed and used by attackers pose a serious security problem, and all aspects of the security program within an enterprise must be considered when addressing this issue. For example, ensure that the keys are not accessible by unauthorized individuals, that appropriate encryption is used to protect the keying material, that audit trails are maintained and protected, and that processes exist to prevent unauthorized modification of the keying material.

While cryptography is a technology subject, effective use of cryptography within a business is not just a technology issue.

Encryption's Number-One Rule

According to Mark Wilson, vice president of engineering at Embedics, a data security software and hardware design firm in Columbia, Maryland, "The No. 1 rule is that encryption needs to be based on standards. You want to follow well-known specifications for algorithms. For public key, you want to use an authenticated key agreement mechanism with associated digital signatures.

"A lot of people are trying new technologies for public key-based schemes. Most of the time they are not using published standards. They're not open to scrutiny. There are also often interoperability problems." Interoperability is important because it allows vendors to create cryptographic products that will seamlessly integrate with other applications. For example, vendors planning to develop cryptographic hardware should follow the RSA PKCS #11 standard for cryptographic hardware. If they do, then their product will work with several applications seamlessly, including Lotus Notes.

Russell and Wilson agree that even if a company is using widely tested and accepted encryption technologies, its data can be exposed to prying eyes. One Andersen client encrypted highly sensitive information using an encryption key, but the key was stored on a database that was not properly secured. Consequently, several individuals could have obtained the encryption key and accessed highly sensitive information without being noticed.

"Encryption is an important component of security, but it must be seen as a part of the whole. Encryption by itself doesn't solve anything, but as part of a system it can give security and confidence," says Russell.

Auditors also need to evaluate network, physical, and application security, and ask what algorithms the company is using and if they are commonly accepted. For example, Wilson says he often encounters companies that use good encryption technology but do not encrypt every dial-up port. Very important, too, is that while cryptography may be an important component of the technology component of security, process (including policies and procedures) and people (including organization, training) also are key factors in successful security within the enterprise. "A lot of times they have a secure encryptor, but the dial-up port is open," Wilson says. "They should look at secure modems for dial-in. The problem comes in the actual outside support for networks that have unsecured modems on them."

Remember to Encrypt E-Mail

Russell says that, in his view, the most common mistake is in e-mail. "Information is sent all the time internally that is sensitive and accessible," he says. "Ideas, contracts, product proposals, client lists, all kinds of stuff goes through e-mail, yet nobody considers it as an important area to secure. Nearly all organizations have underestimated the need to encrypt e-mail."

Most firms are using encryption somewhere within their organization, particularly for secure Web pages. While this protects information at the front end, it does not protect it at the back end, according to Russell. "On the back end, inside the company, somebody could get that information," he says. He suggests asking who should have access to it and how can it be kept out of everyone else's hands.

"Anything you consider sensitive information that you don't want to get into the wrong hands, you should consider encrypting," Russell says. "It must be sensitive and potentially accessible. If a computer is locked in a vault and nobody can get to it, it doesn't need encryption. If that computer is on a network, it becomes vulnerable."

Russell suggests internal auditors ask the following questions when evaluating security applications.

Does the Vendor Have Credibility in Security Circles?

As security awareness has increased, so has the number of security start-ups. Many of them are unqualified, according to Russell. Look for companies that frequent security conferences, such as RSA Security Inc.'s annual conference. Also look for vendors that are recognized in security journals. Although doing this is not foolproof, it will narrow the field of credible vendors. Depending on the criticality of the system and the intended investment, it may be best to solicit the help of a security consultant.

Does the Product Use Well-Known Cryptographic Algorithms?

The marketing of security applications tends to be an alphabet soup of acronyms. For this reason, it is helpful to know which ones really matter. There are essentially three categories of algorithms: asymmetric key, symmetric key, and hashing. Asymmetric key algorithms are normally used for negotiating a key between two parties. Symmetric key algorithms are normally used for traffic encryption. And hashing is used to create a message digest, which is a number computationally related to the message. It is generally used in relationship with an asymmetric key algorithm to create digital signatures. It also should be noted that although these three categories of algorithms are typical of new systems that are being built today, there exist many legacy applications at larger companies using crypto-systems from the 1970s. Because of the high associated costs, many of these companies have not been retrofitted with the "appropriate" form of cryptography.

The following list represents a few of the more popular algorithms that are tried and true:

- *RSA*. Named after Rivest, Shamir, and Adleman who created it, this asymmetric key algorithm is used for digital signatures and key exchanges.
- *Triple DES*. This algorithm uses the Data Encryption Standard three times in succession in order to provide 112-bit encryption. If it uses three keys, then sometimes it is referred to as having 168-bit encryption.
- *RC4*. This is a widely used variable-key-size symmetric key encryption algorithm that was created by RSA. The algorithm should be used with 128-bit encryption.
- *AES*. Advanced Encryption Standard is a new symmetric key algorithm also known as Rijndael. This new standard is intended to replace DES for protecting sensitive information.
- *SHA1*. The Secure Hash Algorithm was developed by the U.S. government. This algorithm is used for creating message digests and may be used to create a digital signature.
- *MD5*. Message Digest 5 was created by RSA, and is used to create message digests. It is frequently used with an asymmetric key algorithm to create a digital signature.

Does the Product Use SSL v3.0?

Secure Sockets Layer v3.0 is a transport-layer security protocol that is responsible for authenticating one or both parties, negotiating a key exchange, selecting an encryption algorithm, and transferring data securely. Although not every application needs to send information to another computer using this protocol, using it avoids some of the possible pitfalls that may go unnoticed in the development of a proprietary protocol.

Does the Company Report and Post Bug Fixes for Security Weaknesses?

No product is ever perfectly secure, but some vendors want you to think they are. When a company posts bug fixes and notices for security weaknesses, this should be considered a strength. This means they are committed to security, regardless of the impression it might give otherwise.

Does the Product Use an Accepted Random Number Generator to Create Keys?

Random number generators are notoriously difficult to implement. When they are implemented incorrectly, their output becomes predictable, negating the randomness required. Regardless of the encryption algorithm used, a sensitive message can be compromised if the key protecting it is predictable. RSA is currently developing a standard to address this issue. It will be called PKCS #14.

Does the Product Allow for Easy Integration of Hardware Tokens to Store Keys?

Whenever keys are stored as a file on a computer, they are accessible. Often the business case will determine the level of effort used to protect the keys, but the best protection for encryption keys is hardware. Smart cards and PCMCIA cards are often used for this purpose. An application should have the ability to utilize these hardware tokens seamlessly.

Has the Product Received a Federal Information Processing Standards (FIPS) 140-1 Verification?

The National Institute of Standards and Technology (NIST) has created a government-approved standard, referred to as FIPS 140-1, for cryptographic modules. NIST created four levels, which correspond to increasing levels of security. Depending on whether the crypto-module is a stand-alone component or one that is embedded in a larger component, and whether the crypto-model is a hardware device or a software implementation, the crypto-module is subjected to varying requirements to achieve specific validation levels. Issues such as tamper detection and response are addressed at Level 3 (that is, the ability for the cryptographic module to sense when it is being tampered with and to take appropriate action to zeroize the cryptographic keying material and sensitive unencrypted information within the module at the time of tamper). Level 4 considers the operating environment and requires that the module appropriately handle cryptographic security when the module is exposed to temperatures and voltages that are outside of the normal operating range of the module. Because FIPS 140-1 validation considered both the design and implementation of cryptographic modules, the following 11 components are scrutinized during the validation:

1. Basic design and documentation
2. Module interfaces
3. Roles and services
4. Finite state machine model
5. Physical security
6. Software security
7. Operating system security
8. Key management
9. Cryptographic algorithms
10. Electromagnetic compatibility (EMC/EMI)
11. Self-test

"Although no checklist will help you to avoid every security weakness, asking these questions could help you to avoid making a potentially bad decision," Russell says.

Resources

1. Symmetrical and asymmetrical encryption: http://glbld5001/InternalAudit/website.nsf/content/HotIssues-SupportSymmetricalandasymmetricalencryption!OpenDocument.
2. NIST Cryptographic Module Validation: http://csrc.nist.gov/.

110

Message Authentication

James S. Tiller, CISA, CISSP

For centuries, various forms of encryption have provided confidentiality of information and have become integral components of computer communication technology**.** Early encryption techniques were based on shared knowledge between the communication participants. Confidentiality and basic authentication were established by the fact that each participant must know a common secret to encrypt or decrypt the communication, or as with very early encryption technology, the diameter of a stick.

The complexity of communication technology has increased the sophistication of attacks and has intensified the vulnerabilities confronting data. The enhancement of communication technology inherently provides tools for attacking other communications. Therefore, mechanisms are employed to reduce the new vulnerabilities that are introduced by new communication technology. The mechanisms utilized to ensure confidentiality, authentication, and integrity are built on the understanding that encryption alone, or simply applied to the data, will not suffice any longer. The need to ensure that the information is from the purported sender, that it was not changed or viewed in transit, and to provide a process to validate these concerns is, in part, the responsibility of message authentication.

This chapter describes the technology of message authentication, its application in various communication environments, and the security considerations of those types of implementations.

History of Message Authentication

An encrypted message could typically be trusted for several reasons. First and foremost, the validity of the message content was established by the knowledge that the sender had the appropriate shared information to produce the encrypted message. An extension of this type of assumed assurance was also recognized by the possession of the encrypting device. An example is the World War II German Enigma, a diabolically complex encryption machine that used three or four wheels to produce ciphertext as an operator typed in a message. The Enigma was closely guarded; if it fell into the enemy's possession, the process of deciphering any captured encrypted messages would become much less complex. The example of the Enigma demonstrates that possession of a device in combination with the secret code for a specific message provided insurance that the message contents received were genuine and authenticated.

As the growth of communication technology embraced computers, the process of encryption moved away from complex and rare mechanical devices to programs that provided algorithms for encryption. The mechanical algorithm of wheels and electrical conduits was replaced by software that could be loaded onto computers, which are readily available, to provide encryption. As algorithms were developed, many became open to the public for inspection and verification for use as a standard. Once the algorithm was exposed, the power of protection was in the key that was combined with the clear message and fed into the algorithm to produce ciphertext.

Why Authenticate a Message?

The ability of a recipient to trust the content of a message is placed squarely on the trust of the communication medium and the expectation that it came from the correct source. As one would imagine, this example of open communication is not suitable for information exchange and is unacceptable for confidential or any form of valuable data.

There are several types of attacks on communications that range from imposters posing as valid participants replaying or redelivering outdated information, to data modification in transit.

Communication technology has eliminated the basic level of interaction between individuals. For two people talking in a room, it can be assured — to a degree — that the information from one individual has not been altered prior to meeting the listener's ears. It can be also assumed that the person that is seen talking is the originator of the voice that is being heard. This example is basic, assumed, and never questioned — it is trusted. However, the same type of communication over an alternate medium must be closely scrutinized due to the massive numbers of vulnerabilities to which the session is exposed.

Computers have added several layers of complexity to the trusting process and the Internet has introduced some very interesting vulnerabilities. With a theoretically unlimited number of people on a single network, the options of attacks are similarly unlimited. As soon as a message takes advantage of the Internet as a communication medium, all bets are off without layers of protection.

How are senders sure that what they send will be the same when it reaches the intended recipient? How can senders be sure that the recipients are who they claim to be? The same questions hold true for the recipients and the question of initiator identity.

Technology Overview

It is virtually impossible to describe message authentication without discussing encryption. Message authentication is nothing more than a form of cryptography and, in certain implementations, takes advantage of encryption algorithms.

Hash Function

Hash functions are computational functions that take a variable-length input of data and produce a fixed-length result that can be used as a fingerprint to represent the original data. Therefore, if the hashes of two messages are identical, it can be reasonably assumed that the messages are identical as well. However, there are caveats to this assumption, which are discussed later.

Hashing information to produce a fingerprint will allow the integrity of the transmitted data to be verified. To illustrate the process, Alice creates the message "Mary loves basketball," and hashes it to produce a smaller, fixed-length message digest, "a012f7." Alice transmits the original message and the hash to Bob. Bob hashes the message from Alice and compares his result with the hash received with the original message from Alice. If the two hashes match, it can be assumed that the message was not altered in transit. If the message was changed after Alice sent it and before Bob received it, Bob's hash will not match, resulting in discovering the loss of message integrity. This example is further detailed in Exhibit 110.1.

In the example, a message from Alice in cleartext is used as input for a hash function. The result is a message digest that is a much smaller, fixed-length value unique to the original cleartext message. The message digest is attached to the original cleartext message and sent to the recipient, Bob. At this point, the message and the hash value are in the clear and vulnerable to attack. When Bob receives the message, he separates the message from the digest and hashes the message using the same hash function Alice used. Once the hash process is complete, Bob compares his message digest result with the one included with the original message from Alice. If the two match, the message was not modified in transit.

The caveat to the example illustrated is that an attacker using the same hashing algorithm could simply intercept the message and digest, create a new message and corresponding message digest, and forward it on to the original recipient. The type of attack, known as the "man in the middle," described here is the driving reason why message authentication is used as a component in overall message protection techniques.

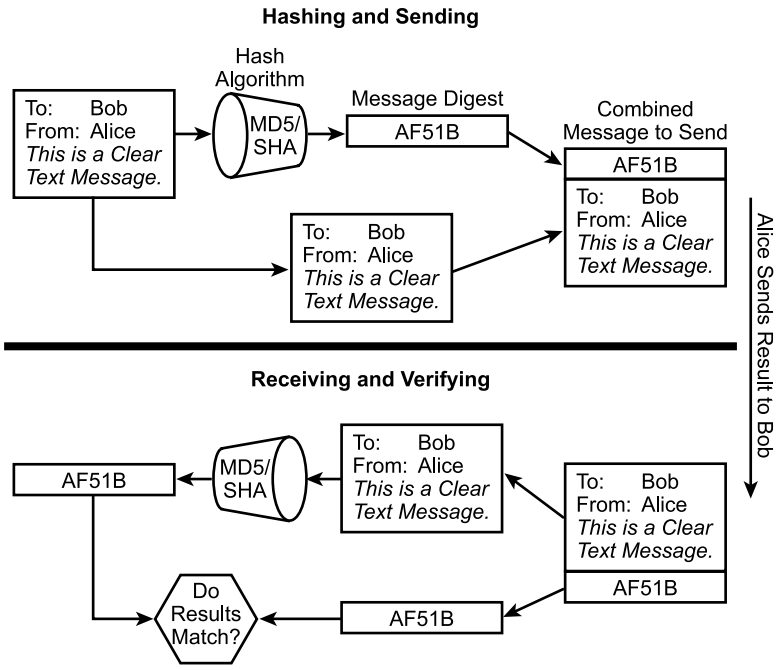


EXHIBIT 110.1 Hash function.

Encryption

Encryption, simply stated, is the conversion of plaintext into unintelligible ciphertext. Typically, this is achieved with the use of a key and an algorithm. The key is combined with the plaintext and computed with a specific algorithm.

There are two primary types of encryption keys: symmetrical and asymmetrical.

Symmetrical

Symmetrical keys, as shown in Exhibit 110.2, are used for both encryption and decryption of the same data. It is necessary for all the communication participants to have the same key to perform the encryption and decryption. This is also referred to as a shared secret.

In the example, Alice creates a message that is input into an encryption algorithm that uses a unique key to convert the clear message into unintelligible ciphertext. The encrypted result is sent to Bob, who has obtained the same key through a mechanism called "out-of-band" messaging. Bob can now decrypt the ciphertext by providing the key and the encrypted data as input for the encryption algorithm. The result is the original plaintext message from Alice.

Asymmetrical

To further accentuate authentication by means of encryption, the technology of public key cryptography, or asymmetrical keys, can be leveraged to provide message authentication and confidentiality.

Alice and Bob each maintain a private and public key pair that is mathematically related. The private key is well protected and is typically passphrase protected. The public key of the pair is provided to anyone who wants it and wishes to send an encrypted message to the owner of the key pair.

An example of public key cryptography, as shown in Exhibit 110.3, is that Alice could encrypt a message with Bob's public key and send the ciphertext to Bob. Because Bob is the only one with the matching private key, he would be the only recipient who could decrypt the message. However, this interaction only provides confidentiality and not authentication because anyone could use Bob's public key to encrypt a message and claim to be Alice.

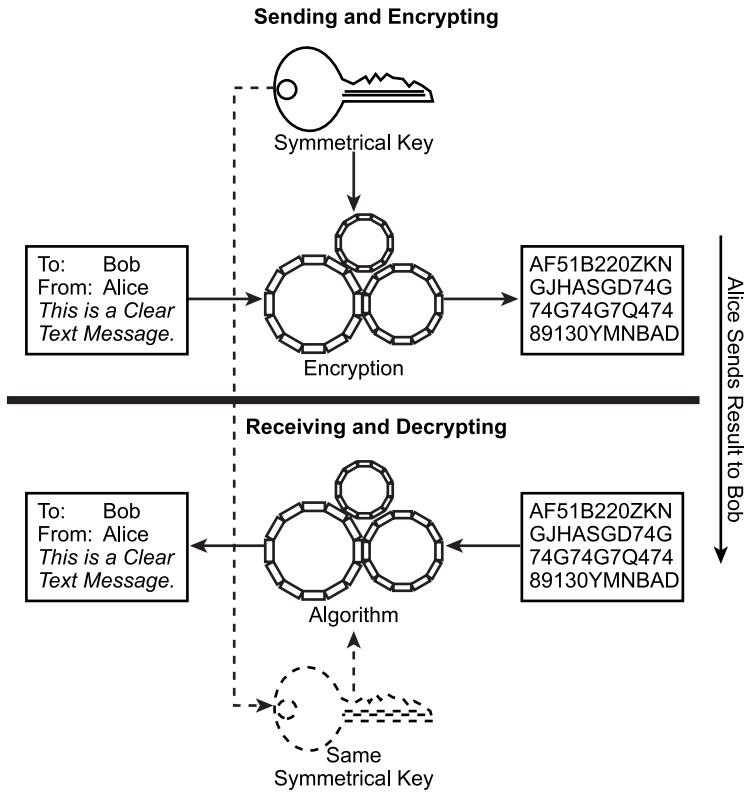


EXHIBIT 110.2 Symmetrical key encryption.

As illustrated in Exhibit 110.3, the encryption process is very similar to normal symmetrical encryption. A message is combined with a key and processed by an algorithm to construct ciphertext. However, the key being used in the encryption cannot be used for decryption. As detailed in the example, Alice encrypts the data with the public key and sends the result to Bob. Bob uses the corresponding private key to decrypt the information.

To provide authentication, Alice can use her private key to encrypt a message digest generated from the original message, then use Bob's public key to encrypt the original cleartext message, and send it with the encrypted message digest. When Bob receives the message, he can use his private key to decrypt the message. The output can then be verified using Alice's public key to decrypt the message authentication that Alice encrypted with her private key. The process of encrypting information with a private key to allow the recipient to authenticate the sender is called digital signature. An example of this process is detailed in Exhibit 110.4.

The illustration conveys a typical application of digital signature. There are several techniques of creating digital signatures; however, the method detailed in the exhibit represents the use of a hash algorithm. Alice generates a message for Bob and creates a message digest with a hash function. Alice then encrypts the message digest with her private key. By encrypting the digest with her private key, Alice reduces the system load created by the processor-intensive encryption algorithm and provides an authenticator. The encrypted message digest is attached to the original cleartext message and encrypted using Bob's public key. The example includes the encrypted digest with the original message for the final encryption, but this is not necessary. The final result is sent to Bob. The entire package is decrypted with Bob's private key — ensuring recipient authentication. The result is the cleartext message and an encrypted digest. Bob decrypts the digest with Alice's public key, which authenticates the sender. The result is the original hash created by Alice that is compared to the hash Bob created using the cleartext message. If the two match, the message content has been authenticated along with the communication participants.

Digital signatures are based on the management of public and private keys and their use in the communication. The process of key management and digital signatures has evolved into certificates. Certificates, simply stated, are public keys digitally signed by a trusted Certificate Authority. This provides comfort in the knowledge

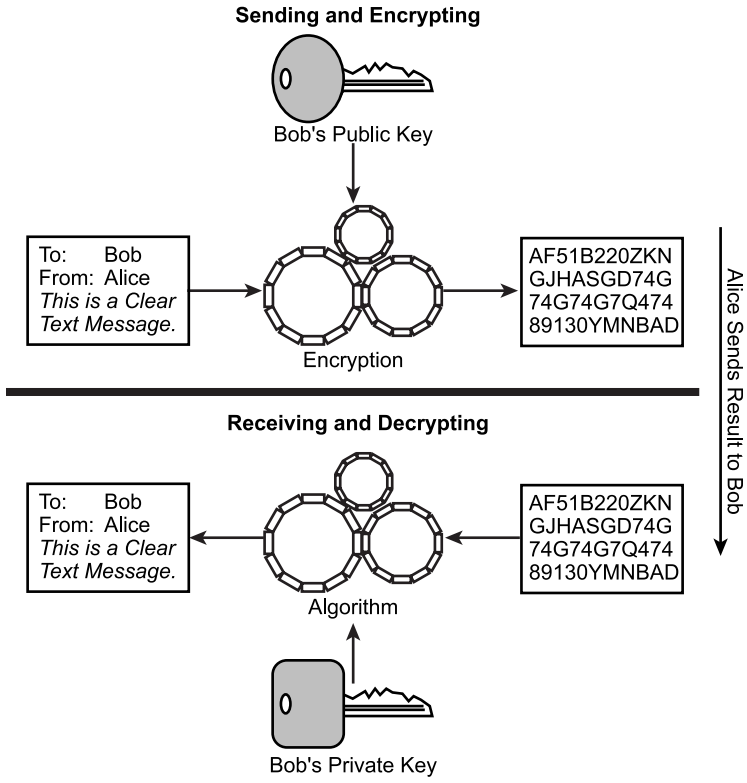


EXHIBIT 110.3 Asymmetrical key encryption.

that the public key being used to establish encrypted communications is owned by the proper person or organization.

Message Authentication Code

Message authentication code (MAC) with DES is the combination of encryption and hashing. As illustrated in Exhibit 110.5, as data is fed into a hashing algorithm, a key is introduced into the process.

MAC is very similar to encryption but the MAC is designed to be irreversible, like a standard hash function. Because of the computational properties of the MAC process, and the inability to reverse the encryption designed into the process, MACs are much less vulnerable to attacks than encryption with the same key length. However, this does not prevent an attacker from forging a new message and MAC.

MAC ensures data integrity like a message digest but adds limited layers of authentication because the recipient would have to have the shared secret to produce the same MAC to validate the message.

The illustration of a message authentication code function appears very similar to symmetrical encryption; however, the process is based on compressing the data into a smaller fixed length that is not designed for decryption. A message is passed into the algorithm, such as DES-CBC, and a symmetrical key is introduced. The result is much like that of a standard message digest, but the key is required to reproduce the digest for verification.

The Need for Authentication

As data is shared across networks — networks that are trusted or not — the opportunities for undesirables to interact with the session are numerous. Of the attacks that communications are vulnerable to, message authentication, in general application, addresses only a portion of the attacks. Message authentication is used as a tool to combine various communication-specific data that can be verified by the valid parties for each

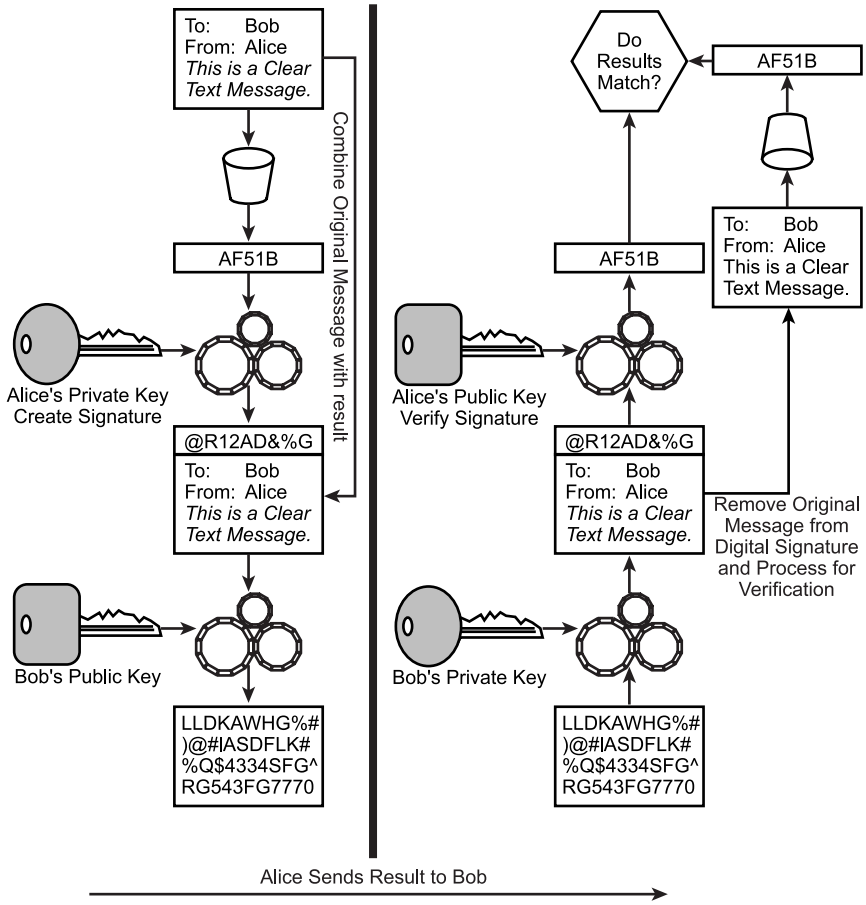


EXHIBIT 110.4 Digital signature with the use of hash functµions.

message received. Message authentication alone is not an appropriate countermeasure; but when combined with unique session values, it can protect against four basic categories of attacks:

1. Masquerading
2. Content modification
3. Sequence manipulation
4. Submission modification

To thwart these vulnerabilities inherent in communications, hash functions can be used to create message digests that contain information for origination authentication and timing of the communications. Typically, time-sensitive random information, or a nonce, is provided during the initialization of the session. The nonce can be input with the data in the hashing process or used as key material to further identify the peer during communications. Also, sequence numbers and time stamps can be generated and hashed for communications that require consistent session interaction — not like that of nontime-sensitive data such as e-mail. The process of authentication, verification through the use of a nonce, and the creation of a key for MAC computations provides an authenticated constant throughout the communication.

Masquerading

The process of masquerading as a valid participant in a network communication is a type of attack. This attack includes the creation of messages from a fraudulent source that appears to come from an authorized origin.

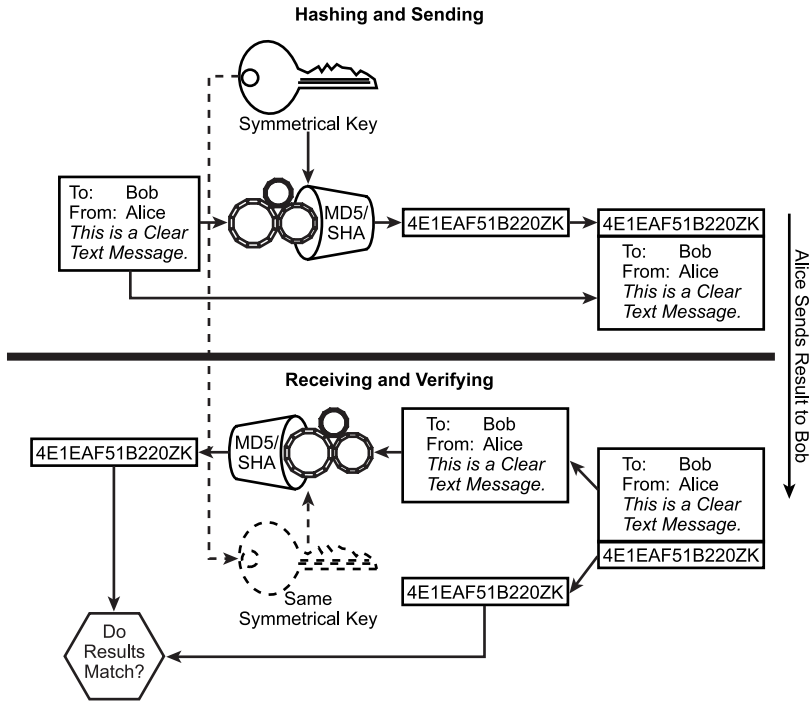


EXHIBIT 110.5 Message authentication code.

Masquerading can also represent the acknowledgment of a message by an attacker in place of the original recipient. False acknowledgment or denial of receipt could complicate non-repudiation issues. The nonce that may have been used in the hash or the creation of a symmetrical key assists in the identification of the remote system or user during the communication. However, to accommodate origin authentication, there must be an agreement on a key prior to communication. This is commonly achieved by a preshared secret or certificate that can be used to authenticate the initial messages and create specific data for protecting the remainder of the communication.

Content Modification

Content modification is when the attacker intercepts a message, changes the content, and then forwards it to the original recipient. This type of attack is quite severe in that it can manifest itself in many ways, depending on the environment.

Sequence Manipulation

Sequence manipulation is the process of inserting, deleting, or reordering datagrams. This type of attack can have several types of effects on the communication process, depending on the type of data and communication standard. The primary result is denial of service. Destruction of data or confusion of the communication can also result.

Submission Modification

Timing modification appears in the form of delay or replay. Both of these attacks can be quite damaging. An example is session establishment. In the event that the protocol is vulnerable to replay, an attacker could use the existence of a valid session establishment to gain unauthorized access.

Message authentication is a procedure to verify that the message received is from the intended source and has not been modified or made susceptible to the previously outlined attacks.

Authentication Foundation

To authenticate a message, an authenticator must be produced that can be used later by the recipient to authenticate the message. An authenticator is a primitive reduction or representation of the primary message to be authenticated. There are three general concepts in producing an authenticator.

Encryption

With encryption, the ciphertext becomes the authenticator. This is related to the trust relationship discussed earlier by assuming the partner has the appropriate secret and has protected it accordingly.

Consider typical encrypted communications: a message sent from Alice to Bob encrypted with a shared secret. If the secret's integrity is maintained, confidentiality is assured by the fact that no unauthorized entities have the shared secret.

Bob can be assured that the message is valid because the key is secret and an attacker without the key would be unable to modify the ciphertext in a manner to make the desired modifications to the original plaintext message.

Message Digest

As briefly described above, hashing is a function that produces a unique fixed-length value that serves as the authenticator for the communication. Hash functions are one-way, in that the creation of the hash is quite simple, but the reverse is infeasible. A well-constructed hash function should be collision resistant. A collision is when two different messages produce the same result or digest. For a function to take a variable length of data and produce a much smaller fixed-length result, it is mathematically feasible to experience collisions. However, a well-defined algorithm with a large result should have a high resistance to collisions.

Hash functions are used to provide message integrity. It can be argued that encryption can provide much of the same integrity. An example is an attacker could not change an encrypted message to modify the resulting cleartext. However, hash functions are much faster than encryption processes and can be utilized to enhance performance while maintaining integrity. Additionally, the message digest can be made public without revealing the original message.

Message Authentication Code

Message authentication code with DES is a function that uses a secret key to produce a unique fixed-length value that serves as the authenticator. This is much like a hash algorithm but provides the added protection by use of a key. The resulting MAC is appended to the original message prior to sending the data. MAC is similar to encryption but cannot be reversed and does not directly provide any authentication process because both parties share the same secret key.

Hash Process

As mentioned, a hash function is a one-way computation that accepts a variable-length input and produces a fixed-length result. The hash function calculates each bit in a message; therefore, if any portion of the original message changes, the resulting hash will be completely different.

Function Overview

A hash function must meet several requirements to be used for message authentication. The function must:

- Be able to accept any size data input
- Produce a fixed-length output
- Be relatively easy to execute, using limited resources
- Make it computationally impractical to derive a message from the digest (one-way property)
- Make it computationally impractical to create a message digest that is equal to a message digest created from different information (collision resistance)

Hash functions accommodate these requirements by a set of basic principles. A message is processed in a sequence of blocks, as shown in Exhibit 110-6. The size of the blocks is determined by the hash function. The function addresses each block one at a time and produces parity for each bit. Addressing each bit provides the message digest with the unique property that dramatic changes will occur if a single bit is modified in the original message.

As detailed in Exhibit 110.6, the message is separated into specific portions. Each portion is XOR with the next portion, resulting in a value the same size of the original portions, not their combined value. As each result is processed, it is combined with the next portion until the entire message has been sent through the function. The final result is a value the size of the original portions that were created and a fixed-length value is obtained.

Message Authentication Codes and Processes

Message authentication code with DES is applying an authentication process with a key. MACs are created using a symmetrical key so the intended recipient or the bearer of the key can only verify the MAC. A plain hash function can be intercepted and replaced or brute-force attacked to determine collisions that can be of use to the attacker. With MACs, the addition of a key complicates the attack due to the secret key used in its computation.

There are four modes of DES that can be utilized:

1. Block cipher-based
2. Hash function-based
3. Stream cipher-based
4. Unconditionally secure

Block Cipher-Based Mode

Block cipher-based message authentication can be derived from block cipher algorithms. A commonly used version is DES-CBC-MAC, which, simply put, is DES encryption based on the Cipher Block Chaining (CBC) mode of block cipher to create a MAC. A very common form of MAC is Data Authentication Algorithm (DAA), which is based on DES. The process uses the CBC mode of operation of DES with a zero initialization vector. As illustrated in Exhibit 110-7, the message is grouped into contiguous blocks of 64 bits; the last group is padded on the right with zeros to attain the 64-bit requirement. Each block is fed into the DES algorithm with a key to produce a 64-bit Data Authentication Code (DAC). The resulting DAC is XOR and the next 64 bits of data is then fed again into the DES algorithm. This process continues until the last block, and returns the final MAC.

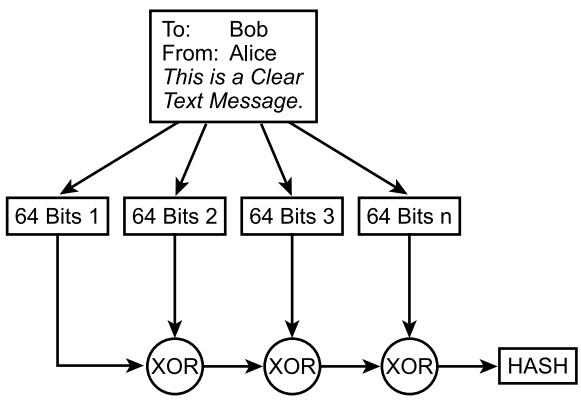


EXHIBIT 110.6 Simple hash function example.

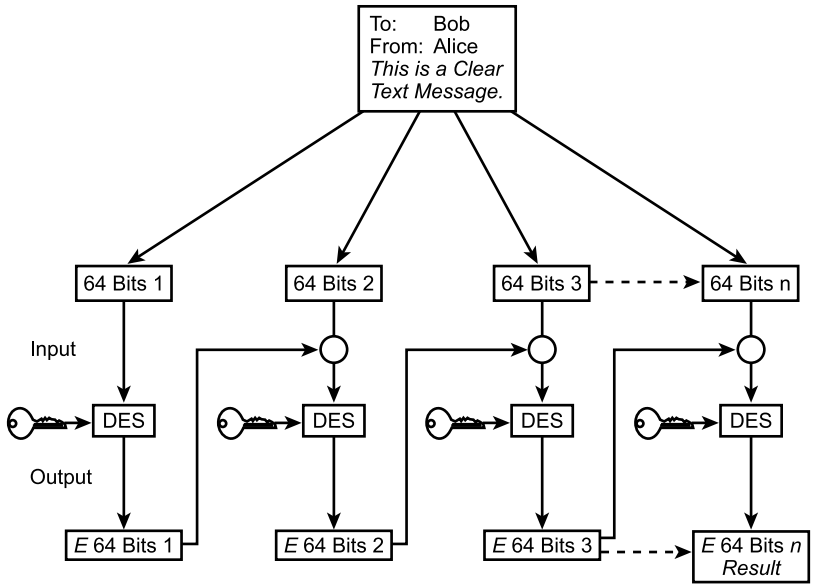


EXHIBIT 110.7 MAC based on DES CBC.

A block cipher is a type of symmetric key encryption algorithm that accepts a fixed block of plaintext to produce ciphertext of the same length — a linear relationship. There are four primary modes of operation on which the block ciphers can be based:

1. *Electronic Code Book (ECB)*. Electronic Code Book mode accepts each block of plaintext and encrypts it independently of previous block cipher results. The weakness in ECB is that identical input blocks will produce identical cipher results of the same length. Interestingly, this is a fundamental encryption flaw that affected the Enigma. For each input, there was a corresponding output of the same length. The "step" of the last wheel in an Enigma could be derived from determinations in ciphertext patterns.
2. *Cipher Block Chaining (CBC)*. With CBC mode, each block result of ciphertext is exclusively OR'ed (XOR) with the previous calculated block, and then encrypted. Any patterns in plaintext will not be transferred to the cipher due to the XOR process with the previous block.
3. *Cipher Feedback (CFB)*. Similar to CBC, CFB executes an XOR between the plaintext and the previous calculated block of data. However, prior to being XORed with the plaintext, the previous block is encrypted. The amount of the previous block to be used (the feedback) can be reduced and not utilized as the entire feedback value. If the full feedback value is used and two cipher blocks are identical, the output of the following operation will be identical. Therefore, any patterns in the message will be revealed.
4. *Output Feedback (OFB)*. Output Feedback is similar to CFB in that the result is encrypted and XORed with the plaintext. However, the creation of the feedback is generated independently of the ciphertext and plaintext processes. A sequence of blocks is encrypted with the previous block, the result is then XORed with the plaintext.

Hash Function-Based Mode

Hash function-based message authentication code (HMAC) uses a key in combination with hash functions to produce a checksum of the message. RFC 2104 defines that HMAC can be used with any iterative cryptographic hash function (e.g., MD5, SHA-1) in combination with a secret shared key. The cryptographic strength of HMAC depends on the properties of the underlying hash function.

The definition of HMAC requires a cryptographic hash function and a secret key. The hash function is where data is hashed by iterating a basic compression function on blocks of data, typically 64 bytes in each

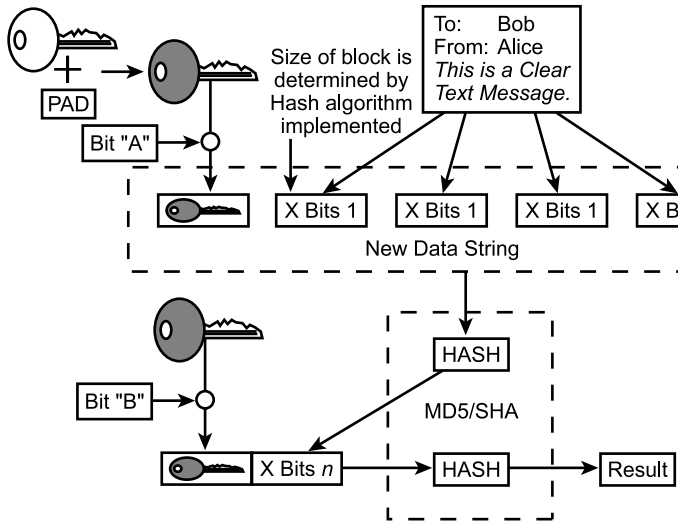


EXHIBIT 110.8 Simple HMAC example.

block. The symmetrical key to be used can be any length up to the block size of the hash function. If the key is longer than the hash block size, the key is hashed and the result is used as the key for the HMAC function.

This process is very similar to the DES-CBC-MAC discussed above; however, the use of the DES algorithm is significantly slower than most hashing functions, such as MD5 and SHA-1.

HMAC is a process of combining existing cryptographic functions and a keyed process. The modularity of the standard toward the type of cryptographic function that can be used in the process has become the point of acceptance and popularity. The standards treat the hash function as a variable that can consist of any hash algorithm. The benefits are that legacy or existing hash implementations can be used in the process and the hash function can be easily replaced without affecting the process. The latter example represents an enormous security advantage. In the event the hash algorithm is compromised, a new one can be immediately implemented.

There are several steps to the production of an HMAC; these are graphically represented in Exhibit 110.8. The first step is to determine the key length requested and compare it to the block size of the hash being implemented. As described above, if the key is longer than the block size it is hashed, the result will match the block size defined by the hash. In the event the key is smaller, it is padded with zeros to accommodate the required block size.

Once the key is defined, it is XOR'ed with a string of predefined bits "A" to create a new key that is combined with the message. The new message is hashed according to the function defined (see Exhibit 110.6). The hash function result is combined with the result of XOR the key with another defined set of bits "B." The new combination of the second key instance and the hash results are hashed again to create the final result.

Stream Cipher-Based Mode

A stream cipher is a symmetric key algorithm that operates on small units of plaintext, typically bits. When data is encrypted with a stream cipher, the transformation of the plaintext into ciphertext is dependent on when the bits were merged during the encryption. The algorithm creates a keystream that is combined with the plaintext. The keystream can be independent of the plaintext and ciphertext (typically referred to as a synchronous cipher), or it can depend on the data and the encryption (typically referred to as self-synchronizing cipher).

Unconditionally Secure Mode

Unconditional stream cipher is based on the theoretical aspects of the properties of a one-time pad. A one-time pad uses a string of random bits to create the keystream that is the same length as the plaintext message.

The keystream is combined with the plaintext to produce the ciphertext. This method of employing a random key is very desirable for communication security because it is considered unbreakable by brute force. Security at this level comes with an equally high price: key management. Each key is the same size and length as the message it was used to encrypt, and each message is encrypted with a new key.

Message Authentication over Encryption

Why use message authentication (e.g., hash functions and message authentication codes) when encryption seems to meet all the requirements provided by message authentication techniques? Following are brief examples and reasoning to support the use of message authentication over encryption.

Speed

Cryptographic hash functions, such as MD5 and SHA-1, execute much faster and use less system resources than typical encryption algorithms. In the event that a message only needs to be authenticated, the process of encrypting the entire message, such as a document or large file, is not entirely logical and consumes valuable system resources.

The reasoning of reducing load on a system holds true for digital signatures. If Alice needs to send a document to Bob that is not necessarily confidential but may contain important instructions, authentication is paramount. However, encrypting the entire document with Alice's private key is simply overkill. Hashing the document will produce a very small rendition of the original message, which then can be encrypted with her private key. The much smaller object encrypts quickly and provides ample authentication and abundant message integrity.

Limited Restrictions

No export restrictions on cryptographic functions are defined. Currently, the laws enforcing import and export restrictions in the international community are complicated and constantly changing. Basically, these laws are to control the level of technology and intellectual property of one country from another. Message authentication releases the communication participants from these restrictions.

Application Issues

There are applications where the same message is broadcast to several destinations. One system is elected as the communication monitor and verifies the message authentication on behalf of the other systems. If there is a violation, the monitoring system alerts the other systems.

Simple Network Management Protocol (SNMP) is an example where command messages can be forged or modified in transit. With the application of MAC, or HMAC, a password can be implemented to act as a key to allow a degree of authentication and message authentication. Each system in the community is configured with a password that can be combined with the data during the hash process and verified upon receipt. Because all the members are configured with the same password, the data can be hashed with the locally configured password and verified. It can also be forged at the destination.

System Operation

In the event that one of a communication pair is overburdened, the process of decryption would be overwhelming. Authentication can be executed in random intervals to ensure authentication with limited resources. Given the hashing process is much less intensive than encryption, periodical hashing and comparisons will consume fewer system cycles.

Code Checksum

Application authentication is achieved by adding the checksum to the program. While the program itself may be open to modification, the checksum can be verified at runtime to ensure that the code is in the original format and should produce the expected results. Otherwise, an attacker could have constructed a malicious activity to surreptitiously operate while the original application was running. It can be argued that if an attacker

can modify the code, the checksum should pose little resistance because it can also be simply regenerated. Given the typically small size of checksums, it is typically published on several Web pages or included in an e-mail. In other words, an attacker would have to modify every existence of the checksum to ensure that the recipient would inadvertently verify the modified application. If encryption was utilized, the program would have to decrypt at each runtime, consuming time and resources. This is very important for systems that provide security functions, such as firewalls, routers, and VPN access gateways.

An example of the need for code protection can be illustrated by the heavy reliance on the Internet for obtaining software, updates, or patches. In early computing, systems patches and software were mailed to the recipient as the result of a direct request, or as a registered system user. As communication technology advanced, Bulletin Board Systems (BBS) could be directly accessed with modems to obtain the necessary data. In both of these examples, a fair amount of trust in the validity of the downloaded code is assumed.

In comparison, the complexity of the Internet is hidden from the user by a simple browser that is used to access the required files. The data presented in a Web page can come from dozens of different sources residing on many different servers throughout the Internet. There are few methods to absolutely guarantee that the file being downloaded is from a trusted source. To add to the complexity, mirrors can be established to provide a wider range of data sources to the Internet community. However, the security of a mirrored site must be questioned. The primary site may have extensive security precautions, but a mirror site may not. An attacker could modify the code on an alternate download location. When the code is finally obtained, a checksum can be validated to ensure that the code obtained is the code the creator intended for receipt.

Utilization of Existing Resources

There is available installed technology designed for DES encryption processes. The use of DEC-CBC-MAC can take advantage of existing technology to increase performance and support the requirements of the communication. The DES encryption standard has been available for quite some time. There are many legacy systems that have hardware designed specifically for DES encryption. As more advanced encryption becomes available and new standards evolve, the older hardware solutions can be utilized to enhance the message authentication process.

Security Considerations

The strength of any message authentication function, such as a MAC or hash, is determined by two primary factors:

1. One-way property
2. Collision resistance

One-way property is the ability of the hash to produce a message digest that cannot be used to determine the original message. This is one of the most significant aspects of message authentication algorithms. If a message authentication algorithm is compromised and a weakness is discovered, the result could have a detrimental effect on various forms of communication.

MD4 is an example of a function's poor one-way property. Within MD4, the data is padded to obtain a length divisible by 512, plus 448. A 64-bit value that defines the original message's length is appended to the padded message. The result is separated into 512-bit blocks and hashed using three distinct rounds of computation. Weaknesses were quickly discovered if the first or last rounds were not processed. However, it was later discovered that without the last round, the original message could be derived. MD4 had several computation flaws that proved the function had limited one-way capabilities.

Collision resistance is the most considered security aspect of message authentication functions. A collision is typically defined as when two different messages have the same hash result. In the event that a hash function has a collision vulnerability, such as MD2, a new message can be generated and used to replace the original in a communication, and the hash will remain valid. The combination of the original hash and the known vulnerability will provide the attacker with enough information to produce an alternative message that will produce the same checksum. An example is the hash algorithm MD2. It was created for 8-bit computers in the late 1980s and uses 16-bit blocks of the message against which to execute the hash. MD2 produces a 16-bit checksum prior to passing through the hash function. If this checksum is omitted, the production of a

collision would be trivial. MD4 was subject to weak collision resistance as well, and it was proven that collisions could be produced in less than a minute on a simple personal computer.

The concept of a collision is a fundamental issue concerning probabilities. Take, for example, a hash function that produces an n-bit digest. If one is looking for a result of x, it can be assumed that one would have to try 2^n input possibilities. This type of brute-force attack is based on a surprising outcome referred to as the "birthday paradox": What is the least number of people in a group that can provide the probability, greater than half, that at least two people will have the same birthday?

If there are 365 days per year, and if the number of people exceeds 365, there will be a successful collision. If the number of people in the group is less than 365, then the number of possibilities is 365^n, where n is the number of people in a group. For those still wondering, the number of people, assuming there is a collision, is 23. This is a very small number; but when calculated against the number of possibilities that any two people's birthdays match, one sees that there are 253 possibilities. This is simply calculated as $n(n-1)/2$, which results in the probability of $P(365, 23) = 0.5073$, or greater than one half.

The birthday paradox states that given a random integer with a constant value between 1 and n, what is the selection of the number of permutations (the number of people required to meet 0.5 probability) that will result in a collision?

Given a fixed-length output that can represent an infinite amount of variation, it is necessary to understand the importance of a robust algorithm. It is also necessary for the algorithm to produce a relatively large result that remains manageable.

However, as certificates and other public key cryptography is utilized, message authentication processes will not be exposed to direct attack. The use of a hash to accommodate a digital signature process is based on the ownership and trust of a private key; the hash, while important, is only a step in a much more complicated process.

Conclusion

Communication technology has provided several avenues for unauthorized interaction with communications requiring the need to address security in ways previously unanalyzed. Message authentication provides a means to thwart various forms of attack and can enhance other aspects of communication security. A message "fingerprint" can be created in several ways, ranging from simple bit parity functions (hash) to utilization of encryption algorithms (DES-CBC-MAC) to complicated hybrids (HMAC). This fingerprint cannot only be used to ensure message integrity, but also given the inherent process of message reduction, it lends itself to authentication and signature processes.

Message authentication is a broad activity that employs several types of technology in various applications to achieve timely, secure communications. The combinations of the application of these technologies are virtually limitless and, as advancements in cryptography, cryptanalysis, and overall communication technology are realized, message authentication will most certainly remain an interesting process.

Fundamentals of Cryptography and Encryption

Ronald A. Gove

This chapter presents an overview of some basic ideas underlying encryption technology. The chapter begins by defining some basic terms and follows with a few historical notes so the reader can appreciate the long tradition that encryption, or secret writing, has had. The chapter then moves into modern cryptography and presents some of the underlying mathematical and technological concepts behind private and public key encryption systems such as DES and RSA. We will provide an extensive discussion of conventional private key encryption prior to introducing the concept of public key cryptography. We do this for both historical reasons (private key did come first) and technical reasons (public key can be considered a partial solution to the key management problem).

SOME BASIC DEFINITIONS

We begin our discussion by defining some terms that will be used throughout the chapter. The first term is *encryption*. In simplest terms, encryption is the process of making information unreadable by unauthorized persons. The process may be manual, mechanical, or electronic, and the core of this chapter is to describe the many ways that the encryption process takes place. Encryption is to be distinguished from message-hiding. Invisible inks, microdots, and the like are the stuff of spy novels and are used in the trade; however, we will not spend any time discussing these techniques for hiding information. Exhibit 19.1 shows a conceptual version of an encryption system. It consists of a sender and a receiver, a message (called the "plain text"), the encrypted message (called the "cipher text"), and an item called a "key." The encryption process, which transforms the plain text into the cipher text, may be thought of as a "black box." It takes inputs (the plain text and key) and produces output (the cipher text). The

messages may be handwritten characters, electromechanical representations as in a Teletype, strings of 1s and 0s as in a computer or computer network, or even analog speech. The black box will be provided with whatever input/output devices it needs to operate; the insides, or cryptographic algorithm will, generally, operate independently of the external representation of the information.

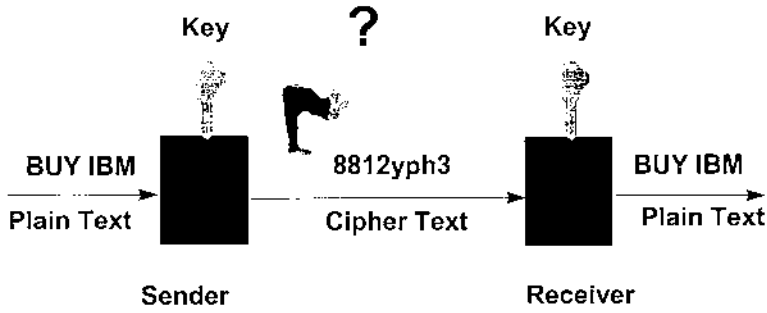


Exhibit 19.1. Conceptual Version of an Encryption System

The *key* is used to select a specific instance of the encryption process embodied in the machine. It is more properly called the "*cryptovariable*." The use of the term "key" is a holdover from earlier times. We will discuss cryptovariables (keys) in more detail in later sections. It is enough at this point to recognize that the cipher text depends on both the plain text and the cryptovariable. Changing either of the inputs will produce a different cipher text. In typical operation, a cryptovariable is inserted prior to encrypting a message and the same key is used for some period of time. This period of time is known as a "cryptoperiod." For reasons having to do with cryptanalysis, the key should be changed on a regular basis. The most important fact about the key is that it embodies the security of the encryption system. By this we mean the system is designed so that complete knowledge of all system details, including specific plain and cipher text messages, is not sufficient to derive the cryptovariable.

It is important that the system be designed in this fashion because the encryption process itself is seldom secret. The details of the data encryption standard (DES), for example, are widely published so that anyone may implement a DES-compliant system. In order to provide the intended secrecy in the cipher text, there has to be some piece of information that is not available to those who are not authorized to receive the message; this piece of information is the cryptovariable, or key.

Inside the black box is an implementation of an algorithm that performs the encryption. Exactly how the algorithm works is the main topic of this chapter, and the details depend on the technology used for the message.

Cryptography is the study of the means to do encryption. Thus cryptographers design encryption systems. Cryptanalysis is the process of figuring out the message without knowledge of the cryptovariable (key), or more generally, figuring out which key was used to encrypt a whole series of messages.

SOME HISTORICAL NOTES

The reader is referred to Kahn[1] for a well-written history of this subject. We note that the first evidence of cryptography occurred over 4000 years ago in Egypt. Almost as soon as writing was invented, we had secret writing. In India, the ancients' version of Dr. Ruth's Guide to Good Sex, the *Kama-Sutra,* places secret writing as 45th in a list of arts women should know. The Arabs in the 7th century AD were the first to write down methods of cryptanalysis. Historians have discovered a text dated about 855 AD that describes cipher alphabets for use in magic.

One of the better known of the ancient methods of encryption is the Caesar Cipher, so called because Julius Caesar used it. The Caesar Cipher is a simple alphabetic substitution. In a Caesar Cipher, each plain text letter is replaced by the letter 3 letters away to the right. For example, the letter A is replaced by D, B by E, and so forth. (See Exhibit 19.2, where the plain-text alphabet is in lower case and the cipher text is in upper case.)

a b c d e f g h i j k l m n o p q r s t u v w x y z

D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Plain text: Omnia Gallia est divisa in partes tres . . . .

Cipher Text: RPQLD JDOOLD HVW GLYLVD LQ SDUWHV WUHV . . .

Exhibit 19.2. The Caesar Cipher

Caesar's Cipher is a form of a more general algorithm known as monoalphabetic substitution. While Julius Caesar always used an offset of 3, in principal one can use any offset, from one to 25. (An offset of 26 is the original alphabet.) The value of the offset is in fact the cryptovariable for this simplest of all monoalphabetic substitutions. All such ciphers with any offset are now called Caesar Ciphers.

There are many ways to produce alphabetic substitution ciphers. In fact, there are 26! (26 factorial or 26X25X24 . . . X2X1) ways to arrange the 26 letters of the alphabet. All but one of these yields a nonstandard alphabet. Using a different alphabet for each letter according to some well-defined rule can make a more complicated substitution. Such ciphers are called polyalphabetic substitutions.

Cryptography underwent many changes through the centuries often following closely with advances in technology. When we wrote by hand, encryption was purely manual. After the invention of the printing press various mechanical devices appeared such as Leon Batista Alberti's cipher disk in Italy. In the 18th century, Thomas Jefferson invented a ciphering device consisting of a stack of 26 disks each containing the alphabet around the face of the edge. Each disk had the letters arranged in a different order. A positioning bar was attached that allowed the user to align the letters along a row. To use the device, one spelled out the message by moving each disk so that the proper letter lay along the alignment bar. The bar was then rotated a fixed amount (the cryptovariable for that message) and the letters appearing along the new position of the bar were copied off as the cipher text. The receiver could then position the cipher text letters on his "wheel" and rotate the cylinder until the plain text message appeared.

By World War II very complex electromechanical devices were in use by the Allied and Axis forces. The stories of these devices can be found in many books such as Hodges.[2] The need for a full-time, professional cryptographic force was recognized during and after WWII and led to the formation of the National Security Agency by Presidential memorandum signed by Truman. See Bamford[3] for a history of the NSA.

Except for a few hobbyists, cryptography was virtually unknown outside of diplomatic and military circles until the mid-seventies. During this period, as the use of computers, particularly by financial institutions, became more widespread, the need arose for a "public," (non-military or diplomatic) cryptographic system. In 1973 the National Bureau of Standards (now the National Institute of Standards and Technology) issued a request for proposals for a standard cryptographic algorithm. They received no suitable response at that time and reissued the request in 1974. IBM responded to the second request with their Lucifer system, which they had been developing for their own use. This algorithm was evaluated with the help of the NSA and eventually was adopted as the Data Encryption Standard (DES) in 1976. See Federal Information Processing Standard NBS FIPS PUB 46.

The controversy surrounding the selection of DES[4] stimulated academic interest in cryptography and cryptanalysis. This interest led to the discovery of many cryptanalytic techniques and eventually to the concept of public key cryptography. Public key cryptography is a technique that uses

distinct keys for encryption and decryption, only one of which need be secret. We will discuss this technique later in this chapter, as public key cryptography is more understandable once one has a firm understanding of conventional cryptography.

The 20 years since the announcement of DES and the discovery of public key cryptography have seen advances in computer technology and networking that were not even dreamed of in 1975. The Internet has created a demand for instantaneous information exchange in the military, government, and most importantly, private sectors that is without precedent. Our economic base, the functioning of our government, and our military effectiveness are more dependent on automated information systems than any country in the world. However, the very technology that created this dependence is its greatest weakness: the infrastructure is fundamentally vulnerable to attacks from individuals, groups, or nation-states that can easily deny service or compromise the integrity of information. The users of the Internet, especially those with economic interests, have come to realize that effective cryptography is a necessity.

THE BASICS OF MODERN CRYPTOGRAPHY

Since virtually all of modern cryptography is based on the use of digital computers and digital algorithms, we begin with a brief introduction to digital technology and binary arithmetic. All information in a computer is reduced to a representation as 1s and 0s. (Or the "on" and "off" state of an electronic switch.) All of the operations within the computer can be reduced to logical OR, EXCLUSIVE OR, and AND. Arithmetic in the computer (called binary arithmetic) obeys the rules shown in Exhibit 19.3 (represented by "addition" and "multiplication" tables):

⊕	0	1
0	0	1
1	1	0

⊗	0	1
0	0	0
1	0	1

Exhibit 19.3. Binary Arithmetic Rules

The symbol ⊕ is called modulo 2 addition and ⊗ is called modulo 2 multiplication. If we consider the symbol '1' as representing a logical value of TRUE and '0' as the logical value FALSE then ⊕ is equivalent to exclusive OR in logic (XOR) while ⊗ is equivalent to AND. For example, A XOR B is true only if A or B is TRUE but not both. Likewise, A AND B is true only when both A and B are TRUE.

All messages, both plain text and cipher text, may be represented by strings of 1s and 0s. The actual method used to digitize the message is not relevant to an understanding of cryptography so we will not discuss the details here.

We will consider two main classes of cryptographic algorithms:

- Stream Ciphers — which operate on essentially continuous streams of plain text, represented as 1s and 0s
- Block Ciphers — which operate on blocks of plain text of fixed size.

These two divisions overlap in that a block cipher may be operated as a stream cipher. Generally speaking, stream ciphers tend be implemented more in hardware devices, while block ciphers are more suited to implementation in software to execute on a general-purpose computer. Again, these guidelines are not absolute, and there are a variety of operational reasons for choosing one method over another.

STREAM CIPHERS

We illustrate a simple stream cipher in the table below and in Exhibit 19.4. Here the plain text is represented by a sequence of 1s and 0s. (The binary streams are to be read from right to left. That is, the right-most bit is the first bit in the sequence.) A keystream[5] generator produces a "random" stream of 1s and 0s that are added modulo 2, bit by bit, to the plain-text stream to produce the cipher-text stream.

The cryptovariable (key) is shown as entering the keystream generator. We will explain the nature of these cryptovariables later. There are many different mechanisms to implement the keystream generator, and the reader is referred to Schneier[6] for many more examples. In general, we may represent the internal operation as consisting of a finite state machine and a complex function. The finite state machine consists of a system state and a function (called the "next state" function) that cause the system to change state based on certain input.

The complex function operates on the system state to produce the keystream. Exhibit 19.5 shows the encryption operation. The decryption operation is equivalent; just exchange the roles of plain text and cipher text. This works because of the following relationships in modulo two addition: Letting p represent a plain-text bit, k a keystream bit, and c the cipher text bit

$$c = p \oplus k,$$

so, $$c \oplus k = (p \oplus k) \oplus k = p \oplus (k \oplus k) = p \oplus 0 = p,$$

since in binary arithmetic $x \oplus x$ is always 0. $(1 \oplus 1 = 0 \oplus 0 = 0)$.

Plain Text:	1	0	1	1	0	1	1	0	0
	⊙	⓪	⊕	⊕	⊕	⊙	⓪	⓪	⓪
Keystream	1	1	0	1	0	0	0	1	1
Cipher Text	0	1	1	0	0	1	1	1	1

Exhibit 19.4. Stream Cipher

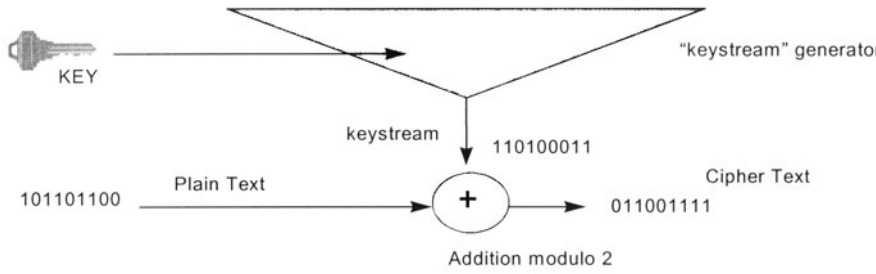


Exhibit 19.5. Stream Ciphers

These concepts are best understood with examples. Exhibit 19.6 shows a simple linear feedback shift register (LFSR). A LFSR is one of the simplest finite state machines and is used as a building block for many stream ciphers (see Schneier's text). In Exhibit 19.6, the four-stage register (shown here filled with 1s) represents the state. During operation, at each tick of the internal clock, the 4 bits shift to the right (the right-most bit is dropped), and the last 2 bits (before the shift) are added (mod 2) and placed in the left-most stage. In general, an LFSR may be of any length, n, and any of the individual stages may be selected for summing and insertion into the left-most stage. The only constraint is that the right-most bit should always be one of the bits selected for the feedback sum. Otherwise, the length is really n – 1, not n. Exhibit 19.6 shows the sequence of system states obtained from the initial value of 1111. In some systems, the initial value of the register is part of the cryptovariable.

Note that if we started the sequence with 0000, then all subsequent states would be 0000. This would not be good for cryptographic applications since the output would be constant. Thus the all-0 state is avoided. Note also that this four-stage register steps through $15 = 2^4 - 1$ distinct

Exhibit 19.6. Simple LFSR

states before repeating. Not all configurations of feedback will produce such a maximal sequence. If we number the stages in Exhibit 19.6 from left to right as 1,2,3,4, and instead of feeding back the sum of stages 3 and 4 we selected 2 and 4, then we would see a very different sequence. This example would produce 2 sequences (we call them cycles) of length 6, one cycle of length 3, and 1 of length 0. For example, starting with 1111 as before will yield:

$$1111 \rightarrow 0111 \rightarrow 0011 \rightarrow 1001 \rightarrow 1100 \rightarrow 1110 \rightarrow 1111$$

It is important to have as many states as possible produced by the internal state machine of the keystream generator. The reason is to avoid repeating the keystream. Once the keystream begins to repeat, the same plain text will produce the same cipher text. This is a cryptographic weakness and should be avoided. While one could select any single stage of the LFSR and use it as the keystream, this is not a good idea. The reason is that the linearity of the sequence of stages allows a simple cryptanalysis. We can avoid the linearity by introducing some more complexity into the system. The objective is to produce a keystream that looks completely random.[7] That is, the keystream will pass as many tests of statistical randomness as one cares to apply. The most important test is that knowledge of the algorithm and knowledge of a sequence of successive keystream bits does not allow a cryptanalyst to predict the next bit in the sequence. The complexity can often be introduced by using some nonlinear polynomial $f(a_1, a_2, \ldots, a_m)$ of a selection of the individual stages of the LFSR. Nonlinear means that some of the terms are multiplied together such as $a_1a_2 + a_3a_4 + \ldots a_{m-1}a_m$. The selection of which register stages are

associated with which inputs to the polynomial can be part of the cryptovariable (key). The reader is encouraged to refer to texts such as Schneier[6] for examples of specific stream-cipher implementations. Another technique for introducing complexity is to use multiple LFSRs and to select output alternately from each based on some pseudorandom process. For example, one might have three LFSRs and create the keystream by selecting bits from one of the two, based on the output of a third.

Some of the features that a cryptographer will design into the algorithm for a stream cipher include:

1. Long periods without a repetition.
2. Functional complexity — each keystream bit should depend on most or all of the cryptovariable bits.
3. Statistically unpredictable — given n successive bits from the keystream it is not possible to predict the n + 1st bit with a probability different from ½.
4. The keystream should be statistically unbiased — there should be as many 0s as 1s, as many 00s as 10s, 01s, and 11s, etc.
5. The keystream should not be linearly related to the cryptovariable.

We also note that in order to send and receive messages encrypted with a stream cipher the sending and receiving systems must satisfy several conditions. First, the sending and receiving equipment must be using identical algorithms for producing the keystream. Second, they must have the same cryptovariable. Third, they must start in the same state; and fourth, they must know where the message begins.

The first condition is trivial to satisfy. The second condition, ensuring that the two machines have the same cryptovariable, is an administrative problem (called key management) that we will discuss in a later section. We can ensure that the two devices start in the same state by several means. One way is to include the initial state as part of the cryptovariable. Another way is to send the initial state to the receiver at the beginning of each message. (This is sometimes called a message indicator, or initial vector.) A third possibility is to design the machines to always default to a specific state. Knowing where the beginning of the message is can be a more difficult problem, and various messaging protocols use different techniques.

BLOCK CIPHERS

A block cipher operates on blocks of text of fixed size. The specific size is often selected to correspond to the word size in the implementing computer, or to some other convenient reference (e.g., 8-bit ASCII text is conveniently processed by block ciphers with lengths that are multiples of 8 bits). Because the block cipher forms a one-to-one correspondence between input and output blocks it is nothing more or less than a permutation. If the blocks

are n bits long, then there are 2^n possible input blocks and 2^n possible output blocks. The relationship between the input and output defines a permutation. There are $(2^n)!$ possible permutations, so theoretically there are $(2^n)!$ possible block cipher systems on n bit blocks.[8]

A simple block cipher on 4-bit blocks is shown in Exhibit 19.7.

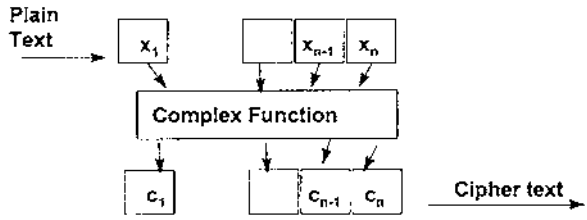


Exhibit 19.7. Block Ciphers

With such a prodigious number of possible block ciphers, one would think it a trivial matter to create one. It is not so easy. First of all, the algorithm has to be easy to describe and implement. Most of the $(2^n)!$ permutations can only be described by listing the entries in a table such as the one in Exhibit 19.8. For a 32-bit block cipher this table would have on the order of $10^{9.6}$ entries, which is quite impractical. Another consideration is that there needs to be a relation between the cryptovariable and the permutation. In most implementations, the cryptovariable selects a specific permutation from a wide class of permutations. Thus one would need as many tables as cryptovariables. We conclude from this that it is not easy to design good block ciphers.

The most well-known block cipher is the Data Encryption Standard, DES. The cryptovariable for DES is 64 bits, 8 of which are parity check bits. Consequently the cryptovariable is effectively 56 bits long. DES operates as follows: a 64-bit plain text block, after going through an initial permutation (which has no cryptographic significance) is split onto left and right halves, L_0 and R_0. These two halves are then processed as follows for i = 0, 1, …, 15

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} + f(R_{i-1}, K_i).$$

The blocks K_i are derived from the cryptovariable. The function f is a very complex function involving several expansions, compressions, and permutations by means of several fixed tables called the S-boxes and

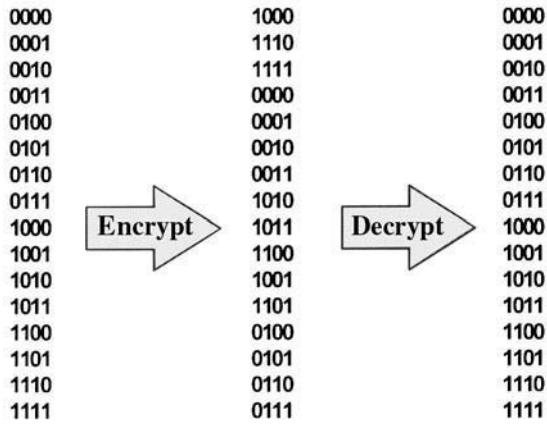


Exhibit 19.8. Simple Block Cipher

P-boxes. The reader is referred to FIPS PUB 46 for a detailed description of the S-boxes and P-boxes.

As was the case with the DES cryptovariable, there has been much discussion about the significance of the S-boxes. Some people have argued that the NSA designed the S-Boxes so as to include a "trap door" that would allow them to decrypt DES-encrypted messages at will. No one has been able to discover such a trap door. More recently it has been stated that the S-boxes were selected to minimize the danger from an attack called differential cryptanalysis.

Because of the widespread belief that the DES cryptovariable is too small, many have suggested that one encrypt a message twice with DES using two different cryptovariables. This "Double DES" is carried out in the following way. Represent the operation of DES encryption on message P and cryptovariable K as C = E(P; K); and the corresponding decryption as P = D(C; K) = D(E(P; K); K). The "Double DES" with cryptovariables K and K′ is

$$C = E(E(P; K); K')$$

Since each cryptovariable is 56 bits long, we have created an effective cryptovariable length of 56 + 56 = 112 bits. However, we shall see in the section on cryptanalysis that there is an attack on double-DES that requires about the same amount of computation as that required to attack a single DES. Thus double DES is really no more secure than single DES.

A third variant is triple DES, which applies the DES algorithm three times with two distinct cryptovariables. Let K and K′ be DES cryptovariables. Then triple DES is

$$C = E(D(E(P; K); K'); K).$$

That is, apply the encrypt function to P using the first cryptovariable, K. Then apply the decrypt function to the result using the second cryptovariable, K′. Since the decrypt function is using a different cryptovariable, the message is not decrypted; it is transformed by a permutation as in any block cipher. The final step is to encrypt once again with the encrypt function using the first key, K. By using the D in the middle, a triple DES implementation can be used to encrypt a single DES message when K = K′:

$$C = E(D(E(P; K); K); K) = E(P; K).$$

Thus, someone using triple DES is still able to communicate securely with persons using single DES. No successful attacks have been reported on triple DES that are any easier than trying all possible pairs of cryptovariables. In the next section we deal with cryptanalysis in more detail.

CRYPTANALYSIS

As we stated in the introduction, cryptography is the science of designing algorithms for encrypting messages. Cryptanalysis is the science (some would say art) of "breaking" the cryptographic systems. In the following we will try to explain just what "breaking" a cryptosystem means, as there are many misconceptions in the press.

There is an obvious analogy between cryptanalysis and cryptography and burglars and locks. As the locksmiths design better locks the burglars develop better ways to pick them. Likewise, as the cryptographer designs better algorithms the cryptanalyst develops new attacks. A typical design methodology would be to have independent design teams and attack teams. The design team proposes algorithms, and the attack teams tries to find weaknesses. In practice, this methodology is used in the academic world. Researchers publish their new algorithms, and the rest of the academic world searches for attacks to be published in subsequent papers. Each cycle provides new papers toward tenure.

Breaking or attacking a cryptosystem means recovering the plain-text message without possession of the particular cryptovariable (or key) used to encrypt that message. More generally, breaking the system means determining the particular cryptovariable (key) that was used. Although it is the message (or the information in the message) that the analyst really wants, possession of the cryptovariable allows the analyst to recover all of the messages that were encrypted in that cryptovariable. Since the cryptoperiod may be days or weeks, the analyst who recovers a cryptovariable will be able to recover many more messages than if he attacks a single message at a time.

Determining the specific details of the algorithm that was used to encrypt the message is generally not considered part of breaking an encryption system. In most cases, e.g., DES, the algorithm is widely known. Even many of the proprietary systems such as RC4 and RC5 have been published. Because it is very difficult to maintain the secrecy of an algorithm it is better to design the algorithm so that knowledge of the algorithm's details is still not sufficient to determine the cryptovariable used for a specific message without trying all possible cryptovariables.

Trying all cryptovariables is called a "brute force" or "exhaustion" attack. It is an attack that will always work as long as one is able to recognize the plain-text message after decryption. That is, in any attack you need to be able to decide when you have succeeded. One also has to be able to find the cryptovariable (and hence the message) in time for it to be of use. For example, in a tactical military environment, to spend one week to recover a message about an attack that will occur before the week is over will not be useful. Last, one has to be able to afford to execute the attack. One may often trade off time and computer power; an attack that may take one year on a PC might take only one day on 365 PCs. If one must have the message within a day for it to be valuable, but one does not have the funds to acquire or run 365 PCs, then one really doesn't have a viable attack.

Often a cryptanalyst might assume that she possesses matched plain and cipher text. This is sometimes possible in real systems because military and diplomatic messages often have stereotyped beginnings. In any case it is not a very restrictive condition and can help the cryptanalyst evaluate the cryptographic strength of an algorithm.

Let us look at a brute force attack on some system. We suppose that the cryptovariable has n binary bits (e.g., DES has n = 56). We suppose that we have a stream cipher and that we have matched plain and cipher text pairs P_i and C_i for I = 1, 2, … . For each possible cryptovariable there is some fixed amount of computation ("work") needed to encrypt a P_i and see if it results in the corresponding C_i. We can convert this work into the total number, W, of basic bit operations in the algorithm such as shifts, mod 2 additions, compares, etc. Suppose for definiteness that W = 1000 or 10^3.

There is a total of 2^n n-bit cryptovariables. For n = 56, 2^{56} is about $10^{16.8}$ or 72,000,000,000,000,000. If we select one of the possible cryptovariables and encrypt P_1 we have a 50:50 chance of getting C_1 since the only choices are 1 and 0. If we do not obtain C_1 we reject the selected cryptovariable as incorrect and test the next cryptovariable. If we do get C_1 then we must test the selected cryptovariable on P_2 and C_2. How many tests do we need to make in order to be sure that we have the correct cryptovariable? The answer is: at least 56. The rationale is that the probability of the wrong cryptovariable successfully matching 56 or more bits is 2^{-56}. Since we potentially have to try 2^{56} cryptovariables the expected number of cryptovariables passing all the

tests is $(2^{56})(2^{-56}) = 1$. With one "survivor" we may correctly assume it is the cryptovariable we want. If we tested only 2^{55} cryptovariables, then we would expect two survivors. (Cryptanalysts call a cryptovariable that passes all of the tests by chance a "non-causal survivor.") If we test a few more than 56, the expected number of non-causal survivors is much less than 1. Thus we can be sure that the cryptovariable that does successfully match the 56 P_i and C_i is the one actually used. In a block cipher, such as DES, testing one block is usually sufficient since a correct block has 64 correct bits.

A natural question is how long does it take to execute a brute force attack (or any other kind of attack for that matter). The answer depends on how much computational power is available to the analyst. And since we want cryptographic systems to be useful for many years we also need to know how much computational power will be available in years hence. Gordan Moore, one of the founders of Intel, once noted that processing speeds seem to double (or costs halved) every 18 months. This is equivalent to a factor of 10 increase in speed per dollar spent about every 5 years. This trend has continued quite accurately for many years and has come to be known as "Moore's law."

Using Moore's law we can make some predictions. We first introduce the idea of a MIPS year (M.Y.). This is the number of instructions a million-instruction-per-second computer can execute in one year. One M.Y. is approximately $10^{13.5}$ instructions. At today's prices, one can get a 50 MIPS PC for about $750. We can then estimate the cost of a MIPS year at about $750/50 or $15, assuming we can run the computer for one year.

Let's look at what this means in two examples. We consider two cryptographic systems. One with a 56-bit cryptovariable (e.g., DES) and the other a 40-bit cryptovariable. Note that 40 bits is the maximum cryptovariable length allowed for export by the U.S. government. We assume that each algorithm requires about 1000 basic instructions to test each cryptovariable. Statistics tells us that, on average, we may expect to locate the correct cryptovariable after testing about ½ of the cryptovariable space.

There are two perspectives: how much does it cost? And how long does it take? The cost may be estimated from:

$$(½)\ (1000N(15))/\text{M.Y.},$$

where N equals the number of cryptovariables (in the examples, either 2^{56} or 2^{40}), and M.Y. = $10^{13.5}$. The elapsed time requires that we make some assumptions as to the speed of processing. If we set K equal to the number of seconds in one year, and R the number of cryptovariables tested per second, we obtain the formula:

$$\text{Time (in years)} = (½)\ (N/KR).$$

The results are displayed in Exhibit 19.9.

YEAR	M.Y. Cost	On 56 bit cryptovariable	On 40 bit cryptovariable
1998	$15	$17 Million	$260
2003	$1.50	$1.7Million	$26
2008	$0.15	$170 thousand	$2.60

Number of cryptovariables tested per second	On 56 bit cryptovariable	On 40 bit cryptovariable
1,000	300 million years	17.5 years
1,000,000	300,000 years	6.2 days
1,000,000,000	300 years	9 minutes
1,000,000,000,000	109 days	0.5 seconds

Exhibit 19.9. Cost and Time for Brute Force Attack

One of the first public demonstrations of the accuracy of these estimates occurred during the summer of 1995. At that time a student at Ecole Polytechnique reported that he had "broken" an encrypted challenge message posted on the Web by Netscape. The message, an electronic transaction, was encrypted using an algorithm with a 40-bit cryptovariable. What the student did was to partition the cryptovariable space across a number of computers to which he had access and set them searching for the correct one. In other words he executed a brute force attack and he successfully recovered the cryptovariable used in the message. His attack ran for about 6 days and processed about 800,000 keys per second. While most analysts did not believe that a 40-bit cryptovariable was immune to a brute force attack, the student's success did cause quite a stir in the press. Additionally the student posted his program on a Web site so that anyone could copy the program and run the attack. At the RSA Data Security Conference, January 1997, it was announced that a Berkeley student using the idle time on a network of 250 computers was able to break the RSA challenge message, encrypted using a 40-bit key, in three and one-half hours.

More recently a brute force attack was completed against a DES message on the RSA Web page. We quote from the press release of the DES Challenge team (found on www.frii.com/~rtv/despr4.htm):

> LOVELAND, COLORADO (June 18, 1997). Tens of thousands of computers, all across the U.S. and Canada, linked together via the Internet in an unprecedented cooperative supercomputing effort to decrypt a message encoded with the government-endorsed Data Encryption Standard (DES).
>
> Responding to a challenge, including a prize of $10,000, offered by RSA Data Security, Inc., the DESCHALL effort successfully decoded RSA's secret message.

> According to Rocke Verser, a contract programmer and consultant who developed the specialized software in his spare time, "Tens of thousands of computers worked cooperatively on the challenge in what is believed to be one of the largest supercomputing efforts ever undertaken outside of government."
>
> Using a technique called "brute-force," computers participating in the challenge simply began trying every possible decryption key. There are over 72 quadrillion keys (72,057,594,037,927,936). At the time the winning key was reported to RSADSI, the DESCHALL effort had searched almost 25% of the total. At its peak over the recent weekend, the DESCHALL effort was testing 7 billion keys per second.
>
> … And this was done with "spare" CPU time, mostly from ordinary PCs, by thousands of users who have never even met each other.

In other words, the DESCHALL worked as follows. Mr. Verser developed a client-server program that would try all possible keys. The clients were available to any and all who wished to participate. Each participant downloaded the client software and set it executing on their PC (or other machine). The client would execute at the lowest priority in the client PC and so did not interfere with the participant's normal activities. Periodically the client would connect to the server over the Internet and would receive another block of cryptovariables to test. With tens of thousands of clients it only took 4 months to hit the correct cryptovariable.

Another RSA Data Security Inc.'s crypto-cracking contest, launched in March 1997, was completed in October 1997. A team of some 4000 programmers from across the globe, calling themselves the "Bovine RC5 Effort," has claimed the $10,000 prize for decoding a message encrypted in 56-bit -RC5 code. The RC5 effort searched through 47 percent of the possible keys before finding the one used to encrypt the message.

RSA Data Security Inc. sponsored the contest to prove its point that 128-bit encryption must become the standard. Under current U.S. policy, software makers can sell only 40-bit key encryption overseas, with some exceptions available for 56-bit algorithms.

A second DES challenge was solved in February 1998 and took 39 days (see Exhibit 19.10). In this challenge, the participants had to test about 90 percent of the keyspace.

This chapter has focused mostly on brute force attacks. There may be, however, other ways to attack an encryption system. These other methods may be loosely grouped as analytic attacks, statistical attacks, and implementation attacks.

Analytic attacks make use of some weakness in the algorithm that enables the attacker to effectively reduce the complexity of the algorithm

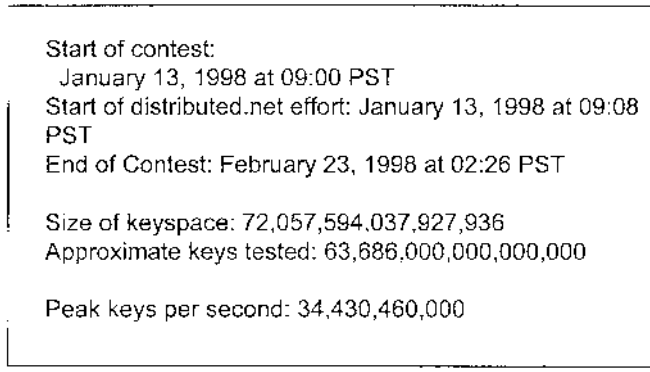


Exhibit 19.10. RSA Project Statistics

through some algebraic manipulation. We will see in the section on public key systems, that the RSA public key algorithm can be attacked by factoring with much less work than brute force. Another example of an analytic attack is the attack on double DES.

Double DES, you recall, may be represented by:

$$C = E(E(P; K); L),$$

where K and L are 56-bit DES keys. We assume that we have matched plain and cipher text pairs C_i, P_i. Begin by noting that if $X = E(P; K)$. Then $D(C; L) = X$. Fix a pair C_1, P_1, and make a table of all 2^{56} values of $D(C_1; L)$ as L ranges through all 2^{56} possible DES keys. Then try each K in succession, computing $E(P_1; K)$ and looking for matches with the values of $D(C_1; L)$ in the table. Each pair K, L for which $E(P_1; K)$ matches $D(C_1; L)$ in the table is a possible choice of the sought-for cryptovariable. Each pair passing the test is then tested against the next plain–cipher pair P_2, C_2.

The chance of a non-causal match (a match given that the pair K, L is not the correct cryptovariable) is about 2^{-64}. Thus of the 2^{112} pairs K, L, about $2^{(112-64)} = 2^{48}$ will match on the first pair P_1, C_1. Trying these on the second block P_2, C_2 and only $2^{(48-64)} = 2^{-16}$ of the non-causal pairs will match. Thus, the probability of the incorrect cryptovariable passing both tests is about $2^{-16} \sim 0$. And the probability of the correct cryptovariable passing both tests is 1.

The total work to complete this attack (called the "meet in the middle" attack) is proportional to $2^{56} + 2^{48} = 2^{56}(1+2^{-8}) \sim 2^{56}$. In other words an attack on double DES has about the same work as trying all possible single DES keys. So there is no real gain in security with double DES.

Statistical attacks make use of some statistical weakness in the design. For example, if there is a slight bias toward 1 or 0 in the keystream, one can sometimes develop an attack with less work than brute force. These attacks are too complex to describe in this short chapter.

The third class of attacks is implementation attacks. Here one attacks the specific implementation of the encryption protocol, not simply the cryptographic engine. A good example of this kind of attack was in the news in late summer 1995. The target was Netscape; and this time the attack was against the 128-bit cryptovariable. Several Berkeley students were able to obtain source code for the Netscape encryption package and were able to determine how the system generated cryptovariables. The random generator was given a seed value that was a function of certain system clock values.

The students discovered that the uncertainty in the time variable that was used to seed the random-number generator was far less than the uncertainty possible in the whole cryptovariable space. By trying all possible seed values they were able to guess the cryptovariable with a few minutes of processing time. In other words, the implementation did not use a randomization process that could, in principle, produce any one of the 2^{128} possible keys. Rather it was selecting from a space more on the order of 2^{20}. The lesson here is that even though one has a very strong encryption algorithm and a large key space, a weak implementation could still lead to a compromise of the system.

KEY (CRYPTOVARIABLE) MANAGEMENT

We have noted in the previous sections that each encryption system requires a key (or cryptovariable) to function and that all of the secrecy in the encryption process is maintained in the key. Moreover, we noted that the sending and receiving party must have the same cryptovariable if they are to be able to communicate. This need translates to a significant logistical problem.

The longer a cryptovariable is used the more likely it is to be compromised. The compromise may occur through a successful attack or, more likely, the cryptovariable may be stolen by or sold to an adversary. Consequently, it is advisable to change the variable frequently. The frequency of change is a management decision based on the perceived strength of the algorithm and the sensitivity of the information being protected.

All communicating parties must have the same cryptovariable. Thus you need to know in advance with whom you plan to exchange messages. If a person needs to maintain privacy among a large number of different persons, then one would need distinct cryptovariables for each possible

communicating pair. In a 1000-person organization, this would amount to almost one million keys.

Next, the keys must be maintained in secrecy. They must be produced in secret, and distributed in secret, and held by the users in a protected area (e.g., a safe) until they are to be used. Finally they must be destroyed after being used.

For centuries, the traditional means of distributing keys was through a trusted courier. A government organization would produce the cryptovariables. And couriers, who have been properly vetted and approved, would distribute the cryptovariables. A rigorous audit trail would be maintained of manufacture, distribution, receipt, and destruction. Careful plans and schedules for using the keys would be developed and distributed.

This is clearly a cumbersome, expensive, and time-consuming process. Moreover the process was and is subject to compromise. Many of history's spies were also guilty of passing cryptovariables (as well as other state secrets) to the enemy.

As our communications systems became more and more dependent on computers and communication networks, the concept of a key distribution center was developed. The key distribution center concept is illustrated in Exhibit 19.11. The operation is as follows: Initially each user, A, B, …, is given (via traditional distribution) a user-unique key that we denote by K_A, K_B, etc. These cryptovariables will change only infrequently, which reduces the key distribution problem to a minimum. The KDC maintains a copy of each user-unique key. When A calls B, the calling protocol first contacts the KDC and tells it that user A is sending a message to user B. The KDC then generates a random "session key," K, i.e., a cryptovariable that will be used only for this communicating session between A and B. The KDC encrypts K in user A's unique cryptovariable, $E(K; K_A)$ and sends this to A. User A decrypts this message obtaining K. The KDC likewise encrypts K in user B's unique cryptovariable, $E(K; K_B)$ and sends this result to B. Now A and B (and no other party) have K, which they use as the cryptovariable for this session.

A session here may be a telephone call or passing a message through a packet switch network; the principles are the same. In practice the complete exchange is done in seconds and is completely transparent to the user.

The KDC certainly simplifies the distribution of cryptovariables. Only the user-unique keys need to be distributed in advance, and only infrequently. The session key only exists for the duration of the message so there is no danger that the key might be stolen and sold to an unauthorized person at some later date. But the KDC must be protected, and one still has

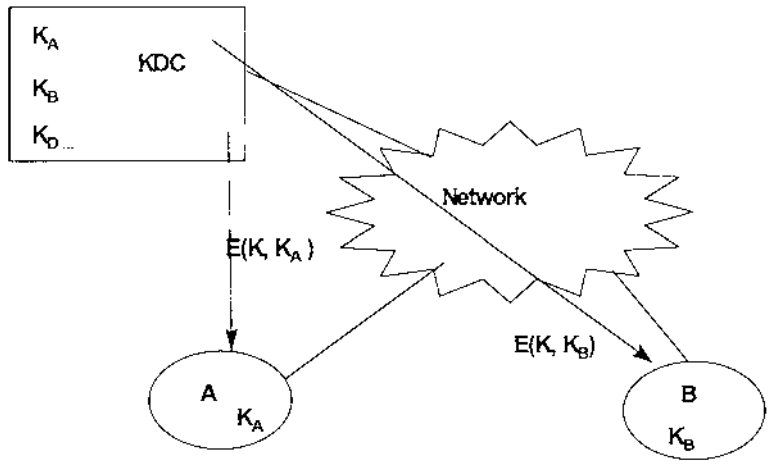


Exhibit 19.11. Key Distribution Center

to know with whom they will be communicating. The KDC will not help if one needs to send an electronic mail message to some new party (i.e., a party unknown to the KDC) for example.

It is clear that cryptovariable (or key) management is difficult and does not provide much in the way of flexibility. Many people have wondered if it would be possible to develop an encryption system that did not require secret keys; a system where one could have a directory of public keys. When you wanted to send an encrypted message to someone, you would look up that person's cryptovariable in a "telephone book," encrypt the message, and send it. And no one intercepting the message would be able to decrypt it except the intended recipient. Can such a system be designed? The answer is yes. It is called public key cryptography.

PUBLIC KEY CRYPTOGRAPHY

The concept of public key cryptography was first discovered and publicly announced by Whitfield Diffie and Martin Hellman (and independently by Ralph Merkle) in 1976. Adm. Bobby Inmann, a former director of the National Security Agency once stated publicly that NSA knew of the idea for many years prior to the publication by Diffie and Hellman.

The public key concept is rather simple (as are most great ideas, once they are explained). We assume that we have two special functions, E and D, that can operate on messages M. (In actual applications large integers will represent the messages, and E and D will be integer functions.) We assume that E and D satisfy the following conditions:

1. $D(E(M)) = M$
2. $E(D(M)) = M$
3. Given E it is not possible to determine D
4. Given D it is not possible to determine E.

The use of the function E in encryption is straightforward. We assume that each person, A, B, C, has pairs of functions E_A, D_A, E_B, D_B, … that satisfy the conditions 1., 2., and 3. given above. Each user X makes their E_X publicly available but keeps their D_X secret and known only to themselves. When A wants to send a message, M, to B, A looks up E_B in the published list and computes $E_B(M)$. By property 2, $D_B(E_B(M) = M$ so B can decrypt the message. From property 3, no person can determine D_B from knowledge of E_B so no one but B can decipher the message.

The functions can also be used to sign messages. Perhaps A wants to send a message M to B and she does not care if anyone else sees the message, but she does want B to know that it really came from her. In this case A computes $D_A(M)$, called a signature, and sends it along with M. When B gets these two messages, he looks up A's function E_A and computes $E_A(D_A(M))$ and obtains M from property 2. If this computed M agrees with the message sent as M, then B is sure that it came from A. Why? Because no one else has or can compute D_A except A and the likelihood of someone producing a fictitious X such that $E_A(X) = M$ is infinitesimally small.

Now suppose A wants to send B a secret message and sign it. Let M be the message. A first computes a "signature" $S = D_A(M)$ and concatenates this to the message M, forming M, S. A then encrypts both the message and the signature, $E_B(M, S)$ and sends it to B. B applies D_B to $E_B(M, S)$ obtaining $D_B(E_B(M, S)) = M, S$. B then computes $E_A(S) = E_A(D_A(M)) = M$ and compares it to the message he decrypted. If both versions of M are the same, he can be assured that A sent the message.

The question the reader should be asking is "Do such functions exist?" The answer is yes, if we relax what we mean by conditions 3 and 4 above. If we only require that it be computationally infeasible to recover D from E (and vice versa) then the functions can be shown to exist. The most well-known example is the RSA algorithm, named for its discoverers, Rivest, Shamir, and Adleman.

A description of RSA requires a small amount of mathematics that we will explain as we proceed. We start with two large prime numbers, p and q. By large we mean they contain hundreds of digits. This is needed in order to meet conditions 3 and 4. A prime number, you recall, is a number that has no divisors except the number itself and 1. (In dealing with integers when we say a divides b we mean that there is no remainder; i.e., $b = ac$ for some integer c.) The numbers 2, 3, 7, 11, 13, 17 are all prime. The number 2 is the only even prime. All other primes must be odd numbers.

We then define a number n as the product of p and q:

$$n = pq$$

We also define a number t as:

$$t = (p - 1)(q - 1)$$

As an example, take p = 3 and q = 7. (These are not large primes, but the mathematics is the same.) Then n = 21 and t = 12. The next step in the construction of RSA is to select a number e that has no common divisors with t. (In this case e and t are said to be relatively prime.) In our numerical example we may take e = 5 since 5 and 12 have no common divisors. Next we must find an integer d such that ed-1 is divisible by t. (This is denoted by ed = 1 mod t.) Since 5*5 – 1 = 25 – 1 = 24 = 2*12 = 2*t, we may take d = 5. (In most examples e and d will not be the same.)

The numbers d, p, and q are kept secret. They are used to create the D function. The numbers e and n are used to create the E function. The number e is usually called the public key and d the secret key. The number n is called the modulus. Once p and q are used to produce n and t, they are no longer needed and may be destroyed, but should never be made public.

To encrypt a message, one first converts the message into a string of integers, m_1, m_2, … all smaller than n. We then compute:

$$c_i = E(m_i) = m_i^e \bmod n$$

This means that we raise m_i to the e^{th} power and then divide by n. The remainder is $c_i = E(m_i)$. In our example, we suppose that the message is m_1 = 9. We compute:

$$c_1 = 9^5 \bmod 21$$

$$= 59049 \bmod 21$$

Because 59049 = 89979*21 + 18, we conclude that c_1 = 18 mod 21.

The decryption, or D function, is defined by:

$$D(c_i) = c_i^d \bmod n$$

In our example,

$$18^d \bmod n$$

$$= 18^5 \bmod 21$$

$$= 1889668 \bmod 21$$

As 1889568 = 889979*21 + 9, we conclude that D(18) = 9, the message we started with.

To demonstrate mathematically that the decryption function always works to decrypt the message (i.e., that properties 1 and 2 above hold) requires a result from number theory called Euler's generalization of Fermat's little theorem. The reader is referred to any book on number theory for a discussion of this result.

The security of RSA depends on the resistance of n to being factored. Since e is made public, anyone who knows the corresponding d can decrypt any message. If one can factor n into its two prime factors, p and q, then one can compute t and then easily find d. Thus it is important to select integers p and q such that it is not likely that someone can factor the product n. In 1983, the best factoring algorithm and the best computers could factor a number of about 71 decimal (235 binary) digits. By 1994, 129 digit (428 bits) numbers were being factored. Current implementations of RSA generate p and q on the order 256 to 1024 bits so that n is about 512 to 2048 bits.

The reader should note that attacking RSA by factoring the modulus n is a form of algebraic attack. The algebraic weakness is that the factors of n lead to a discovery of the "secret key." A brute force attack, by definition, would try all possible values for d. Since d is hundreds of digits long, the work is on the order of 10^{100}, which is a prodigiously large number. Factoring a number, n, takes at most on the order of square root of n operations or about 10^{50} for a 100-digit number. While still a very large number it is a vast improvement over brute force. There are, as we mentioned, factoring algorithms that are much smaller, but still are not feasible to apply to numbers of greater than 500 bits with today's technology, or with the technology of the near future.

As you can see from our examples, using RSA requires a lot of computation. As a result, even with special purpose hardware, RSA is slow; too slow for many applications. The best application for RSA and other public key systems is as key distribution systems.

Suppose A wants to send a message to B using a conventional private key system such as DES. Assuming that B has a DES device, A has to find some way to get a DES cryptovariable to B. She generates such a key, K, through some random process. She then encrypts K using B's public algorithm, $E_B(K)$ and sends it to B along with the encrypted message $E_{DES}(M; K)$. B applies his secret function D_B to $E_B(K)$ and recovers K, which he then uses to decrypt $E_{DES}(M; K)$.

This technique greatly simplifies the whole key management problem. We no longer have to distribute secret keys to everyone. Instead, each person has a public key system that generates the appropriate E and D functions. Each person makes the E public, keeps D secret and we're done. Or are we?

The Man-in-the-Middle

Unfortunately there are no free lunches. If a third party can control the public listing of keys, or E functions, that party can masquerade as both ends of the communication.

We suppose that A and B have posted their E_A and E_B, respectively, on a public bulletin board. Unknown to them, C has replaced E_A and E_B with E_C, his own encryption function. Now when A sends a message to B, A will encrypt it as $E_C(M)$ although he believes he has computed $E_B(M)$. C intercepts the message and computes $D_C(E_C\ (M)) = M$. He then encrypts it with the real E_B and forwards the result to B. B will be able to decrypt the message and is none the wiser. Thus this man in the middle will appear as B to A and as A to B.

The way around this is to provide each public key with an electronically signed signature (a certificate) attesting to the validity of the public key and the claimed owner. The certificates are prepared by an independent third party known as a certificate authority (e.g., VeriSign). The user will provide a public key (E function) and identification to the certificate authority (CA). The CA will then issue a digitally signed token binding the customer's identity to the public key. That is, the CA will produce $D_{CA}(ID_A, E_A)$. A person, B, wishing to send a message to A will obtain A's public key, E_A and the token $D_{CA}(ID_A, E_A)$. Since the CA's public key will be publicized, B computes $E_{CA}(D_{CA}(ID_A, E_A)) = ID_A, E_A$. Thus B, to the extent that he can trust the certification authority, can be assured that he really has the public key belonging to A and not an impostor.

There are several other public key algorithms, but all depend in one way or another on difficult problems in number theory. The exact formulations are not of general interest since an implementation will be quite transparent to the user. The important user issue is the size of the cryptovariable, the speed of the computation, and the robustness of the implementation. However, there is a new implementation that is becoming popular and deserves some explanation.

ELLIPTIC CURVE CRYPTOGRAPHY

A new public key technique based on elliptic curves has recently become popular. To explain this new process requires a brief digression. Recall from the previous section, that the effectiveness of public key algorithms depend on the existence of very difficult problems in mathematics. The security of RSA depends, for example, on the difficulty of factoring large numbers. While factoring small numbers is a simple operation, there are only a few (good) known algorithms or procedures for factoring large integers, and these still take prodigiously long times when factoring numbers that are hundreds of digits long. Another difficult mathematical problem is called

the discrete logarithm problem. Given a number b, the base, and x, the logarithm, one can easily compute b^x or b^x mod N for any N. It turns out to be very difficult to solve the reverse problem for large integers. That is, given a large integer y and a base b, find x so that $b^x = y$ Mod N. The known procedures (algorithms) require about the same level of computation as finding the factors of a large integer. Diffie and Hellman[9] exploited this difficulty to define their public key distribution algorithm.

Diffie and Hellman Key Distribution

Suppose that Sarah and Tanya want to exchange a secret cryptovariable for use in a conventional symmetric encryption system, say a DES encryption device. Sarah and Tanya together select a large prime p and a base b. The numbers p and b are assumed to be public knowledge. Next Sarah chooses a number s and keeps it secret. Tanya chooses a number t and keeps it secret. The numbers s and t must be between 1 and p-1. Sarah and Tanya then compute (respectively):

$$x = b^s \text{ Mod } p \text{ (Sarah)}$$

$$y = b^t \text{ Mod } p \text{ (Tanya)}$$

In the next step of the process Sarah and Tanya exchange the numbers x and y; Tanya sends y to Sarah, and Sarah sends x to Tanya. Now Sarah can compute

$$y^s = b^{ts} \text{ Mod } p$$

And Tanya can compute

$$x^t = b^{st} \text{ Mod } p$$

But,

$$b^{ts} \text{ Mod } p = b^{st} \text{ Mod } p = K$$

which becomes their common key. In order for a third party to recover K, that party must solve the discrete logarithm problem to recover s and t. (To be more precise, solving the discrete logarithm problem is sufficient to recover the key, but it might not be necessary. It is not known if there is another way to find b^{st} given b^s and b^t. It is conjectured that the latter problem is at least as difficult as the discrete logarithm problem.) The important fact regarding the Diffie-Hellman key exchange is that it applies to any mathematical object known as an Abelian group. (See Exhibit 19.12.)

Now we can get into the idea of elliptic curve cryptography, at least at a high level. An elliptic curve is a collection of points in the x-y plane that satisfy an equation of the form

$$y^2 = x^3 + ax + b \text{ . (1)}$$

GROUPS:
A group is a collection of elements, G, together with an operation * (called a "product" or a "sum") that assigns to each pair of elements x, y in G a third element z = x*y. The operation must have an identity element e with e*x = x*e = x for all x in G. Each element must have an inverse with respect to this identity. That is, for each x there is an x' with x*x' = e = x'*x. Last, the operation must be associative. If it is also true that x*y = y*x for all x and y in G, the group is said to be commutative, or Abelian. (In this case the operation is often written as +.

Exhibit 19.12. Definition of Abelian Groups

The elements a and b can be real numbers, imaginary numbers, or elements from a more general mathematical object known as a field. As an example, if we take a = -1 and b = 0. The equation is:

$$y^2 = x^3 - x \ . \ (2)$$

A graph of this curve is shown in Exhibit 19.13. It turns out that the points of this curve (those pairs (x, y) that satisfy the equation 2) can form a group under a certain operation. Given two points P = (x, y) and Q = (x′, y′) on the curve we can define a third point R = (x,″ y″) on the curve called the "sum" of P and Q. Furthermore this operation satisfies all of the requirements for a group. Now that we have a group we may define a Diffie-Hellman key exchange on this group. Indeed, any cryptographic algorithm that may be defined in a general group can be instantiated in the group defined on an elliptic curve. For a given size key, implementing an elliptic curve system seems to be computationally faster than the equivalent RSA. Other than the speed of the implementation there does not appear to be any advantage for using elliptic curves over RSA. RSA Data Security Inc. includes an elliptic curve implementation in their developer's kit (BSAFE) but they strongly recommend that the technique not be used except in special circumstances. Elliptic curve cryptographic algorithms have been

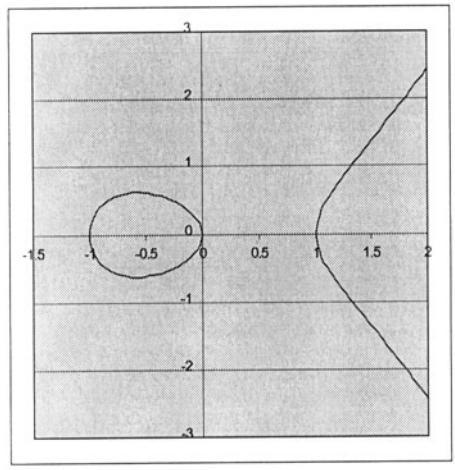


Exhibit 19.13. Graph of Elliptic Curve

subjected to significantly less analysis than the RSA algorithm so it is difficult to state with any confidence that elliptic curves are as secure or more secure than RSA. See Koblitz[10] for a complete discussion.

CONCLUSIONS

This short chapter presented a quick survey of some basic concepts in cryptography. No attempt was made to be comprehensive; the object was to help the reader better understand some of the reports about encryption and "breaking encryption systems" that often appear in the trade press and newspapers. The reader is referred to any of the many fine books that are available for more detail on any of the topics presented.

Notes

1. Kahn, David: *The Codebreakers; The Comprehensive History of Secret Communication from Ancient Times to the Internet*, Scribner, 1996.
2. Hodges, A., *Alan Turing: The Enigma of Intelligence*, Simon and Schuster, 1983.
3. Bamford, J., *The Puzzle Palace*, Houghton Mifflin, 1982.
4. Many thought that NSA had implanted a "trap door" that would allow the government to recover encrypted messages at will. Others argued that the cryptovariable length (56 bits) was too short.
5. The reader is cautioned not to confuse "keystream" with key. The term is used for historical reasons and is not the "key" for the algorithm. It is for this reason that we prefer the term "cryptovariable."
6. Schneier, B., *Applied Cryptography*, John Wiley, 1996.
7. The output cannot be truly random since the receiving system has to be able to produce the identical sequence.

8. For n = 7, $2^n!$ is about 10^{215}. The case n=8 is more than I can calculate. Clearly, there is no lack of possible block ciphers.
9. Diffie, W. and M. E. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory IT-22* (1976) 644-654.
10. Koblitz, Neil, *A Course in Number Theory and Cryptography*, Second Edition, Springer-Verlag, 1994.

111

Steganography: The Art of Hiding Messages

Mark Edmead, CISSP, SSCP, TICSA

Recently, there has been an increased interest in steganography (also called stego). We have seen this technology mentioned during the investigation of the September 11 attacks, where the media reported that the terrorists used it to hide their attack plans, maps, and activities in chat rooms, bulletin boards, and Web sites. Steganography had been widely used long before these attacks and, as with many other technologies, its use has increased due to the popularity of the Internet.

The word *steganography* comes from the Greek, and it means covered or secret writing. As defined today, it is the technique of embedding information into something else for the sole purpose of hiding that information from the casual observer. Many people know a distant cousin of steganography called watermarking — a method of hiding trademark information in images, music, and software. Watermarking is not considered a true form of steganography. In stego, the information is hidden in the image; watermarking actually adds something to the image (such as the word *Confidential*), and therefore it becomes part of the image. Some people might consider stego to be related to encryption, but they are not the same thing. We use encryption — the technology to translate something from readable form to something unreadable — to protect sensitive or confidential data. In stego, the information is not necessarily encrypted, only hidden from plain view.

One of the main drawbacks of using encryption is that with an encrypted message — although it cannot be read without decrypting it — it is recognized as an encrypted message. If someone captures a network data stream or an e-mail that is encrypted, the mere fact that the data is encrypted might raise suspicion. The person monitoring the traffic may investigate why, and use various tools to try to figure out the message's contents. In other words, encryption provides confidentiality but not secrecy. With steganography, however, the information is hidden; and someone looking at a JPEG image, for instance, would not be able to determine if there was any information within it. So, hidden information could be right in front of our eyes and we would not see it.

In many cases, it might be advantageous to use encryption and stego at the same time. This is because, although we can hide information within another file and it is not visible to the naked eye, someone can still (with a lot of work) determine a method of extracting this information. Once this happens, the hidden or secret information is visible for him to see. One way to circumvent this situation is to combine the two — by first encrypting the data and then using steganography to hide it. This two-step process adds additional security. If someone manages to figure out the steganographic system used, he would not be able to read the data he extracted because it is encrypted.

Hiding the Data

There are several ways to hide data, including data injection and data substitution. In data injection, the secret message is directly embedded in the host medium. The problem with embedding is that it usuallymakes the

EXHIBIT 111.1 Eight-Bit Pixel

1	1	0	0	1	1	0	1

host file larger; therefore, the alteration is easier to detect. In substitution, however, the normal data is replaced or substituted with the secret data. This usually results in very little size change for the host file. However, depending on the type of host file and the amount of hidden data, the substitution method can degrade the quality of the original host file.

In the article "Techniques for Data Hiding," Walter Bender outlines several restrictions to using stego:

- The data that is hidden in the file should not significantly degrade the host file. The hidden data should be as imperceptible as possible.
- The hidden data should be encoded directly into the media and not placed only in the header or in some form of file wrapper. The data should remain consistent across file formats.
- The hidden (embedded) data should be immune to modifications from data manipulations such as filtering or resampling.
- Because the hidden data can degrade or distort the host file, error-correction techniques should be used to minimize this condition.
- The embedded data should still be recoverable even if only portions of the host image are available.

Steganography in Image Files

As outlined earlier, information can be hidden in various formats, including text, images, and sound files. In this chapter, we limit our discussion to hidden information in graphic images. To better understand how information can be stored in images, we need to do a quick review of the image file format. A computer image is an array of points called pixels (which are represented as light intensity). Digital images are stored in either 24- or 8-bit pixel files. In a 24-bit image, there is more room to hide information, but these files are usually very large in size and not the ideal choice for posting them on Web sites or transmitting over the Internet. For example, a 24-bit image that is 1024×768 in size would have a size of about 2 MB. A possible solution to the large file size is image compression. The two forms of image compression to be discussed are lossy and lossless compression. Each one of these methods has a different effect on the hidden information contained within the host file. Lossy compression provides high compression rates, but at the expense of data image integrity loss. This means the image might lose some of its image quality. An example of a lossy compression format is JPEG (Joint Photographic Experts Group). Lossless, as the name implies, does not lose image integrity, and is the favored compression used for steganography. GIF and BMP files are examples of lossless compression formats.

A pixel's makeup is the image's raster data. A common image, for instance, might be 640×480 pixels and use 256 colors (eight bits per pixel).

In an eight-bit image, each pixel is represented by eight bits, as shown in Exhibit 111.1. The four bits to the left are the most-significant bits (MSB), and the four bits to the right are the least-significant bits (LSB). Changes to the MSB will result in a drastic change in the color and the image quality, while changes in the LSB will have minimal impact. The human eye cannot usually detect changes to only one or two bits of the LSB. So if we hide data in any two bits in the LSB, the human eye will not detect it. For instance, if we have a bit pattern of 11001101 and change it to 11001100, they will look the same. This is why the art of steganography uses these LSBs to store the hidden data.

A Practical Example of Steganography at Work

To best demonstrate the power of steganography, Exhibit 111.2 shows the host file before a hidden file has been introduced. Exhibit 111.3 shows the image file we wish to hide. Using a program called Invisible Secrets 3, by NeoByte Solution, Exhibit 111.3 is inserted into Exhibit 111.2. The resulting image file is shown in Exhibit 111.4. Notice that there are no visual differences to the human eye. One significant difference is in the size of the resulting image. The size of the original Exhibit 111.2 is 18 kb. The size of Exhibit 111.3 is 19 kb. The size of the resulting stego-file is 37 kb. If the size of the original file were known, the size of the new file

EXHIBIT 111.2 Unmodified image.

EXHIBIT 111.3 Image to be hidden in Exhibit 111.2.

would be a clear indication that something made the file size larger. In reality, unless we know what the sizes of the files should be, the size of the file would not be the best way to determine if an image is a stego carrier. A practical way to determine if files have been tampered with is to use available software products that can take a snapshot of the images and calculate a hash value. This baseline value can then be periodically checked for changes. If the hash value of the file changes, it means that tampering has occurred.

Practical (and Not So Legal) Uses for Steganography

There are very practical uses for this technology. One use is to store password information on an image file on a hard drive or Web page. In applications where encryption is not appropriate (or legal), stego can be used

EXHIBIT 111.4 Image with Exhibit 111.3 inserted into Exhibit 111.2.

for covert data transmissions. Although this technology has been used mainly for military operations, it is now gaining popularity in the commercial marketplace. As with every technology, there are illegal uses for stego as well. As we discussed earlier, it was reported that terrorists use this technology to hide their attacks plans. Child pornographers have also been known to use stego to illegally hide pictures inside other images.

Defeating Steganography

Steganalysis is the technique of discovering and recovering the hidden message. There are terms in steganography that are closely associated with the same terms in cryptography. For instance, a steganalyst, like his counterpart a cryptanalyst, applies steganalysis in an attempt to detect the existence of hidden information in messages. One important — and crucial — difference between the two is that in cryptography, the goal is not to detect if something has been encrypted. The fact that we can see the encrypted information already tells us that it is. The goal in cryptanalysis is to decode the message. In steganography, the main goal is first to determine if the image has a hidden message and to determine the specific steganography algorithm used to hide the information. There are several known attacks available to the steganalyst: stego-only, known cover, known message, chosen stego, and chosen message. In a stego-only attack, the stego host file is analyzed. A known cover attack is used if both the original (unaltered) media and the stego-infected file are available. A known message attack is used when the hidden message is revealed. A chosen stego attack is performed when the algorithm used is known and the stego host is available. A chosen message attack is performed when a stego-media is generated using a predefined algorithm. The resulting media is then analyzed to determine the patterns generated, and this information is used to compare it to the patterns used in other files. This technique will not extract the hidden message, but it will alert the steganalyst that the image in question does have embedded (and hidden) information.

Another attack method is using dictionary attacks against steganographic systems. This will test to determine if there is a hidden image in the file. All of the stenographic systems used to create stego images use some form of password validation. An attack could be perpetrated on this file to try to guess the password and determine what information had been hidden. Much like cryptographic dictionary attacks, stego dictionary attacks can be performed as well. In most steganographic systems, information is embedded in the header of the image file that contains, among other things, the length of the hidden message. If the size of the image header embedded by the various stego tools is known, this information could be used to verify the correctness of the guessed password.

Protecting yourself against steganography is not easy. If the hidden text is embedded in an image, and you have the original (unaltered) image, a file comparison could be made to see if they are different. This comparison would not be to determine if the size of the image has changed — remember, in many cases the image size does not change. However, the data (and the pixel level) does change. The human eye usually cannot easily detect subtle changes — detection beyond visual observation requires extensive analysis. Several techniques are used to do this. One is the use of stego signatures. This method involves analysis of many different types of untouched images, which are then compared to the stego images. Much like the analysis of viruses using signatures, comparing the stego-free images to the stego-images may make it possible to determine a pattern (signature) of a particular tool used in the creation of the stego-image.

Summary

Steganography can be used to hide information in text, video, sound, and graphic files. There are tools available to detect steganographic content in some image files, but the technology is far from perfect. A dictionary attack against steganographic systems is one way to determine if content is, in fact, hidden in an image.

Variations of steganography have been in use for quite some time. As more and more content is placed on Internet Web sites, the more corporations — as well as individuals — are looking for ways to protect their intellectual properties. Watermarking is a method used to mark documents, and new technologies for the detection of unauthorized use and illegal copying of material are continuously being improved.

References

W. Bender, D. Gruhl, N. Morimoto, and A. Lu, Techniques for data hiding, *IBM Syst. J.*, 35, 3–4, 313–336, February 1996.

Additional Sources of Information

http://www.cs.uct.ac.za/courses/CS400W/NIS/papers99/dsellars/ stego.html — Great introduction to steganography by Duncan Sellars.

http://www.jjtc.com/Steganography/ — Neil F. Johnson's Web site on steganography. Has other useful links to other sources of information.

http://stegoarchive.com/ — Another good site with reference material and software you can use to make your own image files with hidden information.

http://www.sans.org/infosecFAQ/covertchannels/steganography3. htm — Article by Richard Lewis on steganography.

http://www.sans.org/infosecFAQ/encryption/steganalysis2.htm — Great article by Jim Bartel on steganalysis.

112

An Introduction to Cryptography

Javek Ikbal, CISSP

This chapter presents some basic ideas behind cryptography. This is intended for an audience evaluators, recommenders, and end users of cryptographic algorithms and products rather than implementers. Hence, the mathematical background will be kept to a minimum. Only widely adopted algorithms are described with some mathematical detail. We also present promising technologies and algorithms that information security practitioners might encounter and may have to choose or discard.

The Basics

What Is Cryptography?

Cryptography is the art and science of securing messages so unintended audiences cannot read, understand, or alter that message.

Related Terms and Definitions

A message in its original form is called the plaintext or cleartext. The process of securing that message by hiding its contents is encryption or enciphering. An encrypted message is called ciphertext, and the process of turning the ciphertext back to cleartext is called decryption or deciphering. Cryptography is often shortened to crypto.

Practitioners of cryptography are known as cryptographers. The art and science of breaking encryptions is known as cryptanalysis, which is practiced by cryptanalysts. Cryptography and cryptanalysis are covered in the theoretical and applied branch of mathematics known as cryptology, and practiced by cryptologists.

A cipher or cryptographic algorithm is the mathematical function or formula used to convert cleartext to ciphertext and back. Typically, a pair of algorithms is used to encrypt and decrypt.

An algorithm that depends on keeping the algorithm secret to keep the ciphertext safe is known as a restricted algorithm. Security practitioners should be aware that restricted algorithms are inadequate in the current world. Unfortunately, restricted algorithms are quite popular in some settings. Exhibit 112.1 shows the schematic flow of restricted algorithms. This can be mathematically expressed as $E(M) = C$ and $D(C) = M$, where M is the cleartext message, E is the encryption function, C is the ciphertext, and D is the decryption function.

A major problem with restricted algorithms is that a changing group cannot use it; every time someone leaves, the algorithm has to change. Because of the need to keep it a secret, each group has to build its own algorithms and software to use it.

These shortcomings are overcome by using a variable known as the key or cryptovariable. The range of possible values for the key is called the keyspace. With each group using its own key, a common and well-known algorithm may be shared by any number of groups.

Exhibit 112.1 Encryption and decryption with restricted algorithms.

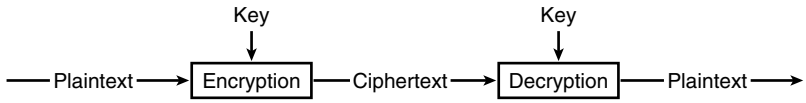


EXHIBIT 112.2 Encryption and decryption with keys.

The mathematical representation now becomes: $E_k(M) = C$ and $D_k(C) = M$, where the subscript k refers to the encryption and decryption key. Some algorithms will utilize different keys for encryption and decryption. Exhibit 112.2 illustrates that the key is an input to the algorithm.

Note that the security of all such algorithms depends on the key and not the algorithm itself. We submit to the information security practitioner that any algorithm that has not been publicly discussed, analyzed, and withstood attacks (i.e., zero restriction) should be presumed insecure and rejected.

A Brief History

Secret writing probably came right after writing was invented. The earliest known instance of cryptography occurred in ancient Egypt 4000 years ago, with the use of hieroglyphics. These were purposefully cryptic; hiding the text was probably not the main purpose — it was intended to impress. In ancient India, government spies communicated using secret codes. Greek literature has examples of cryptography going back to the time of Homer. Julius Caesar used a system of cryptography that shifted each letter three places further through the alphabet (e.g., A shifts to D, Z shifts to C, etc.). Regardless of the amount of shift, all such monoalphabetic substitution ciphers (MSCs) are also known as Caesar ciphers. While extremely easy to decipher if you know how, a Caesar cipher called ROT-13 (N = A, etc.) is still in use today as a trivial method of encryption. Why ROT-13 and not any other ROT-N? By shifting down the middle of the English alphabet, ROT-13 is self-reversing — the same code can be used to encrypt and decrypt. How this works is left as an exercise for the reader. Exhibit 112.3 shows the alphabet and corresponding Caesar cipher and ROT-13.

During the seventh century A.D., the first treatise on cryptanalysis appeared. The technique involves counting the frequency of each ciphertext letter. We know that the letter E occurs the most in English. So if we are trying to decrypt a document written in English where the letter H occurs the most, we can assume that H stands for E. Provided we have a large enough sample of the ciphertext for the frequency count to be statistically significant, this technique is powerful enough to cryptanalyze any MSC and is still in use.

Leon Battista Alberti invented a mechanical device during the 15th century that could perform a polyalphabetic substitution cipher (PSC). A PSC can be considered an improvement of the Caesar cipher because each letter is shifted by a different amount according to a predetermined rule.

The device consisted of two concentric copper disks with the alphabet around the edges. To start enciphering, a letter on the inner disk is lined up with any letter on the outer disk, which is written as the first character of the ciphertext. After a certain number of letters, the disks are rotated and the encryption continues. Because the cipher is changed often, frequency analysis becomes less effective.

The concept of rotating disks and changing ciphers within a message was a major milestone in cryptography.

The public interest in cryptography dramatically increased with the invention of the telegraph. People wanted the speed and convenience of the telegraph without disclosing the message to the operator, and cryptography provided the answer.

English Alphabet	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Caesar Cipher (3)	D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
ROT-13	N O P Q R S T U V W X Y Z A B C D E F G H I J K L M

EXHIBIT 112.3 Caesar cipher (Shift-3) and ROT-13.

After World War I, U.S. military organizations poured resources into cryptography. Because of the classified nature of this research, there were no general publications that covered cryptography until the late 1960s; and the public interest went down again.

During this time, computers were also gaining ground in nongovernment areas, especially the financial sector; and the need for a nonmilitary crypto-system was becoming apparent. The organization currently known as the National Institute of Standards and Technology (NIST), then called the National Bureau of Standards (NBS), requested proposals for a standard cryptographic algorithm. IBM responded with Lucifer, a system developed by Horst Feistel and colleagues. After adopting two modifications from the National Security Agency (NSA), this was adopted as the federal Data Encryption Standard (DES) in 1976.[1] NSA's changes caused major controversy, specifically because it suggested DES use 56-bit keys instead of 112-bit keys as originally submitted by IBM.

During the 1970s and 1980s, the NSA also attempted to regulate cryptographic publications but was unsuccessful. However, general interest in cryptography increased as a result. Academic and business interest in cryptography was high, and extensive research led to significant new algorithms and techniques.

Advances in computing power have made 56-bit keys breakable. In 1998, a custom-built machine from the Electronic Frontier Foundation costing $210,000 cracked DES in four and a half days.[2] In January 1999, a distributed network of 100,000 machines cracked DES in 22 hours and 15 minutes.

As a direct result of these DES cracking examples, NIST issued a Request for Proposals to replace DES with a new standard called the Advanced Encryption Standard (AES).[3] On November 26, 2001, NIST selected Rijndael as the AES.

The Alphabet-Soup Players: Alice, Bob, Eve, and Mike

In our discussions of cryptographic protocols, we will use an alphabet soup of names that are participating in (or are trying to break into) a secure message exchange:

- *Alice*, first participant
- *Bob*, second participant
- *Eve*, eavesdropper
- *Mike*, masquerader

Ties to Confidentiality, Integrity, and Authentication

Cryptography is not limited to confidentiality only — it can perform other useful functions.

- *Authentication.* If Alice is buying something from Bob's online store, Bob has to assure Alice that it is indeed Bob's Web site and not Mike's, the masquerader pretending to be Bob. Thus, Alice should be able to authenticate Bob's Web site, or know that a message originated from Bob.
- *Integrity.* If Bob is sending Alice, the personnel manager, a message informing her of a $5000 severance pay for Mike, Mike should not be able to intercept the message in transit and change the amount to $50,000. Cryptography enables the receiver to verify that a message has not been modified in transit.
- *Non-repudiation.* Alice places an order to sell some stocks at $10 per share. Her stockbroker, Bob, executes the order, but then the stock goes up to $18. Now Alice claims she never placed that order. Cryptography (through digital signatures) will enable Bob to prove that Alice did send that message.

Section Summary

- Any message or data in its original form is called plaintext or cleartext.
- The process of hiding or securing the plaintext is called encryption (verb: to encrypt or to encipher).
- When encryption is applied on plaintext, the result is called ciphertext.
- Retrieving the plaintext from the ciphertext is called decryption (verb: to decrypt or to decipher).
- The art and science of encryption and decryption is called cryptography, and its practitioners are cryptographers.
- The art and science of breaking encryption is called cryptanalysis, and its practitioners are cryptanalysts.

- The process and rules (mathematical or otherwise) to encrypt and decrypt are called ciphers or cryptographic algorithms.
- The history of cryptography is over 4000 years old.
- Frequency analysis is an important technique in cryptanalysis.
- Secret cryptographic algorithms should not be trusted by an information security professional.
- Only publicly available and discussed algorithms that have withstood analysis and attacks may be used in a business setting.
- Bottom line: do not use a cryptographic algorithm developed in-house (unless you have internationally renowned experts in that field).

Symmetric Cryptographic Algorithms

Algorithms or ciphers that use the same key to encrypt and decrypt are called symmetric cryptographic algorithms. There are two basic types: stream and block.

Stream Ciphers

This type of cipher takes messages in a stream and operates on individual data elements (characters, bits, or bytes).

Typically, a random-number generator is used to produce a sequence of characters called a key stream. The key stream is then combined with the plaintext via exclusive-OR (XOR) to produce the ciphertext. Exhibit 112.4 illustrates this operation of encrypting the letter Z, the ASCII value of which is represented in binary as 01011010. Note that in an XOR operation involving binary digits, only XORing 0 and 1 yields 1; all other XORs result in 0. Exhibit 112.4 shows how a stream cipher operates.

Before describing the actual workings of a stream cipher, we will examine how shift registers work because they have been the mainstay of electronic cryptography for a long time.

A linear feedback shift register (LFSR) is very simple in principle. For readers not versed in electronics, we present a layman's representation. Imagine a tube that can hold four bits with a window at the right end. Because the tube holds four bits, we will call it a four-bit shift register. We shift all bits in the tube and, as a result, the bit showing through the window changes. Here, shifting involves pushing from the left so the right-most bit falls off; and to keep the number of bits in the tube constant, we place the output of some addition operation as the new left-most bit. In the following example, we will continue with our four-bit LFSR, and the new left-most bit will be the result of adding bits three and four (the feedback) and keeping the right-most bit (note that in binary mathematics, 1+ 1 = 10, with 0 being the right-most bit, and 1+ 0 = 1). For every shift that occurs, we look through the window and note the right-most bit. As a result, we will see the sequence shown in Exhibit 112.5.

Note that after $2^{(N=4)} - 1 = 15$ iterations, we will get a repetition. This is the maximum number of unique sequences (also called period) when dealing with a four-bit LFSR (because we have to exclude 0000, which will always produce a sequence of 0000s). Choosing a different feedback function may have reduced the period, and the longest unique sequence is called the maximal length. The maximal length is important because

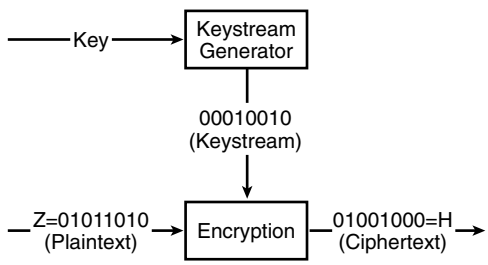


EXHIBIT 112.4 Stream cipher operation.

1111.-> 0111 -> 0011 -> 0001 -> 1000 -> 0100 -> 0010 -> 1001 -> 1100 -> 0110 -> 1011 -> 0101 -> 1010 -> 1101 -> 1110 -> 1111
Keystream: 111100010011010 (Right-most bit through the window before repetition).

EXHIBIT 112.5 4-bit LFSR output.

repeating key streams mean the same plaintext will produce the same ciphertext, and this will be vulnerable to frequency analysis and other attacks.

To construct a simple stream cipher, take an LFSR (or take many different sizes and different feedback functions). To encrypt each bit of the plaintext, take a bit from the plaintext, XOR it with a bit from the key stream to generate the ciphertext (refer to Exhibit 112.4), and so on.

Of course, other stream ciphers are more complex and involve multiple LFSRs and other techniques.[4] We will discuss RC4 as an example of a stream cipher. First, we will define the term S-box.

An S-box is also known as a substitution box or table and, as the name implies, it is a table or system that provides a substitution scheme. Shift registers are S-boxes; they provide a substitution mechanism.

RC4 uses an output feedback mechanism combined with 256 S-boxes (numbered $S_0 \ldots S_{255}$) and two counters, i and j.

A random byte K is generated through the following steps:

```
i = (i +1) mod 256
j = (j + Si) mod 256
swap (Si, Sj)
t = (Si + Sj) mod 256
K = St
```

Now, K XOR Plaintext = Ciphertext, and K XOR Ciphertext = Plaintext

Block Ciphers

A block cipher requires the accumulation of some amount of data or multiple data elements before ciphering can begin. Encryption and decryption happen on chunks of data, unlike stream ciphers, which operate on each character or bit independently.

DES

The Data Encryption Standard (DES) is over 25 years old; because of its widespread implementation and use, it will probably coexist with the new Advanced Encryption Standard (AES) for a few years.

Despite initial concern about NSA's role in crafting the standard, DES generated huge interest in cryptography; vendors and users alike were eager to adopt the first government-approved encryption standard that was released for public use.

The DES calls for reevaluations of DES every five years. Starting in 1987, the NSA warned that it would not recertify DES because it was likely that it soon would be broken; they proposed secret algorithms available on tamper-proof chips only. Users of DES, including major financial institutions, protested; DES got a new lease on life until 1992. Because no new standards became available in 1992, it lived on to 1998 and then until the end of 2001, when AES became the standard.

DES is a symmetric block cipher that operates in blocks of 64 bits of data at a time, with 64-bit plaintext resulting in 64-bit ciphertext. If the data is not a multiple of 64 bits, then it is padded at the end. The effective key-length is 56 bits with 8 bits of parity. All security rests with the key.

A simple description of DES is as follows:[1]

Take the 64-bit block of message (M).
Rearrange the bits of M (initial permutation, IP).
Break IP down the middle into two 32-bit blocks (L & R).
Shift the key bits, and take a 48-bit portion from the key.
Save the value of R into R_{old}.
Expand R via a permutation to 48 bits.

XOR R with the 48-bit key and transform via eight S-boxes into a new 32-bit chunk.
Now, R takes on the value of the new R XOR-ed with L.
And L takes on the value of R_{old}.
Repeat this process 15 more times (total 16 rounds).
Join L and R.
Reverse the permutation IP (final permutation, FP).

There are some implementations without IP and FP; because they do not match the published standard, they should not be called DES or DES-compliant, although they offer the same degree of security.

Certain DES keys are considered weak, semiweak, or possibly weak: a key is considered weak if it consists of all 1s or all 0s, or if half the keys are 1s and the other half are 0s.[5]

Conspiracy theories involving NSA backdoors and EFFs DES-cracking machine notwithstanding, DES lives on in its original form or a multiple-iteration form popularly known as Triple-DES.

Triple-DES is DES done thrice, typically with two 56-bit keys. In the most popular form, the first key is used to DES-encrypt the message. The second key is used to DES-decrypt the encrypted message. Because this is not the right key, the attempted decryption only scrambles the data even more. The resultant ciphertext is then encrypted again with the first key to yield the final ciphertext. This three-step procedure is called Triple-DES. Sometimes, three keys are used.

Because this follows an Encryption > Decryption > Encryption scheme, it is often known as DES-EDE.

ANSI standard X9.52 describes Triple-DES encryption with keys k_1, k_2, k_3 as:

$$C = E_{k3}(D_{k2}(E_{k1}(M)))$$

where E_k and D_k denote DES encryption and DES decryption, respectively, with the key k. Another variant is DES-EEE, which consists of three consecutive encryptions. There are three keying options defined in ANSI X9.52 for DES-EDE:

The three keys k_1, k_2, and k_3 are different (three keys).
k_1 and k_2 are different, but $k_1 = k_3$ (two keys).
$k_1 = k_2 = k_3$ (one key).

The third option makes Triple-DES backward-compatible with DES and offers no additional security.

AES (Rijndael)

In 1997, NIST issued a Request for Proposals to select a symmetric-key encryption algorithm to be used to protect sensitive (unclassified) federal information. This was to become the Advanced Encryption Standard (AES), the DES replacement. In 1998, NIST announced the acceptance of 15 candidate algorithms and requested the assistance of the cryptographic research community in analyzing the candidates. This analysis included an initial examination of the security and efficiency characteristics for each algorithm.

NIST reviewed the results of this preliminary research and selected MARS, RC6™, Rijndael, Serpent, and Twofish as finalists. After additional review, in October 2000, NIST proposed Rijndael as AES. For research results and rationale for selection, see Reference 5.

Before discussing AES, we will quote the most important answer from the Rijndael FAQ:

> If you're Dutch, Flemish, Indonesian, Surinamer or South African, it's pronounced like you think it should be. Otherwise, you could pronounce it like reign dahl, rain doll, or rhine dahl. We're not picky. As long as you make it sound different from region deal.[6]

Rijndael is a block cipher that can process blocks of 128-, 192-, and 256-bit length using keys 128-, 192-, and 256-bits long. All nine combinations of block and key lengths are possible.[7] The AES standard specifies only 128-bit data blocks and 128-, 192-, and 256-bit key lengths. Our discussions will be confined to AES and not the full scope of Rijndael. Based on the key length, AES may be referred to as AES-128, AES-192, or AES-256. We will present a simple description of Rijndael. For a mathematical treatment, see References 8 and 9.

Rijndael involves an initial XOR of the state and a round key, nine rounds of transformations (or rounds), and a round performed at the end with one step omitted. The input to each round is called the state. Each round consists of four transformations: SubBytes, ShiftRow, MixColumn (omitted from the tenth round), and AddRoundKey.

In the SubBytes transformation, each of the state bytes is independently transformed using a nonlinear S-box.

In the ShiftRow transformation, the state is processed by cyclically shifting the last three rows of the state by different offsets.

In the MixColumn transformation, data from all of the columns of the state are mixed (independently of one another) to produce new columns.

In the AddRoundKey step in the cipher and inverse cipher transformations, a round key is added to the state using an XOR operation. The length of a round key equals the size of the state.

Weaknesses and Attacks

A well-known and frequently used encryption is the stream cipher available with PKZIP. Unfortunately, there is also a well-known attack involving known plaintext against this — if you know part of the plaintext, it is possible to decipher the file.[10] For any serious work, information security professionals should not use PKZIP's encryption.

In 1975, it was theorized that a customized DES cracker would cost $20 million. In 1998, EFF built one for $220,000.[2] With the advances in computing power, the time and money required to crack DES has significantly gone down even more. Although it is still being used, if possible, use AES or Triple-DES.

Section Summary

- Symmetric cryptographic algorithms or ciphers are those that use the same key to encrypt and decrypt.
- Stream ciphers operate one bit at a time.
- Stream ciphers use a key stream generator to continuously produce a key stream that is used to encrypt the message.
- A repeating key stream weakens the encryption and makes it vulnerable to cryptanalysis.
- Shift registers are often used in stream ciphers.
- Block ciphers operate on a block of data at a time.
- DES is the most popular block cipher.
- DES keys are sometimes referred to as 64-bit, but the effective length is 56 bits with 8 parity bits; hence, the actual key length is 56 bits.
- There are known weak DES keys; ensure that those are not used.
- DES itself has been broken and it should be assumed that it is not secure against attack.
- Make plans to migrate away from DES; use Triple-DES or Rijndael instead of DES, if possible.
- Do not use the encryption offered by PKZIP for nontrivial work.

Asymmetric (Public Key) Cryptography

Asymmetric is the term applied in a cryptographic system where one key is used to encrypt and another is used to decrypt.

Background

This concept was invented in 1976 by Whitfield Diffie and Martin Hellman[11] and independently by Ralph Merkle. The basic theory is quite simple: is there a pair of keys so that if one is used to encrypt, the other can be used to decrypt — and given one key, finding the other would be extremely hard?

Luckily for us, the answer is yes, and this is the basis of asymmetric (often called public key) cryptography.

There are many algorithms available, but most of them are either insecure or produce ciphertext that is larger than the plaintext. Of the algorithms that are both secure and efficient, only three can be used for both encryption and digital signatures.[4] Unfortunately, these algorithms are often slower by a factor of 1000 compared to symmetric key encryption.

As a result, hybrid cryptographic systems are popular: Suppose Alice and Bob want to exchange a large message. Alice generates a random session key, encrypts it using asymmetric encryption, and sends it over to Bob, who has the other half of the asymmetric key to decode the session key. Because the session key is small, the overhead to asymmetrically encipher/decipher it is not too large. Now Alice encrypts the message with the

session key and sends it over to Bob. Bob already has the session key and deciphers the message with it. As the large message is enciphered/deciphered using much faster symmetric encryption, the performance is acceptable.

RSA

We will present a discussion of the most popular of the asymmetric algorithms — RSA, named after its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman. Readers are directed to Reference 12 for an extensive treatment. RSA's patent expired in September 2000; and RSA has put the algorithm in the public domain, enabling anyone to implement it at zero cost.

First, a mathematics refresher:

- If an integer P cannot be divided (without remainders) by any number other than itself and 1, then P is called a prime number. Other prime numbers are 2, 3, 5, and 7.
- Two integers are relatively prime if there is no integer greater than one that divides them both (their greatest common divisor is 1). For example, 15 and 16 are relatively prime, but 12 and 14 are not.
- The mod is defined as the remainder. For example, 5 mod 3 = 2 means divide 5 by 3 and the result is the remainder, 2.

Note that RSA depends on the difficulty of factoring large prime numbers. If there is a sudden leap in computer technology or mathematics that changes that, security of such encryption schemes will be broken. Quantum and DNA computing are two fields to watch in this arena.

Here is a step-by-step description of RSA:

1. Find P and Q, two large (e.g., 1024-bit or larger) prime numbers. For our example, we will use P = 11 and Q = 19, which are adequate for this example (and more manageable).
2. Calculate the product PQ, and also the product (P – 1)(Q – 1). So PQ = 209, and (P – 1)(Q – 1) = 180.
3. Choose an odd integer E such that E is less than PQ, and such that E and (P – 1)(Q – 1) are relatively prime. We will pick E = 7.
4. Find the integer D so that (DE – 1) is evenly divisible by (P – 1)(Q – 1). D is called the multiplicative inverse of E. This is easy to do: let us assume that the result of evenly dividing (DE – 1) by (P – 1)(Q – 1) is X, where X is also an integer. So we have X = (DE – 1)/(P – 1)(Q – 1); and solving for D, we get D = (X(P – 1)(Q – 1) + 1)/E. Start with X = 1 and keep increasing its value until D is an integer. For our example, D works out to be 103.
5. The public key is (E and PQ), the private key is D. Destroy P and Q (note that given P and Q, it would be easy to work out E and D; but given only PQ and E, it would be hard to determine D). Give out your public key (E, PQ) and keep D secure and private.
6. To encrypt a message M, we raise M to the *E*th power, divide it by PQ, and the remainder (the mod) is the ciphertext. Note that M must be less than PQ. A mathematical representation will be ciphertext = M^E mod PQ. So if we are encrypting 13 (M = 13), our ciphertext = 13^7 mod 209 = 29.
7. To decrypt, we take the ciphertext, raise it to the *D*th power, and take the mod with PQ. So plaintext = 29^{103} mod 209 = 13.

Compared to DES, RSA is about 100 times slower in software and 1000 times slower in hardware. Because AES is even faster than DES in software, the performance gap will widen in software-only applications.

Elliptic Curve Cryptosystems (ECC)

As we saw, solving RSA depends on a hard math problem: factoring very large numbers. There is another hard math problem: reversing exponentiation (logarithms). For example, it is possible to easily raise 7 to the 4th power and get 2401; but given only 2401, reversing the process and obtaining 7^4 is more difficult (at least as hard as performing large factorizations).

The difficulty in performing discrete logarithms over elliptic curves (not to be confused with an ellipse) is even greater;[13] and for the same key size, it presents a more difficult challenge than RSA (or presents the same difficulty/security with a smaller key size). There is an implementation of ECC that uses the factorization problem, but it offers no practical advantage over RSA.

An elliptic curve has an interesting property: it is possible to define a point on the curve as the sum of two other points on the curve. Following is a high-level discussion of ECC. For details, see Reference 13.

Example: Alice and Bob agree on a nonsecret elliptic curve and a nonsecret fixed curve point F. Alice picks a secret random integer A_k as her secret key and publishes the point $A_P = A_k * F$ as her public key. Bob picks a secret random integer B_k as his secret key and publishes the point $B_P = B_k * F$ as his public key. If Alice wants to send a message to Bob, she can compute $A_k * B_P$ and use the result as the secret key for a symmetric block cipher like AES. To decrypt, Bob can compute the same key by finding $B_k * A_P$, because $B_k * A_P = B_k * (A_k * F) = A_k * (B_k * F) = A_k * B_P$.

ECC has not been subject to the extensive analysis that RSA has and is comparatively new.

Attacks

It is possible to attack RSA by factoring large numbers, or guessing all possible values of $(P - 1)(Q - 1)$ or D. These are computationally infeasible, and users should not worry about them. But there are chosen ciphertext attacks against RSA that involve duping a person to sign a message (provided by the attacker). This can be prevented by signing a hash of the message, or by making minor cosmetic changes to the document by signing it. For a description of attacks against RSA, see Reference 14. Hash functions are described later in this chapter.

Real-World Applications

Cryptography is often a business enabler. Financial institutions encrypt the connection between the user's browser and Web pages that show confidential information such as account balances. Online merchants similarly encrypt the link so customer credit card data cannot be sniffed in transit. Some even use this as a selling point: "Our Web site is protected with the highest encryption available." What they are really saying is that this Web site uses 128-bit Secure Sockets Layer (SSL).

As an aside, there are no known instances of theft of credit card data in transit; but many high-profile stories of customer information theft, including theft of credit card information, are available. The theft was possible because enough safeguards were not in place, and the data was usable because it was in cleartext, that is, not encrypted. Data worth protecting should be protected in all stages, not just in transit.

SSL and TLS

Normal Web traffic is cleartext — your ISP can intercept it easily. SSL provides encryption between the browser and a Web server to provide security and identification. SSL was invented by Netscape[15] and submitted to the Internet Engineering Task Force (IETF). In 1996, IETF began with SSL v3.0 and, in 1999, published TLS v1.0 as a proposed standard.[16] TLS is a term not commonly used, but we will use TLS and SSL interchangeably.

Suppose Alice, running a popular browser, wants to buy a book from Bob's online book store at bobs-books.com, and is worried about entering her credit card information online. (For the record, SSL/TLS can encrypt connections between any two network applications and not Web browsers and servers only.) Bob is aware of this reluctance and wants to allay Alice's fears — he wants to encrypt the connection between Alice's browser and bobsbooks.com. The first thing he has to do is install a digital certificate on his Web server.

A certificate contains information about the owner of the certificate: e-mail address, owner's name, certificate usage, duration of validity, and resource location or distinguished name (DN), which includes the common name (CN, Web site address or e-mail address, depending on the usage), and the certificate ID of the person who certifies (signs) this information. It also contains the public key, and finally a hash to ensure that the certificate has not been tampered with.

Anyone can create a digital certificate with freely available software, but just like a person cannot issue his own passport and expect it to be accepted at a border, browsers will not recognize self-issued certificates. Digital certificate vendors have spent millions to preinstall their certificates into browsers, so Bob has to buy a certificate from a well-known certificate vendor, also known as root certificate authority (CA). There are certificates available with 40- and 128-bit encryptions. Because it usually costs the same amount, Bob should buy a 128-bit certificate and install it on his Web server. As of this writing, there are only two vendors with wide acceptance of certificates: VeriSign and Thawte. Interestingly, VeriSign owns Thawte, but Thawte certificate prices are significantly lower.

So now Alice comes back to the site and is directed toward a URL that begins with https instead of http. That is the browser telling the server that an SSL session should be initiated. In this negotiation phase, the browser also tells the server what encryption schemes it can support. The server will pick the strongest of the supported ciphers and reply back with its own public key and certificate information. The browser will check

if it has been issued by a root CA. If not, it will display a warning to Alice and ask if she still wants to proceed. If the server name does not match the name contained in the certificate, it will also issue a warning.

If the certificate is legitimate, the browser will:

- Generate a random symmetric encryption key
- Encrypt this symmetric key with the server's public key
- Encrypt the URL it wants with the symmetric key
- Send the encrypted key and encrypted URL to the server

The server will:

- Decrypt the symmetric key with its private key
- Decrypt the URL with the symmetric key
- Process the URL
- Encrypt the reply with the symmetric key
- Send the encrypted reply back to the browser

In this case, although encryption is two-way, authentication is one-way only: the server's identity is proven to the client but not vice versa. Mutual authentication is also possible and performed in some cases. In a high-security scenario, a bank could issue certificates to individuals, and no browser would be allowed to connect without those individual certificates identifying the users to the bank's server.

What happens when a browser capable of only 40-bit encryption (older U.S. laws prohibited export of 128-bit browsers) hits a site capable of 128 bits? Typically, the site will step down to 40-bit encryption. But CAs also sell super or step-up certificates that, when encountered with a 40-bit browser, will temporarily enable 128-bit encryption in those browsers. Step-up certificates cost more than regular certificates.

Note that the root certificates embedded in browsers sometimes expire; the last big one was VeriSign's in 1999. At that time, primarily financial institutions urged their users to upgrade their browsers. Finally, there is another protocol called Secure HTTP that provides similar functionality but is very rarely used.

Choosing an Algorithm

What encryption algorithm, with what key size, would an information security professional choose? The correct answer is: it depends; what is being encrypted, who do we need to protect against, and for how long?

If it is stock market data, any encryption scheme that will hold up for 20 minutes is enough; in 20 minutes, the same information will be on a number of free quote services. Your password to the *New York Times* Web site? Assuming you do not use the same password for your e-mail account, SSL is overkill for that server. Credit card transactions, bank accounts, and medical records need the highest possible encryption, both in transit and in storage.

Export and International Use Issues

Until recently, exporting 128-bit Web browsers from the United States was a crime, according to U.S. law. Exporting software or hardware capable of strong encryption is still a crime. Some countries have outlawed the use of encryption, and some other countries require a key escrow if you want to use encryption. Some countries have outlawed use of all but certain approved secret encryption algorithms. We strongly recommend that information security professionals become familiar with the cryptography laws of the land, especially if working in an international setting.[17]

Section Summary

- In asymmetric cryptography, one key is used to encrypt and another is used to decrypt.
- Asymmetric cryptography is often also known as public key cryptography.
- Asymmetric cryptography is up to 1000 times slower than symmetric cryptography.
- RSA is the most popular and well-understood asymmetric cryptographic algorithm.
- RSA's security depends on the difficulty of factoring very large (>1024-bit) numbers.
- Elliptic curve cryptography depends on the difficulty of finding discrete logarithms over elliptic curves.

- Smaller elliptic curve keys offer similar security as comparatively larger RSA keys.
- It is possible to attack RSA through chosen plaintext attacks.
- SSL is commonly used to encrypt information between a browser and a Web server.
- Choosing a cipher and key length depends on what needs to be encrypted, for how long, and against whom.
- There are significant legal implications of using encryption in a multinational setting.

Key Management and Exchange

In symmetric encryption, what happens when one person who knows the keys goes to another company (or to a competitor)? Even with public key algorithms, keeping the private key secret is paramount: without it, all is lost. For attackers, the reverse is true; it is often easier to attack the key storage instead of trying to crack the algorithm. A person who knows the keys can be bribed or kidnapped and tortured to give up the keys, at which time the encryption becomes worthless. Key management describes the problems and solutions to securely generating, exchanging, installing and storing, verifying, and destroying keys.

Generation

Encryption software typically generates its own keys (it is possible to generate keys in one program and use them in another); but because of the implementation, this can introduce weaknesses. For example, DES software that picks a known weak or semiweak key will create a major security issue. It is important to use the largest possible keyspace: a 56-bit DES key can be picked from the 256 ASCII character set, the first 128 of ASCII, or the 26 letters of the alphabet. Guessing the 56-bit DES key (an exhaustive search) involves trying out all 56-bit combinations from the keyspace. Common sense tells us that the exhaustive search of 256 bytes will take much longer than that for 26 bytes. With a large keyspace, the keys must be random enough so as to be not guessable.

Exchange

Alice and Bob are sitting on two separate islands. Alice has a bottle of fine wine, a lock, its key, and an empty chest. Bob has another lock and its key. An islander is willing to transfer items between the islands but will keep anything that he thinks is not secured, so you cannot send a key, an unlocked lock, or a bottle of wine on its own.

How does Alice send the wine to Bob? See the answer at the end of this section.

This is actually a key exchange problem in disguise: how does Alice get a key to Bob without its being compromised by the messenger? For asymmetric encryption, it is easy — the public key can be given out to the whole world. For symmetric encryption, a public key algorithm (like SSL) can be used; or the key may be broken up and each part sent over different channels and combined at the destination.

Answer to our key/wine exchange problem: Alice puts the bottle into the chest and locks it with her lock, keeps her key, and sends the chest to the other island. Bob locks the chest with his lock, and sends it back to Alice. Alice takes her lock off the chest and sends it back to Bob. Bob unlocks the chest with his key and enjoys the wine.

Installation and Storage

How a key is installed and stored is important. If the application does no initial validation before installing a key, an attacker might be able to insert a bad key into the application. After the key is installed, can it be retrieved without any access control? If so, anyone with access to the computer would be able to steal that key.

Change Control

How often a key is changed determines its efficiency. If a key is used for a long time, an attacker might have sufficient samples of ciphertext to be able to cryptanalyze the information. At the same time, each change brings up the exchange problem.

Destruction

A key no longer in use has to be disposed of securely and permanently. In the wrong hands, recorded ciphertext may be decrypted and give an enemy insights into current ciphertext.

Examples and Implementations

PKI

A public key infrastructure (PKI) is the set of systems and software required to use, manage, and control public key cryptography. It has three primary purposes: publish public keys, certify that a public key is tied to an individual or entity, and provide verification as to the continued validity of a public key. As discussed before, a digital certificate is a public key with identifying information for its owner. The certificate authority (CA) "signs" the certificate and verifies that the information provided is correct. Now all entities that trust the CA can trust that the identity provided by a certificate is correct. The CA can revoke the certificate and put it in the certificate revocation list (CRL), at which time it will not be trusted anymore. An extensive set of PKI standards and documentation is available.[18] Large companies run their own CA for intranet/extranet use. In Canada and Hong Kong, large public CAs are operational. But despite the promises of the "year of the PKI," market acceptance and implementation of PKIs are still in the future.

Kerberos

From the comp.protocol.kerberos FAQ:

> Kerberos; also spelled Cerberus. *n.* The watchdog of Hades, whose duty it was to guard the entrance — against whom or what does not clearly appear; it is known to have had three heads.
>
> **— Ambrose Bierce**
>
> *The Enlarged Devil's Dictionary*

Kerberos was developed at MIT in the 1980s and publicly released in 1989. The primary purposes were to prevent cleartext passwords from traversing the network and to ease the log-in process to multiple machines.[19] The current version is 5 — there are known security issues with version 4. The three heads of Kerberos comprise the key distribution center (KDC), the client, and the server that the client wants to access. Kerberos 5 is built into Windows 2000 and later, and will probably result in wider adoption of Kerberos (notwithstanding some compatibility issues of the Microsoft implementation of the protocol[20]).

The KDC runs two services: authentication service (AS) and ticket granting service (TGS). A typical Kerberos session (shown in Exhibit 112.6) proceeds as follows when Alice wants to log on to her e-mail and retrieve it.

1. She will request a ticket granting ticket (TGT) from the KDC, where she already has an account. The KDC has a hash of her password, and she will not have to provide it. (The KDC must be extremely secure to protect all these passwords.)
2. The TGS on the KDC will send Alice a TGT encrypted with her password hash. Without knowing the password, she cannot decrypt the TGT.
3. Alice decrypts the TGT; then, using the TGT, she sends another request to the KDC for a service ticket to access her e-mail server. The service ticket will not be issued without the TGT and will only work for the e-mail server.
4. The KDC grants Alice the service ticket.
5. Alice can access the e-mail server.

Note that both the TGT and the ST have expiration times (default is ten hours); so even if one or both tickets are captured, the exposure is only until the ticket expiration time. All computer system clocks participating in a Kerberos system must be within five minutes of each other and all services that grant access. Finally, the e-mail server must be kerberized (support Kerberos).

Section Summary

- Key management (generating/exchanging/storing/installing/destroying keys) can compromise security.
- Public key cryptography is often the best solution to key distribution issues.

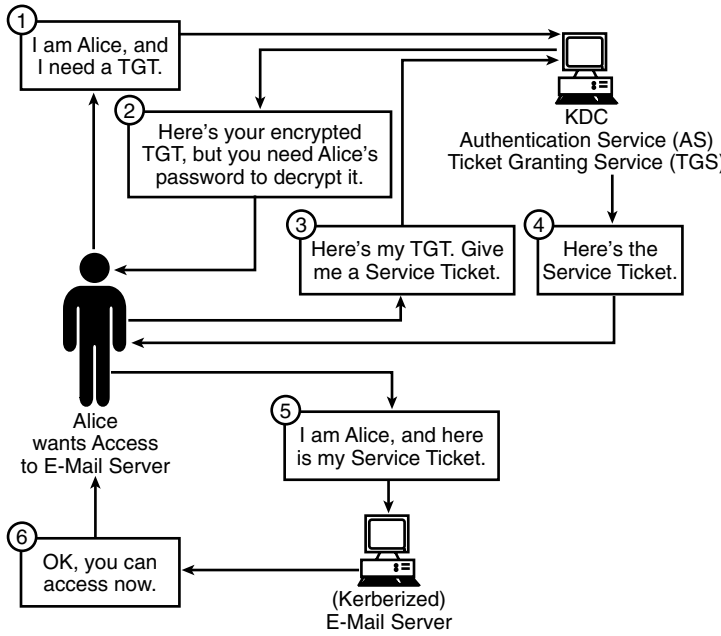


EXHIBIT 112.6 Kerberos in operation.

- A public key infrastructure (PKI) is a system that can manage public keys.
- A certificate authority (CA) is a PKI that can validate public keys.
- Digital certificates are essentially public keys that also include key owner information. The key and information are verified by a CA.
- If an entity trusts a CA, it can also trust digital certificates that the CA signs (authenticates).
- Kerberos is a protocol for eliminating cleartext passwords across networks.
- A ticket granting ticket (TGT) is issued to the user, who will use that to request a service ticket. All tickets expire after a certain time.
- Under Kerberos, tickets are encrypted and cleartext passwords never cross the network.

Hash Functions

A hash function is defined as a process that can take an arbitrary-length message and return a fixed-length value from that message. For practical use, we require further qualities:

- Given a message, it should be easy to find the hash.
- Given the hash, it should be hard to find the message.
- Given the message, it should be hard to find another (specific or random) message that produces the same hash.

Message Digests

A message digest is the product of a one-way hash function applied on a message: it is a fingerprint or a unique summary that can uniquely identify the message.

MD2, MD4, and MD5

Ron Rivest (the R in RSA) designed all of these. All three produce 128-bit hashes. MD4 has been successfully attacked. MD5 has been found weak in certain cases; it is possible to find another random message that will produce the same hash. MD2 is slower, although no known weaknesses exist.

SHA

The secure hash algorithm (SHA) was designed by NIST and NSA, and is used in the digital signature standard, officially known as the Secure Hash Standard (SHS) and is available as FIPS-180-1.[21]

The current SHA produces a 160-bit hash and is also known as SHA-1. There are additional standards undergoing public comments and reviews that will offer 256-, 384-, and 512-bit hashes. The draft standard is available.[16] The proposed standards will offer security matching the level of AES. The draft is available as FIPS-180-2.[22]

Applications of Message Digests

Message digests are useful and should be used to provide message integrity. Suppose Alice wants to pay $2000 to Eve, a contract network administrator. She types an e-mail to Bob, her accountant, to that effect. Before sending the message, Alice computes the message digest (SHA-1 or MD5) of the message and then sends the message followed by the message digest. Eve intercepts the e-mail and changes $2000 to $20,000; but when Bob computes the message digest of the e-mail, it does not match the one from Alice, and he knows that the e-mail has been tampered with.

But how do we ensure that the e-mail to Bob indeed came from Alice, when faking an e-mail source address is notoriously easy? This is where digital signatures come in.

Digital Signatures

Digital signatures were designed to provide the same features of a conventional ("wet") signature. The signature must be non-repudiatable, and it must be nontransferable (cannot be lifted and reused on another document). It must also be irrevocably tied back to the person who owns it.

It is possible to use symmetric encryption to digitally sign documents using an intermediary who shares keys with both parties, but both parties do not have a common key. This is cumbersome and not practical.

Using public key cryptography solves this problem neatly. Alice will encrypt a document with her private key, and Bob will decrypt it with Alice's public key. Because it could have been encrypted with only Alice's private key, Bob can be sure it came from Alice. But there are two issues to watch out for: (1) the rest of the world may also have Alice's public key, so there will be no privacy in the message; and (2) Bob will need a trusted third party (a certificate authority) to vouch for Alice's public key.

In practice, signing a long document may be computationally costly. Typically, first a one-way hash of the document is generated, the hash is signed, and then both the signed hash and the original document are sent. The recipient also creates a hash and compares the decrypted signed hash to the generated one. If both match, then the signature is valid.

Digital Signature Algorithm (DSA)

NIST proposed DSA in 1991 to be used in the Digital Signature Standard and the standard issued in May 1994. In January 2000, it announced the latest version as FIPS PUB 186-2.[23] As the name implies, this is purely a signature standard and cannot be used for encryption or key distribution.

The operation is pretty simple. Alice creates a message digest using SHA-1, uses her private key to sign it, and sends the message and the digest to Bob. Bob also uses SHA-1 to generate the message digest from the message and uses Alice's public key on the received message digest to decrypt it. Then the two message digests are compared. If they match, the signature is valid.

Finally, digital signatures should not be confused with the horribly weakened "electronic signature" law passed in the United States, where a touch-tone phone press could be considered an electronic signature and enjoy legal standing equivalent to an ink signature.

Message Authentication Codes (MACs)

MACs are one-way hash functions that include the key. People with the identical key will be able to verify the hash. MACs provide authentication of files between users and may also provide file integrity to a single user to ensure files have not been altered in a Web site defacement. On a Web server, the MAC of all files could be computed and stored in a table. With only a one-way hash, new values could have been inserted in the table

and the user will not notice. But in a MAC, because the attacker will not know the key, the table values will not match; and an automated process could alert the owner (or automatically replace files from backup).

A one-way hash function can be turned into a MAC by encrypting the hash using a symmetric algorithm and keeping the key secret. A MAC can be turned into a one-way hash function by disclosing the key.

Section Summary

- Hash functions can create a fixed-length digest of arbitrary-length messages.
- One-way hashes are useful: given a hash, finding the message should be very hard.
- Two messages should not generate the same hash.
- MD2, MD4, and MD5 all produce 128-bit hashes.
- SHA-1 produces a 160-bit hash.
- Encrypting a message digest with a private key produces a digital signature.
- Message authentication codes are one-way hashes with the key included.

Other Cryptographic Notes

Steganography

Steganography is a Greek word that means sheltered writing. This is a method that attempts to hide the existence of a message or communication. In February 2001, *USA Today* and various other news organizations reported that terrorists are using steganography to hide their communication in images on the Internet.[24] A University of Michigan study[25] examined this by analyzing two million images downloaded from the Internet and failed to find a single instance.

In its basic form, steganography is simple. For example, every third letter of a memo could hide a message. And it has the added advantage over encryption that it does not arouse suspicion: often, the presence of encryption could set off an investigation; but a message hidden in plain sight would be ignored.

The medium that hides the message is called the cover medium, and it must have parts that can be altered or used without damaging or noticeably changing the cover media. In case of digital cover media, these alterable parts are called redundant bits. These redundant bits or a subset can be replaced with the message we want to hide.

Interestingly, steganography in digital media is very similar to digital watermarking, where a song or an image can be uniquely identified to prevent theft or unauthorized use.

Digital Notary Public

Digital notary service is a logical extension of digital signatures. Without this service, Alice could send a digitally signed offer to Bob to buy a property; but after property values drop the next day, she could claim she lost her private key and call the message a forgery. Digital notaries could be trusted third parties that will also time-stamp Alice's signature and give Bob legal recourse if Alice tries to back out of the deal. There are commercial providers of this type of service.

With time-sensitive offers, this becomes even more important. Time forgery is a difficult if not impossible task with paper documents, and it is easy for an expert to detect. With electronic documents, time forgeries are easy and detection is almost impossible (a system administrator can change the time stamp of an e-mail on the server). One do-it-yourself time-stamping method suggests publishing the one-way hash of the message in a newspaper (as a commercial notice or advertisement). From then on, the date of the message will be time-stamped and available for everyone to verify.

Backdoors and Digital Snake Oil

We will reiterate our warnings about not using in-house cryptographic algorithms or a brand-new encryption technology that has not been publicly reviewed and analyzed. It may promise speed and security or low cost, but remember that only algorithms that withstood documented attacks are worthy of serious use — others should be treated as unproven technology, not ready for prime time.

Also, be careful before using specific software that a government recommends. For example, Russia mandates use of certain approved software for strong encryption. It has been mentioned that the government certifies all such software after behind-the-scenes key escrow. To operate in Russia, a business may not have any choice in this matter, but knowing that the government could compromise the encryption may allow the business to adopt other safeguards.

References

1. Data Encryption Standard (DES): http://www.itl.nist.gov/fipspubs/fip46–2.htm.
2. Specialized DES cracking computer: http://www.eff.org/descracker.html.
3. Advanced Encryption Standard (AES): http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.
4. Bruce Schneier, *Applied Cryptography*, 2nd edition,
5. Weak DES keys: http://www.ietf.org/rfc/rfc2409.txt, Appendix A.
6. AES selection report: http://csrc.nist.gov/encryption/aes/round2/r2report.pdf.
7. Rijndael developer's site: http://www.esat.kuleuven.ac.be/~rijmen/rijndael/.
8. Rijndael technical overview: http://www.baltimore.com/devzone/aes/tech_overview. html.
9. Rijndael technical overview: http://www.sans.org/infosecFAQ/encryption/mathematics. htm.
10. PKZIP encryption weakness: http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get. cgi/1994/CS/CS0 842.ps.gz.
11. Diffie and Hellman paper on Public Key Crypto: http://cne.g.,mu.edu/modules/acmpkp/ security/texts/ NEWDIRS.PDF.
12. RSA algorithm: http://www.rsasecurity.com/rsalabs/rsa_algorithm/index.html.
13. Paper on elliptic curve cryptography: ftp://ftp.rsasecurity.com/pub/ctryptobytes/ crypto1n2. pdf.
14. Attacks on RSA: http://crypto.stanford.edu/~dabo/abstracts/RSAattack-survey.html.
15. SSL 3.0 protocol: http://www.netscape.com/eng/ssl3/draft302.txt.
16. TLS 1.0 protocol: http://www.ietf.org/rfc/rfc2246.txt.
17. International encryption regulations: http://cwis.kub.nl/~frw/people/koops/lawsurvy. htm.
18. IETF PKI working group documents: http://www.ietf.org/html.charters/pkix-charter.html.
19. Kerberos documentation collection: http://web.mit.edu/kerberos/www/.
20. Kerberos issues in Windows 2000: http://www.nrl.navy.mil/CCS/people/kenh/ kerberos-faq.html#ntbroken.
21. Secure Hash Standard (SHS): http://www.itl.nist.gov/fipspubs/fip180-1.htm.
22. Improved SHS draft: http://csrc.nist.gov/encryption/shs/dfips-180-2.pdf.
23. Digital Signature Standard (DSS): http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1. pdf.
24. *USA Today* story on steganography: http://www.usatoday.com/life/cyber/tech/2001-02-05-binladen. htm#more.
25. Steganography study: http://www.citi.umich.edu/techreports/reports/citi-tr-01-11.pdf.

113

Hash Algorithms: From Message Digests to Signatures

Keith Pasley, CISSP

There are many information-sharing applications that are in use on modern networks today. Concurrently, there are a growing number of users sharing data of increasing value to both sender and recipient. As the value of data increases among users of information-sharing systems, the risks of unauthorized data modification, user identity theft, fraud, unauthorized access to data, data corruption, and a host of other business-related problems mainly dealing with data integrity and user authentication, are introduced. The issues of integrity and authentication play an important part in the economic systems of human society. Few would do business with companies and organizations that do not prove trustworthy or competent.

For example, the sentence "I owe Alice US$500" has a hash result of "gCWXVcL3fPV8VrJNajm8JKA==," while the sentence "I owe Alice US$5000" has a hash of "DSAyXRTza2bHLH46IPMrSq==." As can be seen, there is a big difference in hash results between the two sentences. If an attacker were trying to misappropriate the $4500 difference, hashing would allow detection.

Why Hash Algorithms Are Needed and the Problems They Solve

- Is the e-mail you received really from who it says it is?
- Can you ensure the credit card details you submit are going to the site you expected?
- Can you be sure the latest anti-virus, firewall, or operating system software upgrade you install is really from the vendor?
- Do you know if the Web link you click on is genuine?
- Does the program hash the password when performing authentication or just passing it in the clear?
- Is there a way to know who you are really dealing with when disclosing your personal details over the Internet?
- Are you really you?
- Has someone modified a Web page or file without authorization?
- Can you verify that your routers are forwarding data only to authorized peer routers?
- Has any of the data been modified in route to its destination?
- Can hash algorithms help answer these questions?

What Are Hash Algorithms?

A hash algorithm is a one-way mathematical function that is used to compress a large block of data into a smaller, fixed-size representation of that data.

To understand the concept of hash functions, it is helpful to review some underlying mathematical structures. One such structure is called a function. When hash functions were first introduced in the 1950s, the goal was to map a message into a smaller message called a message digest. This smaller message was used as a sort of shorthand of the original message. The digest was used originally for detection of random and unintended errors in processing and transmission by data processing equipment

Functions

A function is a mathematical structure that takes one or more variables and outputs a variable. To illustrate how scientists think about functions, one can think of a function in terms of a machine (see Exhibit 113.1). The machine in this illustration has two openings. In this case the input opening is labeled x and the output opening is labeled y. These are considered traditional names for input and output. The following are the basic processing steps of mathematical functions:

1. A number goes in.
2. Something is done to it.
3. The resulting number is the output.

The same thing is done to every number input into the function machine. Step 2 above describes the actual mathematical transformation done to the input value, or hashed value, which yields the resulting output, or hash result. In this illustration, Step 2 can be described as a mathematical rule as follows: x + 3 = y. In the language of mathematics, if x is equal to 1, then y equals 4. Similarly, if x is equal to 2, then y equals 5. In this illustration the function, or mathematical structure, called an algorithm, is: for every number x, add 3 to the number. The result, y, is dependent on what is input, x.

As another example, suppose that, to indicate an internal company product shipment, the number 43738 is exchanged. The hash function, or algorithm, is described as: multiply each number from left to right, and the first digit of any multiplied product above 9 is dropped. The hash function could be illustrated in mathematical notation as: x * the number to the right = y (see Exhibit 113.1).

The input into a hash algorithm can be of variable length, but the output is usually of fixed length and somewhat shorter in length than the original message. The output of a hash function is called a message digest. In the case of the above, the hash input was of arbitrary (and variable) length; but the hash result, or message digest, was of a fixed length of 1 digit, 8. As can be seen, a hash function provides a shorthand representation of the original message. This is also the concept behind error checking (checksums) done on data transmitted across communications links. Checksums provide a nonsecure method to check for message accuracy or message integrity. It is easy to see how the relatively weak mathematical functions described above could be manipulated by an intruder to change the hash output. Such weak algorithms could result in the successful alteration of message content leading to inaccurate messages. If you can understand the concept of what a function is and does, you are on your way to understanding the basic concepts embodied in hash functions. Providing data integrity and authentication for such applications requires reliable, secure hash algorithms.

Secure Hash Algorithms

A hash algorithm was defined earlier as a one-way mathematical function that is used to compress a large block of data into a smaller, fixed size representation of that data. An early application for hashing was in detecting unintentional errors in data processing. However, due to the critical nature of their use in the high-

EXHIBIT 113.1 The Hash Function

4 * 3	12
Drop the first digit (1) leaves	2
2 * next number (3)	6
6 * next number (7)	42
Drop the first digit (4) leaves	2
2 * next number (3)	6
6 * next number (8)	48
Drop the first digit (4)	8

security environments of today, hash algorithms must now also be resilient to deliberate and malicious attempts to break secure applications by highly motivated human attackers — more so than by erroneous data processing. The one-way nature of hash algorithms is one of the reasons they are used in public key cryptography. A one-way hash function processes a bit stream in a manner that makes it highly unlikely that the original message can be deduced by the output value. This property of a secure hash algorithm has significance in situations where there is zero tolerance for unauthorized data modification or if the identity of an object needs to be validated with a high assurance of accuracy. Applications such as user authentication and financial transactions are made more trustworthy by the use of hash algorithms.

Hash algorithms are called secure if they have the following properties:

- The hash result should not be predictable. It should be computationally impractical to recover the original message from the message digest (one-way property).
- No two different messages, over which a hash algorithm is applied, will result in the same digest (collision-free property).

Secure hash algorithms are designed so that any change to a message will have a high probability of resulting in a different message digest. As such, the message alteration can be detected by comparing hash results before and after hashing. The receiver can tell that a message has suspect validity by the fact that the message digest computed by the sender does not match the message digest computed by the receiver, assuming both parties are using the same hash algorithm. The most common hash algorithms as of this writing are based on Secure Hash Algorithm-1 (SHA-1) and Message Digest 5 (MD5).

Secure Hash Algorithm

SHA-1, part of the Secure Hash Standard (SHS), was one of the earliest hash algorithms specified for use by the U.S. federal government (see Exhibit 113.2). SHA-1 was developed by NIST and the NSA. SHA-1 was published as a federal government standard in 1995. SHA-1 was an update to the SHA, which was published in 1993.

How SHA-1 Works

Think of SHA-1 as a hash machine that has two openings, input and output. The input value is called the hashed value, and the output is called the hash result. The hashed values are the bit streams that represent an electronic message or other data object. The SHA-1 hash function, or algorithm, transforms the hashed value by performing a mathematical operation on the input data. The length of the message is the same as the number of bits in the message. The SHA-1 algorithm processes blocks of 512 bits in sequence when computing the message digest. SHA-1 produces a 160-bit message digest. SHA-1 has a limitation on input message size of less than 18 quintillion (that is, 2^{64} or 18,446,744,073,709,551,616) bits in length.

SHA-1 has five steps to produce a message digest:

1. Append padding to make message length 64 bits less than a multiple of 512.
2. Append a 64-bit block representing the length of the message before padding out.
3. Initialize message digest buffer with five hexadecimal numbers. These numbers are specified in the FIPS 180-1 publication.
4. The message is processed in 512-bit blocks. This process consists of 80 steps of processing (four rounds of 20 operations), reusing four different hexadecimal constants, and some shifting and adding functions.
5. Output blocks are processed into a 160-bit message digest.

EXHIBIT 113.2 Output Bit Lengths

Hash Algorithm	Output Bit Length
SHA-1	160
SHA-256	256
SHA-384	384
SHA-512	512

MD5

SHA was derived from the secure hash algorithms MD4 and MD5, developed by Professor Ronald L. Rivest of MIT in the early 1990s. As can be expected, SHA and MD5 work in a similar fashion. While SHA-1 yields a 160-bit message digest, MD5 yields a 128-bit message digest. SHA-1, with its longer message digest, is considered more secure than MD5 by modern cryptography experts, due in part to the longer output bit length and resulting increased collision resistance. However, MD5 is still in common use as of this writing.

Keyed Hash (HMAC)

Modern cryptographers have found the hash algorithms discussed above to be insufficient for extensive use in commercial cryptographic systems or in private electronic communications, digital signatures, electronic mail, electronic funds transfer, software distribution, data storage, and other applications that require data integrity assurance, data origin authentication, and the like. The use of asymmetric cryptography and, in some cases, symmetric cryptography, has extended the usefulness of hashing by associating identity with a hash result. The structure used to convey the property of identity (data origin) with a data object's integrity is hashed message authentication code (HMAC), or keyed hash.

For example, how does one know if the message and the message digest have not been tampered with? One way to provide a higher degree of assurance of identity and integrity is by incorporating a cryptographic key into the hash operation. This is the basis of the keyed hash or hashed message authentication code (HMAC). The purpose of a message authentication code (MAC) is to provide verification of the source of a message and integrity of the message without using additional mechanisms. Other goals of HMAC are as follows:

- To use available cryptographic hash functions without modification
- To preserve the original performance of the selected hash without significant degradation
- To use and handle keys in a simple way
- To have a well-understood cryptographic analysis of the strength of the mechanism based on reasonable assumptions about the underlying hash function
- To enable easy replacement of the hash function in case a faster or stronger hash is found or required

To create an HMAC, an asymmetric (public/private) or a symmetric cryptographic key can be appended to a message and then processed through a hash function to derive the HMAC. In mathematical terms, if x = (key + message) and f = SHA-1, then f(x) = HMAC. Any hash function can be used, depending on the protocol defined, to compute the type of message digest called an HMAC. The two most common hash functions are based on MD5 and SHA. The message data and HMAC (message digest of a secret key and message) are sent to the receiver. The receiver processes the message and the HMAC using the shared key and the same hash function as that used by the originator. The receiver compares the results with the HMAC included with the message. If the two results match, then the receiver is assured that the message is authentic and came from a member of the community that shares the key.

Other examples of HMAC usage include challenge–response authentication protocols such as Challenge Handshake Authentication Protocol (CHAP, RFC 1994). CHAP is defined as a peer entity authentication method for Point-to-Point Protocol (PPP), using a randomly generated challenge and requiring a matching response that depends on a cryptographic hash of the challenge and a secret key. Challenge–Response Authentication Mechanism (CRAM, RFC 2195), which specifies an HMAC using MD5, is a mechanism for authenticating Internet Mail Access Protocol (IMAP4) users. Digital signatures, used to authenticate data origin and integrity, employ HMAC functions as part of the "signing" process. A digital signature is created as follows:

1. A message (or some other data object) is input into a hash function (i.e., SHA-1, MD5, etc.).
2. The hash result is encrypted by the private key of the sender.

The result of these two steps yields what is called a *digital signature* of the message or data object. The properties of a cryptographic hash ensure that, if the data object is changed, the digital signature will no longer match it. There is a difference between a digital signature and an HMAC. An HMAC uses a shared secret key (symmetric cryptography) to "sign" the data object, whereas a digital signature is created by using a private key from a private/public key pair (asymmetric cryptography) to sign the data object. The strengths of digital signatures lend themselves to use in high-value applications that require protection against forgery and fraud.

See Exhibit 113.3 for other hash algorithms.

EXHIBIT 113.3 Other Hash Algorithms

Hash Algorithm	Output Bit Length	Country
RIPEMD (160,256,320)	160, 256, 320	Germany, Belgium
HAS-160	160	Korea
Tiger	128,160,192	United Kingdom

How Hash Algorithms Are Used in Modern Cryptographic Systems

In the past, hash algorithms were used for rudimentary data integrity and user authentication; today hash algorithms are incorporated into other protocols — digital signatures, virtual private network (VPN) protocols, software distribution and license control, Web page file modification detection, database file system integrity, and software update integrity verification are just a few. Hash algorithms used in hybrid cryptosystems discussed next.

Transport Layer Security (TLS)

TLS is a network security protocol that is designed to provide data privacy and data integrity between two communicating applications. TLS was derived from the earlier Secure Sockets Layer (SSL) protocol developed by Netscape in the early 1990s. TLS is defined in IETF RFC 2246. TLS and SSL do not interoperate due to differences between the protocols. However, TLS 1.0 does have the ability to drop down to the SSL protocol during initial session negotiations with an SSL client. Deference is given to TLS by developers of most modern security applications. The security features designed into the TLS protocol include hashing.

The TLS protocol is composed of two layers:

1. The Record Protocol provides in-transit data privacy by specifying that symmetric cryptography be used in TLS connections. Connection reliability is accomplished by the Record Protocol through the use of HMACs.
2. TLS Handshake Protocol (really a suite of three subprotocols). The Handshake Protocol is encapsulated within the Record Protocol. The TLS Handshake Protocol handles connection parameter establishment. The Handshake Protocol also provides for peer identity verification in TLS through the use of asymmetric (public/private) cryptography.

There are several uses of keyed hash algorithms (HMAC) within the TLS protocol.

TLS uses HMAC in a conservative fashion. The TLS specification calls for the use of both HMAC MD5 and HMAC SHA-1 during the Handshake Protocol negotiation. Throughout the protocol, two hash algorithms are used to increase the security of various parameters:

- Pseudorandom number function
- Protect record payload data
- Protect symmetric cryptographic keys (used for bulk data encrypt/decrypt)
- Part of the mandatory cipher suite of TLS

If any of the above parameters were not protected by security mechanisms such as HMACs, an attacker could thwart the electronic transaction between two or more parties. The TLS protocol is the basis for most Web-based in-transit security schemes. As can be seen by this example, hash algorithms provide an intrinsic security value to applications that require secure in-transit communication using the TLS protocol.

IPSec

The Internet Protocol Security (IPSec) Protocol was designed as the packet-level security layer included in IPv6. IPv6 is a replacement TCP/IP protocol suite for IPv4. IPSec itself is flexible and modular in design, which allows the protocol to be used in current IPv4 implementations. Unlike the session-level security of TLS, IPSec provides packet-level security. VPN applications such as intranet and remote access use IPSec for communications security.

Two protocols are used in IPSec operations, Authentication Header (AH) and Encapsulating Security Payload (ESP). Among other things, ESP is used to provide data origin authentication and connectionless integrity. Data origin authentication and connectionless integrity are joint services and are offered as an option in the implementation of the ESP. RFC 2406, which defines the ESP used in IPSec, states that either HMAC or one-way hash algorithms may be used in implementations. The authentication algorithms are used to create the integrity check value (ICV) used to authenticate an ESP packet of data. HMACs ensure the rapid detection and rejection of bogus or replayed packets. Also, because the authentication value is passed in the clear, HMACs are mandatory if the data authentication feature of ESP is used. If data authentication is used, the sender computes the integrity check value (ICV) over the ESP packet contents minus the authentication data. After receiving an IPSec data packet, the receiver computes and compares the ICV of the received datagrams. If they are the same, then the datagram is authentic; if not, then the data is not valid, it is discarded, and the event can be logged. MD5 and SHA-1 are the currently supported authentication algorithms.

The AH protocol provides data authentication for as much of the IP header as possible. Portions of the IP header are not authenticated due to changes to the fields that are made as a matter of routing the packet to its destination. The use of HMAC by the ESP has, according to IPSec VPN vendors, negated the need for AH.

Digital Signatures

Digital signatures serve a similar purpose as those of written signatures on paper — to prove the authenticity of a document. Unlike a pen-and-paper signature, a digital signature can also prove that a message has not been modified. HMACs play an important role in providing the property of integrity to electronic documents and transactions. Briefly, the process for creating a digital signature is very much like creating an HMAC. A message is created, and the message and the sender's private key (asymmetric cryptography) serve as inputs to a hash algorithm. The hash result is attached to the message. The sender creates a symmetric session encryption key to optionally encrypt the document. The sender then encrypts the session key with the sender's private key, reencrypts it with the receiver's public key to ensure that only the receiver can decrypt the session key, and attaches the signed session key to the document. The sender then sends the digital envelope (keyed hash value, encrypted session key, and the encrypted message) to the intended receiver. The receiver performs the entire process in reverse order. If the results match when the receiver decrypts the document and combines the sender's public key with the document through the specified hash algorithm, the receiver is assured that (1) the message came from the original sender and (2) the message has not been altered. The first case is due to use of the sender's private key as part of the hashed value. In asymmetric cryptography, a mathematical relationship exists between the public and private keys such that either can encrypt and decrypt; but the same key cannot both encrypt and decrypt the same item. The private key is known only to its owner. As such, only the owner of the private key could have used it to develop the HMAC.

Other Applications

HMACs are useful when there is a need to validate software that is downloaded from download sites. HMACs are used in logging onto various operating systems, including UNIX. When the user enters a password, the password is usually run through a hash algorithm; and the hashed result is compared to a user database or password file.

An interesting use of hash algorithms to prevent software piracy is in the Windows XP registration process. SHA-1 is used to develop the installation ID used to register the software with Microsoft.

During installation of Windows XP, the computer hardware is identified, reduced to binary representation, and hashed using MD5. The hardware hash is an eight-byte value that is created by running ten different pieces of information from the PC's hardware components through the MD5 algorithm. This means that the resultant hash value cannot be backward-calculated to determine the original values. Further, only a portion of the resulting hash value is used in the hardware hash to ensure complete anonymity.

Unauthorized file modification such as Web page defacement, system file modification, virus signature update, signing XML documents, and signing database keys are all applications for which various forms of hashing can increase security levels.

Problems with Hash Algorithms

Flaws have been discovered in various hash algorithms. One such basic flaw is called the birthday attack.

Birthday Attack

This attack's name comes from the world of probability theory out of any random group of 23 people, it is probable that at least two share a birthday. Finding two numbers that have the same hash result is known as the birthday attack. If hash function f maps into message digests of length 60 bits, then an attacker can find a collision using only 230 inputs ($2^{f/2}$). Differential cryptanalysis has proven to be effective against one round of MD5. (There are four rounds of transformation defined in the MD5 algorithm.) When choosing a hash algorithm, speed of operation is often a priority. For example, in asymmetric (public/private) cryptography, a message may be hashed into a message digest as a data integrity enhancement. However, if the message is large, it can take some time to compute a hash result. In consideration of this, a review of speed benchmarks would give a basis for choosing one algorithm over another. Of course, implementation in hardware is usually faster than in a software-based algorithm.

Looking to the Future

SHA-256, -384, and -512

In the summer of 2001, NIST published for public comment a proposed update to the Secure Hash Standard (SHS) used by the U.S. government. Although SHA-1 appears to be still part of SHS, the update includes the recommendation to use hash algorithms with longer hash results. Longer hash results increase the work factor needed to break cryptographic hashing. This update of the Secure Hash Standard coincides with another NIST update — selection of the Rijndael symmetric cryptography algorithm for U.S. government use for encrypting data. According to NIST, it is thought that the cryptographic strength of Rijndael requires the higher strength of the new SHS algorithms. The new SHS algorithms feature similar functions but different structures. Newer and more secure algorithms, such as SHA-256, -384, and –512, may be integrated into the IPSec specification in the future to complement the Advanced Encryption Standard (AES), Rijndael. In May 2002, NIST announced that the Rijndael algorithm had been selected as the AES standard, FIPS 197.

Summary

Hash algorithms have existed in many forms at least since the 1950s. As a result of the increased value of data interactions and the increased motivation of attackers seeking to exploit electronic communications, the requirements for hash algorithms have changed. At one time, hashing was used to detect inadvertent errors generated by data processing equipment and poor communication lines. Now, secure hash algorithms are used to associate source of origin with data integrity, thus tightening the bonds of data and originator of data. So-called HMACs facilitate this bonding through the use of public/private cryptography. Protocols such as TLS and IPSec use HMACs extensively. Over time, weaknesses in algorithms have been discovered and hash algorithms have improved in reliability and speed. The present digital economy finds that hash algorithms are useful for creating message digests and digital signatures.

Further Reading

http://www.deja.com/group/sci.crypt.

114

A Look at the Advanced Encryption Standard (AES)

Ben Rothke, CISSP

In the early 1970s, the Data Encryption Standard (DES) became a Federal Information Processing Standard[1,2] (FIPS). This happened with little fanfare and even less public notice. In fact, in the late 1960s and early 1970s, the notion of the general public having an influence on U.S. cryptographic policy was utterly absurd. It should be noted that in the days before personal computers were ubiquitous, the force of a FIPS was immense, given the purchasing power of the U.S. government. Nowadays, the power of a FIPS has a much lesser effect on the profitability of computer companies given the strength of the consumer market.

Jump to the late 1990s and the situation is poles apart. The proposed successor to DES, the Advanced Encryption Standard (AES), was publicized not only in the *Federal Register* and academic journals, but also in consumer computing magazines and the mainstream media.[3]

The entire AES selection process was, in essence, a global town hall event. This was evident from submissions from cryptographers from around the world. The AES process was completely open to public scrutiny and comment. This is important because, when it comes to the design of effective encryption algorithms, history has shown time and time again that secure encryption algorithms cannot be designed, tested, and verified in a vacuum. In fact, if a software vendor decides to use a proprietary encryption algorithm, that immediately makes the security and efficacy of the algorithm suspect.[4] Prudent consumers of cryptography will *never* use a proprietary algorithm.

This notion is based on what is known as Kerckhoff's assumption.[5] This assumption states the security of a cryptosystem should rest entirely in the secrecy of the key and not in the secrecy of the algorithm. History has shown, and unfortunately, that some software vendors still choose to ignore the fact that completely open-source encryption algorithms are the only way to design a truly world-class encryption algorithm.

The AES Process

In January 1997, the National Institute of Standards and Technology (NIST, a branch within the Commerce Department) commenced the AES process.[6] A replacement for DES was needed due to the ever-growing frailty of DES. Not that any significant architectural breaches were found in DES; rather, Moore's law had caught up with it. By 1998, it was possible to build a DES-cracking device for a reasonable sum of money.

The significance of the availability of a DES-cracking device to an adversary cannot be understated because DES is the world's most widely used, general-purpose cryptosystem. For the details of this cracking of DES,[7] see *Cracking DES: Secrets of Encryption Research, Wiretap Politics and Chip Design* by the Electronic Frontier Foundation (1998, O'Reilly & Assoc.).

DES was reengineered and put back into working order via the use of Triple-DES. Triple-DES takes the input data and encrypts it three times. Triple-DES (an official standard in use as ANSI X9.52-1998[8]) is resilient against brute-force attacks, and from a security perspective, it is adequate. So why not simply use Triple-DES

as the new AES? This is not feasible because DES was designed to be implemented in hardware and is therefore not efficient in software implementations. Triple-DES is three times slower than DES; and although DES is fast enough, Triple-DES is far too slow. One of the criteria for AES is that it must be efficient when implemented in software, and the underlying architecture of Triple-DES makes it unsuitable as an AES candidate.

The AES specification called for a symmetric algorithm (same key for encryption and decryption) using block encryption of 128 bits in size, with supporting key sizes of 128, 192, and 256 bits. The algorithm was required to be royalty-free for use worldwide and offer security of a sufficient level to protect data for 30 years. Additionally, it must be easy to implement in hardware as well as software, and in restricted environments (i.e., smart cards, DSP, cell phones, FPGA, custom ASIC, satellites, etc.).

AES will be used for securing sensitive but unclassified material by U.S. government agencies.[9] As a likely outcome, all indications make it likely that it will, in due course, become the *de facto* encryption standard for commercial transactions in the private sector as well.

In August 1998, NIST selected 15 preliminary AES candidates at the first AES Candidate Conference in California. At that point, the 15 AES candidates were given much stronger scrutiny and analysis within the global cryptography community. Also involved with the process was the National Security Agency (NSA).

This is not the place to detail the input of the NSA into the AES selection process, but it is obvious that NIST learned its lesson from the development of DES. An initial complaint against DES was that IBM kept its design principles secret at the request of the U.S. government. This, in turn, led to speculation that there was some sort of trapdoor within DES that would provide the U.S. intelligence community with complete access to all encrypted data. Nonetheless, when the DES design principles were finally made public in 1992,[10] such speculation was refuted.

The AES Candidates

The 15 AES candidates chosen at the first AES conference are listed in Exhibit 114.1.

A second AES Candidate Conference was held in Rome in March 1999 to present analyses of the first-round candidate algorithms. After this period of public scrutiny, in August 1999, NIST selected five algorithms for more extensive analysis (see Exhibit 114.2).

In October 2000, after more than 18 months of testing and analysis, NIST announced that the Rijndael algorithm had been selected as the AES candidate. It is interesting to note that only days after NIST's announcement selecting Rijndael, advertisements were already springing up stating support for the new standard.

In February 2001, NIST made available a Draft AES FIPS[11] for public review and comment, which concluded on May 29, 2001.

This was followed by a 90-day comment period from June through August 2001. In August 2002, NIST announced the approval of Federal Information Processing Standards (FIPS) 180-2, Secure Hash Standard, which contains the specifications for the Secure Hash Algorithm (SHA-1, SHA-256, SHA-384, and SHA-512).

DES Is Dead

It is clear that not only is 56-bit DES ineffective, it is dead. From 1998 on, it is hoped that no organization has implemented 56-bit DES in any type of high-security or mission-critical system. If such is the case, it should be immediately retrofitted with Triple-DES or another secure public algorithm.

Although DES was accepted as an ANSI standard in 1981 (ANSI X3.92) and later incorporated into several American Banking Association Financial Services (X9) standards, it has since been replaced by Triple-DES.

Replacing a cryptographic algorithm is a relatively straightforward endeavor because encryption algorithms are, in general, completely interchangeable. Most hardware implementations allow plug-ins and replacements of different algorithms. The greatest difficulty is in the logistics of replacing the software for companies with tens or hundreds of thousands of disparate devices. Also, for those organizations that have remote sites, satellites, etc., this point is ever more germane.

AES implementations have already emerged in many commercial software security products as an optional algorithm (in addition to Triple-DES and others). Software implementations have always come before hardware products due to the inherent time it takes to design and update hardware. It is generally easier to upgrade software than to perform a hardware replacement or upgrade, and many vendors have already incorporated AES into their latest designs.

EXHIBIT 114.1 AES Candidates Chosen at the First AES Conference

Algorithm	Submitted by	Overview[a]
CAST-256	Entrust Technologies, Canada	A 48-round unbalanced Feistel cipher using the same round functions as CAST-128, which use + — XOR rotates and 4 fixed 6-bit S-boxes; with a key schedule.
Crypton	Future Systems, Inc., Korea	A 12-round iterative cipher with a round function using & | XOR rotates and 2 fixed 8-bit S-boxes; with various key lengths supported, derived from the previous SQUARE cipher.
DEAL	Richard Outerbridge (UK) and Lars Knudsen (Norway)	A rather different proposal, a 6- to 8-round Feistel cipher which uses the existing DES as the round function. Thus a lot of existing analysis can be leveraged, but at a cost in speed.
DFC	Centre National pour la Recherche Scientifique, France	An 8-round Feistel cipher design based on a decorrelation technique and using + x and a permutation in the round function; with a 4-round key schedule.
E2	Nippon Telegraph and Telephone Corporation, Japan	A 12-round Feistel cipher, using a nonlinear function comprised of substitution using a single fixed 8-bit S-box, a permutation, XOR mixing operations, and a byte rotation.
FROG	TecApro International, South Africa	An 8-round cipher, with each round performing four basic operations (with XOR, substitution using a single fixed 8-bit S-box, and table value replacement) on each byte of its input.
HPC	Rich Schroeppel, United States	An 8-round Feistel cipher, which modifies 8 internal 64-bit variables as well as the data using + — x & | XOR rotates and a lookup table.
LOKI97	Lawrie Brown, Josef Pieprzyk, and Jennifer Seberry, Australia	A 16-round Feistel cipher using a complex round function f with two S-P layers with fixed 11-bit and 13-bit S-boxes, a permutation, and + XOR combinations; and with a 256-bit key schedule using 48 rounds of an unbalanced Feistel network using the same complex round function f.
Magenta	Deutsche Telekom, Germany	A 6- to 8-round Feistel cipher, with a round function that uses a large number of substitutions using a single fixed S-box (based on exponentiation on $GF(2^8)$), that is combined together with key bits using XOR.
MARS	IBM, United States	An 8+16+8-round unbalanced Feistel cipher with four distinct phases: key addition and 8 rounds of unkeyed forward mixing, 8 rounds of keyed forwards transformation, 8 rounds of keyed backwards transformation, and 8 rounds of unkeyed backwards mixing and keyed subtraction. The rounds use + — x rotates XOR and two fixed 8-bit S-boxes.
RC6	RSA Laboratories, United States	A 20-round iterative cipher, developed from RC5 (and fully parameterized), which uses a number of 32-bit operations (+ — x XOR rotates) to mix data in each round.
Rijndael	Joan Daemen and Vincent Rijmen, Belgium	A 10- to 14-round iterative cipher, using byte substitution, row shifting, column mixing, and key addition, as well as an initial and final round of key addition, derived from the previous SQUARE cipher.
SAFER+	Cylink Corp., United States	An 8- to 16-round iterative cipher, derived from the earlier SAFER cipher. SAFER+ uses + x XOR and two fixed 8-bit S-boxes.
SERPENT	Ross Anderson (U.K.), Eli Biham (Israel), and Lars Knudsen (Norway)	A 32-round Feistel cipher, with key mixing using XOR and rotates, substitutions using 8 key-dependent 4-bit S-boxes, and a linear transformation in each round.
Twofish	Bruce Schneier et al., United States	A 16-round Feistel cipher using four key-dependent 8-bit S-boxes, matrix transforms, rotations, and based in part on the Blowfish cipher.

[a]From http://www.adfa.edu.au/~lpb/papers/unz99.html.

EXHIBIT 114.2 Five Algorithms Selected by NIST

Algorithm	Main Strength	Main Weaknesses
MARS	High security margin	Complex implementation
RC6	Very simple	Lower security margin as it used operations specific to 32-bit processors
Rijndael	Simple elegant design	Insufficient rounds
Serpent	High security margin	Complex design and analysis, poor performance
Twofish	Reasonable performance, high security margin	Complex design

For those organizations already running Triple-DES, there are not many compelling reasons (except for compatibility) to immediately use AES. It is likely that the speed at which companies upgrade to AES will increase as more products ship in AES-enabled mode.

Rijndael

Rijndael, the AES candidate, was developed by Dr. Joan Daemen of Proton World International and Dr. Vincent Rijmen, a postdoctoral researcher in the electrical engineering department of Katholieke Universiteit of the Netherlands.[12] Drs. Daemen and Rijmen are well-known and respected in the cryptography community. Rijndael has its roots in the SQUARE cipher,[13] also designed by Daemen and Rijmen.

The details on Rijndael are specified in its original AES proposal.[14] From a technical perspective,[15] Rijndael is a substitution-linear transformation network (i.e., non-Feistel[16,17]) with multiple rounds, depending on the key size. Rijndael's key length and block size is either 128, 192, or 256 bits. It does not support arbitrary sizes, and its key and block size must be one of the three lengths.

Rijndael uses a single S-box that acts on a byte input in order to give a byte output. For implementation purposes, it can be regarded as a lookup table of 256 bytes. Rijndael is defined by the equation

$$S(x) = M\ (1/x) + b$$

over the field $GF(2^8)$, where M is a matrix and b is a constant.

A data block to be processed under Rijndael is partitioned into an array of bytes and each of the cipher operations is byte oriented. Rijndael's ten rounds each perform four operations. In the first layer, an 8×8 S-box (S-boxes used as nonlinear components) is applied to each byte. The second and third layers are linear mixing layers, in which the rows of the array are shifted and the columns are mixed. In the fourth layer, subkey bytes are XORed into each byte of the array. In the last round, the column mixing is omitted.[18]

Why Did NIST Select the Rijndael Algorithm?

According to the NIST,[19] Rijndael was selected due to its combination of security, performance, efficiency, ease of implementation, and flexibility.[20] Specifically, NIST felt that Rijndael was appropriate for the following reasons:

- Good performance in both hardware and software across a wide range of computing environments
- Good performance in both feedback and nonfeedback modes
- Key setup time is excellent
- Key agility is good
- Very low memory requirements
- Easy to defend against power and timing attacks (this defense can be provided without significantly impacting performance).

Problems with Rijndael

Although the general consensus is that Rijndael is a fundamentally first-rate algorithm, it is not without opposing views.[21] One issue was with its underlying architecture; some opined that its internal mathematics were simple, almost to the point of being rudimentary. If Rijndael were written down as a mathematical formula, it would look much simpler than any other AES candidate. Another critique was that Rijndael avoids any kind of obfuscation technique to hide its encryption mechanism from adversaries.[22] Finally, it was pointed out that encryption and decryption use different S-boxes, as opposed to DES which uses the same S-boxes for both operations. This means that an implementation of Rijndael that both encrypts and decrypts is twice as large as an implementation that only does one operation, which may be inconvenient on constrained devices.

The Rijndael team defended its design by pointing out that the simpler mathematics made Rijndael easier to implement in embedded hardware. The team also argued that obfuscation was not needed. This, in turn, led to speculation that the Rijndael team avoided obfuscation to evade scrutiny from Hitachi, which had expressed its intentions to seek legal action against anyone threatening its U.S.-held patents. Hitachi claimed to hold exclusive patents on several encryption obfuscation techniques, and had not been forthcoming about whether it would consider licensing those techniques to any outside party.[23] In fact, in early 2000, Hitachi issued patent claims against four of the AES candidates (MARS, RC6, Serpent, and Twofish).

Can AES Be Cracked?

Although a public-DES cracker has been built[24] as detailed in *Cracking DES: Secrets of Encryption Research, Wiretap Politics and Chip Design*, there still exists the question of whether an AES-cracking device can be built?

It should be noted that after nearly 30 years of research, no easy attack against DES has been discovered. The only feasible attack against DES is a brute-force exhaustive search of the entire keyspace. Had the original keyspace of DES been increased, it is unlikely that the AES process would have been undertaken.

DES-cracking machines were built that could recover a DES key after a number of hours by trying all possible key values. Although an AES cracking machine could also be built, the time that would be required to extricate a single key would be overwhelming.

As an example, although the entire DES keyspace can feasibly be cracked in less than 48 hours, this is not the case with AES. If a special-purpose chip, such as a field-programmable gate array[25] (FPGA), could perform a billion AES decryptions per second, and the cracking host had a billion chips running in parallel, it would still require an infeasible amount of time to recover the key. Even if it was assumed that one could build a machine that could recover a DES key in a second (i.e., try 2^{55} keys per second), it would take that machine over 140 trillion years to crack a 128-bit AES key.

Given the impenetrability of AES (at least with current computing and mathematical capabilities), it appears that AES will fulfill its requirement of being secure until 2030. But then again, a similar thought was assumed for DES when it was first designed.

Finally, should quantum computing transform itself from the laboratory to the realm of practical application, it could potentially undermine the security afforded by AES and other cryptosystems.

The Impact of AES

The two main bodies to put AES into production will be the U.S. government and financial services companies. For both entities, the rollout of AES will likely be quite different.

For the U.S. government sector, after AES is confirmed as a FIPS, all government agencies will be required to use AES for secure (but unclassified) systems. Because the government has implemented DES and Triple-DES in tens of thousands of systems, the time and cost constraints for the upgrade to AES will be huge.

AES will require a tremendous investment of time and resources to replace DES, Triple-DES, and other encryption schemes in the current government infrastructure. A compounding factor that can potentially slow down the acceptance of AES is the fact that because Triple-DES is fundamentally secure (its main caveat is its speed), there is no compelling security urgency to replace it. Although AES may be required, it may be easier for government agencies to apply for a waiver for AES as opposed to actually implementing it.[26] With the

budget and time constraints of interchanging AES, its transition will occur over time, with economics having a large part in it.

The financial services community also has a huge investment in Triple-DES. Because there is currently no specific mandate for AES use in the financial services community, and given the preponderance of Triple-DES, it is doubtful that any of the banking standards bodies will require AES use.

While the use of single DES (also standardized as X9.23-1995, Encryption of Wholesale Financial Messages) is being withdrawn by the X9 committee (see X9 TG-25-1999); this nonetheless allows continued use of DES until another algorithm is implemented.

But although the main advantages of AES are its efficiency and performance for both hardware and software implementations, it may find a difficult time being implemented in large-scale nongovernmental sites, given the economic constraints of upgrading it, combined with the usefulness of Triple-DES. Either way, it will likely be a number of years before there is widespread use of the algorithm.

Notes

1. FIPS 46-3, see http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf. Reaffirmed for the final time on October 25, 1999.
2. Under the Information Technology Management Reform Act (Public Law 104-106), the Secretary of Commerce approves standards and guidelines that are developed by the National Institute of Standards and Technology (NIST) for federal computer systems. These standards and guidelines are issued by NIST as Federal Information Processing Standards (FIPS) for use governmentwide. NIST develops FIPS when there are compelling federal government requirements, such as for security and interoperability, and there are no acceptable industry standards or solutions.
3. While IBM and the U.S. government essentially designed DES between them in what was billed as a public process, it attracted very little public interest at the time.
4. See B. Schneier, Security in the Real World: How to Evaluate Security Technology, *Computer Security Journal,* 15(4), 1999; and B. Rothke, Free Lunch, *Information Security Magazine,* Feb. 1999, www.infosecuritymag.com.
5. There are actually six assumptions. Dutch cryptographer Auguste Kerckhoff wrote *La Cryptographie Militare* (Military Cryptography) in 1883. His work set forth six highly desirable elements for encryption systems:
   a. A cipher should be unbreakable. If it cannot be theoretically proven to be unbreakable, it should at least be unbreakable in practice.
   b. If one's adversary knows the method of encipherment, this should not prevent one from continuing to use the cipher.
   c. It should be possible to memorize the key without having to write it down, and it should be easy to change to a different key.
   d. Messages, after being enciphered, should be in a form that can be sent by telegraph.
   e. If a cipher machine, code book, or the like is involved, any such items required should be portable and usable by one person without assistance.
   f. Enciphering or deciphering messages in the system should not cause mental strain, and should not require following a long and complicated procedure.
6. http://csrc.nist.gov/encryption/aes/pre-round1/aes_9701.txt.
7. Details are also available at www.eff.org/descracker.html.
8. The X9.52 standard defines triple-DES encryption with keys k_1, k_2 *and* k_3; k_3 as: $C = E_{k3} (D_{k2} (E_{k1} (M)))$ where E_k and D_k denote DES encryption and DES decryption, respectively, with the key k.
9. It should be noted that AES (like DES) will only be used to protect sensitive but unclassified data. Classified data is protected by separate, confidential algorithms.
10. Dan Coppersmith, The Data Encryption Standard and Its Strength Against Attacks, IBM Report RC18613.
11. http://csrc.nist.gov/encryption/aes/draftfips/fr-AES-200102.html.
12. For a quick technical overview of Rijnadel, see http://www.baltimore.com/devzone/aes/tech_overview.html.
13. www.esat.kuleuven.ac.be/~rijmen/square/index.html.
14. Available at www.esat.kuleuven.ac.be/~rijmen/rijndael/rijndaeldocV2.zip.

15. http://csrc.nist.gov/encryption/aes/round2/r2report.pdf.
16. Feistel ciphers are block ciphers in which the input is split in half. Feistel ciphers are provably invertible. Decryption is the algorithm in reverse, with subkeys used in the opposite order.
17. Of the four other AES finalists, MARS uses an extended Feistel network; RC6 and Twofish use a standard Feistel network; and Serpent uses a single substitution-permutation network.
18. Known as the key schedule, the Rijndael key (which is from 128 to 256 bits) is fed into the key schedule. This key schedule is used to generate the sub-keys, which are the keys used for each round. Each sub-key is as long as the block being enciphered, and thus, if 128 bits long, is made up of 16 bytes. A good explanation of the Rijndael key schedule can be found at http://home.ecn.ab.ca/~jsavard/crypto/co040801.htm.
19. http://csrc.nist.gov/encryption/aes.
20. As clarified in the report by NIST (*Report on the Development of the Advanced Encryption Standard)*, the fact that NIST rejected MARS, RC6, Serpent, and Twofish does not mean that they were inadequate for independent use. Rather, the sum of all benefits dictated that Rijndael was the best candidate for the AES. The report concludes that "all five algorithms appear to have adequate security for the AES."
21. Improved Cryptanalysis of Rijndael, N. Ferguson, J. Kelsey, et al., www.counterpane.com/rijndael.html.
22. Contrast this with Twofish; see *The Twofish Team's Final Comments on AES Selection*, www.counterpane.com/twofish-final.html.
23. www.planetit.com/techcenters/docs/security/qa/PIT20001106S0015.
24. It is an acceptable assumption to believe that the NSA has had this capability for a long time.
25. An FPGA is an integrated circuit that can be programmed in the field after manufacture. They are heavily used by engineers in the design of specialized integrated circuits that can later be produced in large quantities for distribution to computer manufacturers and end users.
26. Similar to those government agencies that applied for waivers to get out of the requirement for C2 (*Orange Book*) certification.

For Further Information

1. Savard, John, How Does Rijndael Work? www.securityportal.com/articles/rijndael20001012. html and http://home.ecn.ab.ca/~jsavard/crypto/co040801.htm.
2. Tsai, Melvin, AES: An Overview of the Rijndael Encryption Algorithm, www.gigascale.org/mescal/forum/65.html.
3. Landau, Susan, Communications Security for the Twenty-first Century: The Advanced Encryption Standard and Standing the Test of Time: The Data Encryption Standard, www.ams.org/notices/200004/fea-landau.pdf and www.ams.org/notices/200003/fea-landau.pdf.
4. Schneier, Bruce, *Applied Cryptography,* John Wiley & Sons, 1996.
5. Menezes, Alfred, *Handbook of Applied Cryptography,* CRC Press, 1996.
6. Anderson, Ross, *Security Engineering,* John Wiley & Sons, 2001.
7. Brown, Lawrie, A Current Perspective on Encryption Algorithms, http://www.adfa.edu.au/~lpb/papers/unz99.html.

Introduction to Encryption

Jay Heiser

THROUGHOUT RECORDED HISTORY, NEW FORMS OF COMMUNICATION HAVE BEEN PARALLELED BY DEVELOPMENTS IN CRYPTOGRAPHY, THE PRACTICE OF SECURING COMMUNICATIONS. Secret writing appeared soon after the development of writing itself — an Egyptian example from 1900 BC is known. During the Renaissance, the significance of the nation state and growth in diplomacy created a requirement for secret communication systems to support diplomatic missions located throughout Europe and the world. The high volume of encrypted messages, vulnerable to interception through slow and careless human couriers, encouraged the first organized attempts to systematically break secret communications. Several hundred years later, the widespread use of the telegraph, and especially the use of radio in World War I, forced the development of efficient and robust encryption techniques to protect the high volume of sensitive communications vulnerable to enemy surveillance. At the start of World War II, highly complex machines, such as the German Enigma, were routinely used to encipher communications. Despite the sophistication of these devices, commensurate developments in cryptanalysis, the systematic technique of determining the plain text content of an encrypted message, provided the Allies with regular access to highly sensitive German and Japanese communications.

The ubiquity of computers — and especially the growth of the Internet — has created a universal demand for secure high-volume, high-speed communications. Governments, businesses of all sizes, and even private individuals now have a routine need for protected Internet communications. Privacy is just one of the necessary services that cryptography is providing for E-commerce implementations. The burgeoning virtual world of online transactions has also created a demand for virtual trust mechanisms. Cryptological techniques, especially those associated with public key technology, enable highly secure identification mechanisms, digital

signature, digital notary services, and a variety of trusted electronic transaction types to replace paper and human mechanisms.

HOW ENCRYPTION FAILS

Encryption has a history of dismal failures. Like any other human device, it can always be circumvented by humans; it is not a universal panacea to security problems. Having an understanding of how encryption implementations are attacked and how they fail is crucial in being able to successfully apply encryption.

CRYPTOGRAPHIC ATTACKS

Brute-force attack is the sequential testing of each possible key until the correct one is found. On average, the correct key will be found once half of the total key space has been tried. The only defense against a brute-force attack is to make the key space so huge that such an attack is *computationally infeasible* (i.e., theoretically possible, but not practical given the current cost/performance ratio of computers). As processing power has increased, the limits of computational infeasibility have been reduced, encouraging the use of longer keys. A 128-bit key space is an awesomely large number of keys — contemporary computing resources could not compute 2^{128} keys before the sun burned out.

Cryptanalysis is the systematic mathematical attempt to discover weaknesses in either cryptographic implementation or practice, and to use these weaknesses to decrypt messages. The idea of cryptanalytic attack is fascinating, and in some circumstances, the successes can be quite dramatic. In reality, more systems are breached through human failure. Several of the more common forms of cryptanalytic attack are described below.

A **ciphertext-only attack** is based purely on intercepted ciphertext. It is the most difficult because there are so few clues as to what has been encrypted, forcing the cryptanalyst to search for patterns within the ciphertext. The more ciphertext available for any given encryption key, the easier it is to find patterns facilitating cryptanalysis. To reduce the amount of ciphertext associated with specific keys, virtual private networks, which can exchange huge amounts of encrypted data, automatically change them regularly.

A **known plaintext attack** is based on knowledge of at least part of the plaintext message, which can furnish valuable clues in cracking the entire text. It is not unusual for an interceptor to be aware of some of a message's plaintext. The name of the sender or recipient, geographical names, standard headers and footers, and other context-dependent text may be assumed as part of many documents and messages. A **reasonable guess attack** is similar to a known plaintext attack.

Password cracking tools are a common example of reasonable guess techniques. Called a **dictionary attack**, they attempt to crack passwords by starting with words known to be common passwords. If that fails, they then attempt to try all of the words in a dictionary list supplied by the operator. Such automated attacks are effectively brute-force attacks on a limited subset of keys. L0phtcrack not only uses a dictionary attack but also exploits weaknesses in NT's password hashing implementation that were discovered through cryptanalysis, making it a highly efficient password guesser.

COMPROMISE OF KEY

In practice, most encryption failures are due to human weakness and sloppy practices. The human password used to access a security domain or crypto subsystem is often poorly chosen and easily guessable. Password guessing can be extraordinarily fruitful, but stolen passwords are also quite common. Theft may be accomplished through physical examination of an office; passwords are often stuck on the monitor or glued underneath the keyboard. Social engineering is the use of deception to elicit private information. A typical social engineering password theft involves the attacker phoning the victim, explaining that they are with the help desk and need the user's password to resolve a problem.

Passwords can also be stolen using a sniffer to capture them as they traverse the network. Older services, such as FTP and Telnet, send the user password and login across the network in plaintext. Automated attack tools that sniff Telnet and FTP passwords and save them are commonly found in compromised UNIX systems. NT passwords are hashed in a very weak fashion, and the L0phtcrack utility includes a function to collect crackable password hashes by sniffing login sessions.

If a system is compromised, there might be several passwords available on it for theft. Windows 95 and 98 systems use a very weak encryption method that is very easy to crack. Software that requires a password entry often leaves copies of the unencrypted passwords on the hard drive, either in temporary files or swap space, making them easy to find. Private keys are typically 1024 bits long, but are protected online by encrypting them with a human password that is usually easy to remember (or else it would be written down). Recent studies have shown that identifying an encrypted public key on a hard drive is relatively straightforward, because it has such a high level of entropy (high randomness) relative to other data. If a system with a private key on the hard drive is compromised, it must be assumed that a motivated attacker will be able to locate and decrypt the private key and would then be able to masquerade as the key holder.

Even if a workstation is not physically compromised, remote attacks are not difficult. If the system has an exploitable remote control application on it — either a legitimate one like PCAnywhere that might be poorly configured, or an overtly hostile backdoor application such as NetBus — then an attacker can capture the legitimate user's password. Once the attacker has a user's password, if the private key is accessible through software, the remote control attacker can create a message or document and sign it with the victim's private key, effectively appropriating their identity. A virus named Caligula is designed to steal encrypted keys from infected systems using PGP Mail, copying them back out to a site on the Internet where potential attackers can download them and attempt to decrypt them. Because software-based keys are so easily compromised, they should only be used for relatively low assurance applications, like routine business and personal mail. Legal and commercial transactions should be signed with a key stored in a protected and removable hardware device.

CREATING RELIABLE ENCRYPTION IS DIFFICULT

As should be clear from the wide variety of attacks, creating and using encryption is fraught with danger. Unsuccessful cryptosystems typically fail in one of four areas.

Algorithm Development

Modern encryption techniques derive their strength from having so many possible keys that a brute-force attack is infeasible. The key space (i.e., the potential population of keys) is a function of key size. A robust encryption implementation should not be breakable by exploiting weaknesses in the algorithm, which is the complex formula of transpositions and substitutions used to perform the data transformation. When designing algorithms, cryptologists assume that not only the algorithm, but even the encryption engine source code will be known to anyone attempting to break encrypted data. This represents a radical change from pre-computing era cryptography, in which the mechanics of the encryption engines were jealously guarded. The success of the American forces over the Japanese at the battle of Midway was facilitated by knowledge of the Japanese naval deployment, allowing American aviators to attack the larger Japanese fleet at the maximum possible range. American cryptanalysts had reverse-engineered the Japanese encryption machines, making it feasible to break their enciphered transmissions.

Suitable encryption algorithms are notoriously difficult to create. Even the best developers have had spectacular failures. History has shown that the creation and thorough testing of new encryption algorithms requires a team of highly qualified cryptologists. Experience has also shown that proprietary encryption techniques, which are common on PCs, usually fail

when subjected to rigorous attack. At best, only a few thousand specialists can claim suitable expertise in the esoteric world of cryptology. Meanwhile, millions of Internet users need access to strong cryptologic technology. The only safe choice for the layperson is to choose encryption products based on standard algorithms that are widely recognized by experts as being appropriately resistant to cryptanalytic attack.

> Even worse, it doesn't do any good to have a bunch of random people examine the code; the only way to tell good cryptography from bad cryptography is to have it examined by experts. Analyzing cryptography is hard, and there are very few people in the world who can do it competently. Before an algorithm can really be considered secure, it needs to be examined by many experts over the course of years.
>
> — Bruce Schneier, CRYPTO-GRAM, September 15, 1999

Implementation

Creation of a robust encryption algorithm is just the first challenge in the development of an encryption product. The algorithm must be carefully implemented in hardware or software so that it performs correctly and is practical to use. Even when an algorithm is correctly implemented, the overall system security posture may be weakened by some other factor. Key generation is a weak spot. If an attacker discovers a pattern in key generation, it effectively reduces the total population of possible keys and greatly reduces the strength of the implementation. A recent example was the failure of one of the original implementations of Netscape's SSL, which used a predictable time-based technique for random number generation. When subjected to statistical analysis, few man-made devices can provide sufficiently random output.

Deployment

Lack of necessary encryption, due to a delayed or cancelled program, can cause as much damage as the use of a flawed system. For a cryptosystem to be successful, the chosen products must be provided to everyone who will be expected to use them.

Operation

Experience constantly demonstrates that people are the biggest concern, not technology. A successful encryption project requires clearly stated goals, which are formally referred to as policies, and clearly delineated user instructions or procedures. Highly sophisticated encryption projects, such as public key infrastructures, require detailed operational documents such as practice statements. Using encryption to meet organizational goals requires constant administrative vigilance over infrastructure and use of keys. Encryption technology will fail without user cooperation.

> It turns out that the threat model commonly used by cryptosystem designers was wrong: most frauds were not caused by cryptanalysis or other technical attacks, but by implementation errors and management failures. This suggests that a paradigm shift is overdue in computer security; we look at some of the alternatives, and see some signs that this shift may be getting under way.
>
> — Ross Anderson, "Why Cryptosystems Fail"
> A United Kingdom-based study of failure modes
> of encryption in banking applications

TYPES OF ENCRYPTION

Two basic types of encryption are used: symmetric and asymmetric. The traditional form is symmetric, in which a single secret key is used for both encryption and decryption. Asymmetric encryption uses a pair of mathematically related keys, commonly called the private key and the public key. It is not computationally feasible to derive the matching private key using the encrypted data and the public key. Public key encryption is the enabler for a wide variety of electronic transactions and is crucial for the implementation of E-commerce.

Symmetric Encryption

A symmetric algorithm is one that uses the same key for encryption and decryption. Symmetric algorithms are fast and relatively simple to implement. The primary disadvantage in using secret key encryption is actually keeping the key secret. In multi-party transactions, some secure mechanism is necessary in order to share or distribute the key so that only the appropriate parties have access to the secret key. Exhibit 17-1 lists the most common symmetric algorithms, all of which have proven acceptably resistant to cyryptanalytic attack in their current implementation.

Asymmetric (Public Key) Encryption

The concept of public key encryption represented a revolution in the applicability of computer-based security in 1976 when it was introduced in a journal article by Whitfield Diffie and Martin Hellman. This was quickly followed in 1978 with a practical implementation. Developed by Ron Rivest, Adi Shamir, and Len Adelman, their "RSA" scheme is still the only public key encryption algorithm in widespread use. Public key encryption uses one simple but powerful concept to enable an extraordinary variety of online trusted transactions: one party can verify that a second party holds a specific secret without having to know what that secret is. It is impossible to imagine what E-commerce would be without it. Many transaction types would be impossible or hopelessly difficult. Unlike secret key encryption, asymmetric encryption uses two keys, either one of which can be used to decrypt ciphertext encrypted with the corresponding key. In practice, one

Exhibit 17-1. Common symmetric algorithms.

Algorithm	Developer	Key Size (bits)	Characteristics
DES	IBM under U.S. government contract	56	Adopted as a U.S. federal standard in 1976 Most widely implemented encryption algorithm Increasing concern over resistance to brute-force attack
3DES	3 sequential applications of DES	112	Slow
IDEA	Developed in Switzerland by Xuejia Lai and James Massey	128	Published in 1991 Widely used in PGP Must be licensed for commercial use
Blowfish	Bruce Schneier	Up to 448	Published in 1993 Fast, compact, and flexible

key is referred to as the secret key, and is carefully protected by its owner, while the matching public key can be freely distributed. Data encrypted with the public key can only be decrypted by the holder of the private key. Likewise, if ciphertext can be successfully decrypted using the public key, it is proof that whoever encrypted the message used a specific private key.

Like symmetric algorithms, public key encryption implementations do not rely on the obscurity of their algorithm, but use key lengths that are so long that a brute-force attack is impossible. Asymmetric encryption keys are based on prime numbers, which limits the population of numbers that can be used as keys. To make it impractical for an attacker to derive the private key, even when ciphertext and the public key are known, RSA key length of 1024 bits has become the standard practice. This is roughly equivalent to an 80-bit symmetric key in resistance to a brute-force attack. Not only does public key encryption require a much longer key than symmetric encryption, it is also exponentially slower. It is so time-consuming that it is usually not practical to encrypt an entire data object. Instead, a one-time session key is randomly generated and used to encrypt the object with an efficient secret key algorithm. The asymmetric algorithm and the recipient's public key are then used to encrypt the session key so that it can only be decrypted with the recipient's private key.

Only a few asymmetric algorithms are in common use today. The Whitfield-Diffie algorithm is used for secure key exchange, and the digital signature algorithm (DSA) is used only for digital signature. Only two algorithms are currently used for encryption; RSA is by far the most widespread. *Elliptic curve* is a newer form of public key encryption that uses smaller key lengths

and is less computationally intensive. This makes it ideal for smart cards, which have relatively slow processors. Because it is newer, and based on unproven mathematical concepts, elliptic curve encryption is sometimes considered riskier than RSA encryption. It is important to understand that RSA encryption, while apparently remaining unbroken in 20 years of use, has not been mathematically proven secure either. It is based on the intuitive belief that the process of factoring very large numbers cannot be simplified. Minor improvements in factoring, such as a technique called Quadratic Sieve, encouraged the increase in typical RSA key length from 512 to 1024 bits. A mathematical or technological breakthrough in factoring is unlikely, but it would quickly obsolete systems based on RSA technology.

Additional Cryptography Types

A hash algorithm is a one-way cryptographic function. When applied to a data object, it outputs a fixed-size output, often called a message digest. It is conceptually similar to a checksum, but is much more difficult to corrupt. To provide a tamper-proof fingerprint of a data object, it must be impossible to derive any information about the original object from its message digest. If the original data is altered and the hash algorithm is reapplied, the new message digest must provide no clue as to what the change in the data was. In other words, even a 1-bit change in the data must result in a dramatically different hash value.

The most widely used secure hash algorithm is MD5, published by Ron Rivest in 1992. Some authorities expect it to be obsolete shortly, suggesting that developments in computational speed might already have rendered it inadequate. SHA-1 outputs a longer hash than MD5. The U.S. federal government is promulgating SHA-1, and it is becoming increasingly common in commercial applications.

Steganography is the practice of hiding data. This differs from encryption, which makes intercepted data unusable, but does not attempt to conceal its presence. While most forms of security do not protect data by hiding it, the mere fact that someone has taken the trouble to encrypt it indicates that the data is probably valuable. The owner may prefer not to advertise the fact that sensitive data even exists. Traditional forms of steganography include invisible ink and microdots; cryptographic steganography uses data transformation routines to hide information within some other digital data.

Multimedia objects, such as bitmaps and audio or video files, are the traditional hiding places, although a steganographic file system was recently announced. Multimedia files are relatively large compared to textual documents, and quite a few bits can be changed without making differences that are discernable to human senses. As an example, this chapter can easily be secreted within a true color photograph suitable as a 1024×768 screen

background. The desired storage object must be both large enough and complex enough to allow the data object to be hidden within it without making detectable changes to the appearance or sound of the object. This is an implementation issue; a secure steganography utility must evaluate the suitability of a storage object before allowing the transformation to occur. An object containing data secreted within it will have a different hash value than the original, but current implementations of steganography do not allow a direct human comparison between the original and modified file to show any detectible visual or audio changes.

While there are legitimate applications for cryptographic steganography, it is certainly a concern for corporations trying to control the outflow of proprietary data and for computer forensic investigators. Research is being conducted on techniques to identify the existence of steganographically hidden data, based on the hypotheses that specific steganography utilities leave characteristic patterns, or fingerprints. Most steganography utilities also provide an encryption option, so finding the hidden data does not mean that its confidentiality is immediately violated.

Digital watermarking is a communication security mechanism used to identify the source of a bitmap. It is most often used to protect intellectual property rights by allowing the owner of a multimedia object to prove that they were the original owners or creators of the object. Watermarking is similar to digital steganographic techniques in that the coded data is hidden in the least significant bits of some larger object.

CRYPTOGRAPHIC SERVICES

The most obvious use of encryption is to provide privacy, or confidentiality. Privacy can be applied in several contexts, depending on the specific protection needs of the data. Messages can be encrypted to provide protection from sniffing while being transmitted over a LAN or over the Internet. Encryption can also be used to protect the confidentiality of stored data that might be physically accessed by unauthorized parties.

Identification is accomplished in one of three ways, sometimes referred to as (1) something you know, (2) something you have, and (3) something you are. "Something you are" refers to biometric mechanisms, which are beyond the scope of this chapter, but the other two identification mechanisms are facilitated through encryption.

Passwords and passphrases are examples of "something you know" and they are normally protected cryptographically. The best practice is not to actually store phrases or passwords themselves, but to store their hash values. Each hash value has the same length, so they provide no clue as to the content or characteristics of the passphrase. The hash values can be further obfuscated through use of a *salt* value. On UNIX systems, for example,

the first two letters of the user name are used as salt as part of the DE-based hash routine. The result is that different logins that happen to have the same password will be associated with different hash values, which greatly complicates brute-force attacks.

Encryption keys can also serve as "something you have." This can be done with either symmetric or asymmetric algorithms. If two people share a secret key, and one of them encrypts a known value, they can recognize the other as being the only one who can provide the same encrypted result. In practice, public key-based identification systems scale much better, and are becoming increasingly common. Identification keys are stored on magnetic media or within a smart card. Usually, they are encrypted themselves and must be unlocked by the entry of a PIN, password, or passphrase by their owner before they can be accessed.

Integrity is provided by hashing a document to create a message digest. The integrity of the object can be verified by deriving the hash sum again, and comparing that value to the original. This simple application of a cryptographic hash algorithm is useful only when the hash value is protected from change. In a transaction in which an object is transmitted from one party to another, simply tacking a message digest onto the end of the object is insufficient — the recipient would have no assurance that the original document had not been modified and a matching new message digest included.

Authorship and Integrity assurance is provided cryptographically by digital signature. To digitally sign a document using RSA encryption, a hash value of the original document is calculated, which the signer then encrypts with their private key. The digital signature can be verified by decrypting the signature value with the signer's public key, and comparing the result to the hash value of the object. If the values do not match, the original object is no longer intact or the public and private keys do not match; in either case, the validation fails. Even if proof of authorship is not a requirement, digital signature is a practical integrity assurance mechanism because it protects the message digest by encrypting it with the signer's public key.

Digital signature provides a high level of assurance that a specific private key was used to sign a document, but it cannot provide any assurance that the purported owner of that private key actually performed the signature operation. The appropriate level of trust for any particular digitally signed object is provided through organizational procedures that are based on formal written policy. Any organization using digital signature must determine what level of systemic rigor is necessary when signing and verifying objects. Because the signature itself can only prove which key was used to sign the document, but not who actually wielded that key,

signature keys must be protected by authentication mechanisms. It is useless to verify a digital signature without having an acceptable level of trust that the public key actually belongs to the purported sender. Manual sharing of public keys is one way to be certain of their origin, but it is not practical for more than a few dozen correspondents. A third-party authentication service is the only practical way to support the trust needs of even a moderately sized organization, let alone the entire Internet.

A digital certificate provides third-party verification of the identity of a key holder. It takes the form of the keyholder's public key signed by the private key of a Certificate Authority (CA). This powerful concept makes it feasible to verify a digitally signed object sent by an unknown correspondent. The CA vouches for the identity of the certificate holder, and anyone with a trusted copy of the CA's public key can validate an individual's digital certificate. Once the authenticity of a certificate has been confirmed, the public key it contains can be used to validate the digital signature on an object from the certificate holder. E-mail applications that support digital signature and public key-based encryption typically include the sender's digital certificate whenever sending a message with a signed or encrypted object, making it easy for the sender to verify the message contents.

A Certificate Revocation List (CRL) is periodically published by some CAs to increase the level of trust associated with their certificates. Although digital certificates include an expiration date, it is often desirable to be able to cancel a certificate before it has expired. If a private key is compromised, a user account is cancelled, or a certificate holder is provided with a replacement certificate, then the original certificate is obsolete. Listing it on a CRL allows the CA to notify verifiers that the certificate issuer no longer considers it a valid certificate. CRLs increase the implementation and administration costs of a CA. Clients must access the revocation list over a network during verification, which increases the time required to validate a signature. Verification is impossible if the network or revocation list server is unavailable. Although their use can significantly increase the level of assurance provided by digital certificates, revocation implementations are rare.

It is not always practical to provide a digital certificate with every signed object, and high-assurance CAs need a CRL server. Directory service is a distributed database optimized for reading that can make both CRLs and certificates available on a wide area network (WAN) or the Internet. Most directory services are based on the X.500 standard and use the extensible format X.509 to store digital certificates.

Public key infrastructure (PKI) refers to the total system installed by an organization to support the distribution and use of digital certificates. A PKI encompasses both infrastructure and organizational process. Examples of organizational control mechanisms include certificate policies (CP)

specifying the exact levels of assurance necessary for specific types of information, and practice statements specifying the mechanisms and procedures that will provide it. A PKI can provide any arbitrary level of assurance, based on the rigor of the authentication mechanisms and practices. The more effort an organization uses to verify a certificate applicant's identity, and the more secure the mechanisms used to protect that certificate holder's private key, the more trust that can be placed in an object signed by that certificate holder. Higher trust exacts a higher cost, so PKIs typically define a hierarchy of certificate trust levels allowing an optimal tradeoff between efficiency and assurance.

Transactional Roles (Witnessing)

Commerce and law rely on a variety of transactions. Over thousands of years of civilization, conventions have been devised to provide the parties to these transactions with acceptable levels of assurance. The same transactions are desirable in the digital realm, but mechanisms requiring that a human mark a specific piece of paper need virtual replacements. Fortunately, trust can be increased using witnessing services that are enabled through public key encryption.

Nonrepudiation describes protection against the disavowal of a transaction by its initiator. Digital signature provides nonrepudiation by making it impossible for the owner of a private key to deny that his key was used to sign a specific object. The key holder can still claim that his private key had been stolen — the level of trust appropriate for any electronically signed document is dependent on the certificate policy. For example, a weak certificate policy may not require any authentication during the certificate request, making it relatively easy to steal someone's identity by obtaining a certificate in his or her name. A CP that requires a more robust vetting process before issuing a certificate, with private keys that can only be accessed through strong authentication mechanisms (such as biometrics), decreases the potential that a signer will repudiate a document.

A digital notary is a trusted third party that provides document signature authentication. The originator digitally signs a document and then registers it with a digital notary, who also signs it and then forwards it the final recipient. The recipient of a digitally notarized document verifies the signature of the notary, not the originator. A digital notary can follow much more stringent practices than is practical for an individual, and might also offer some form of monetary guarantee for documents that it notarizes. The slight inconvenience and cost of utilizing a digital notary allows a document originator to provide a higher level of assurance than they would be able to without using a trusted third party.

Timestamping is a transactional service that can be offered along with notarization, or it might be offered by an automated timestamp service

that is both lower cost and lower assurance than a full notarization service. A timestamp service is a trusted third party guaranteeing the accuracy of their timestamps. Witnessing is desirable for digital object time verification because computer clocks are untrustworthy and easily manipulated through both hardware and software. Like a digitally notarized document, a timestamped document is digitally signed with the private key of the verification service and then forwarded to the recipient. Applications suitable for timestamping include employee or consultant digital time cards, performance data for service level agreements, telemetry or test data registration, and proposal submission.

Key exchange is a process in which two parties agree on a secret key known only to themselves. Some form of key exchange protocol is required in many forms of secure network connectivity, such as the initiation of a virtual private network connection. The Whitfield-Diffie algorithm is an especially convenient key exchange technique because it allows two parties to securely agree on a secret session key without having any prior relationship or need for a certificate infrastructure.

Key Recovery

Clashes between civil libertarians and the U.S. federal government have generated negative publicity on the subject of key escrow. Security practitioners should not let this political debate distract them from understanding that organizations have a legitimate need to protect their own data. Just as employers routinely keep extra keys to employee offices, desks, and safes, they are justified in their concern over digital keys. Very few organizations can afford to allow a single individual to exercise sole control over valuable corporate information. Key recovery is never required for data transmission keys because lost data can be immediately resent. However, if someone with the only key to stored encrypted data resigns is unavailable, or loses his key, then the organization loses that data permanently. Key recovery describes the ability to decrypt data without the permission or assistance of its owner. Organizations that use encryption to protect the privacy of stored data must understand the risk of key loss; and if key loss is unacceptable, their encryption policy should mandate key recovery or backup capabilities.

PUTTING IT INTO PRACTICE

Exhibit 17-2 provides an example process using both secret and public key cryptography to digitally sign and encrypt a message. Contemporary public key-based systems, such as e-mail and file encryption products, are complex hybrids using symmetric algorithms for privacy and the RSA public key algorithm to securely exchange keys. A hashing algorithm and the RSA public key algorithm provide digital signature. Starting in this case

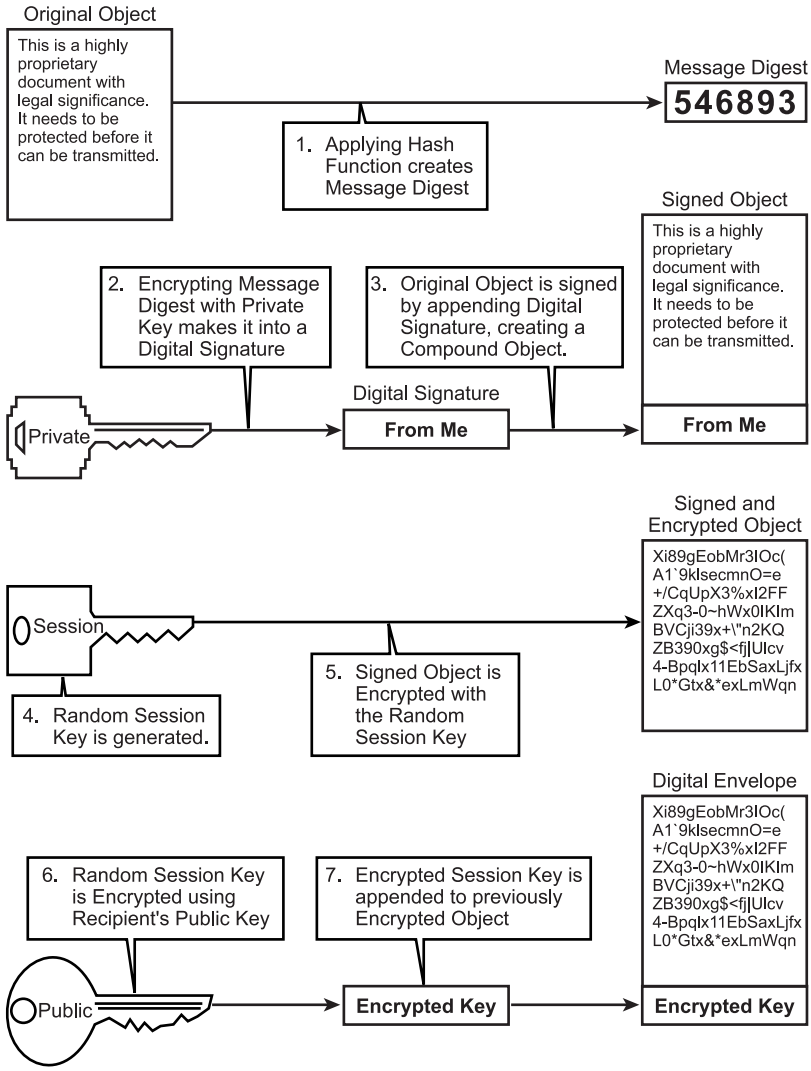


Exhibit 17-2. Using public key encryption to protect an object.

with a text file, the first step is to apply a cryptographic hash function to create a message digest (1). To protect this message digest from manipulation, and to turn it into a digital signature, it is encrypted with the private key of the signer (2). The signer's public key is highly sensitive stored information, and it must be protected with some sort of authentication mechanism. At a minimum, the key owner must enter a password to access the key. (While this is the most common protective mechanism for private

keys, it is by far the weakest link in this entire multi-step process). After creating the digital signature, it is concatenated onto the original file, creating a signed object (3). In practice, a compound object like this normally has additional fields, such as information on the hash algorithm used, and possibly the digital certificate of the signer. At this point, the original object has been turned into a digitally signed object, suitable for transmission. This is effectively an unsealed digital envelope. If privacy is required, the original object and the digital signature must be encrypted.

Although it would be possible to encrypt the signed object using a public key algorithm, this would be extremely slow, and it would limit the potential distribution of the encrypted object. To increase efficiency and provide destination flexibility, the object is encrypted using a secret key algorithm. First, a one-time random session key is generated (4). The signed object is encrypted with a symmetric algorithm, using this session key as the secret key (5). Then the session key, which is relatively small, is encrypted using the public key of the recipient (6). In systems based on RSA algorithms, users normally have two pairs of public and private keys: one pair is used for digital signature and the other is used for session key encryption. If the object is going to be sent to multiple recipients, copies of the session key will be encrypted with each of their public keys. If the message is meant to be stored, one of the keys could be associated with a key recovery system or it might be encrypted for the supervisor of the signer. All of the encrypted copies of the session key are appended onto the encrypted object, effectively creating a sealed digital envelope. Again, in practice, this compound object is in a standardized format that includes information on the encryption algorithms, and mapping information between encrypted session keys and some sort of user identifier is included. A digital envelope standard from RSA called PKCS #7 is widely used. It can serve as either an unsealed (signed but not encrypted) or sealed (signed and encrypted) digital envelope.

The processes are reversed by the recipient. As shown in Exhibit 17-3, the encrypted session key must be decrypted using the recipient's private key (2) (which should be stored in encrypted form and accessed with a password). The decrypted session key is used to decrypt the data portion of the digital envelope (3), providing a new object consisting of the original data and a digital signature. Verification of the digital signature is a three-step process. First, a message digest is derived by performing a hash function on the original data object (5). Then the digital signature is decrypted using the signer's public key. If the decrypted digital signature does not have the same value as the computed hash value, then either the original object has been changed, or the public key used to verify the signature does not match the private key used to sign the object.

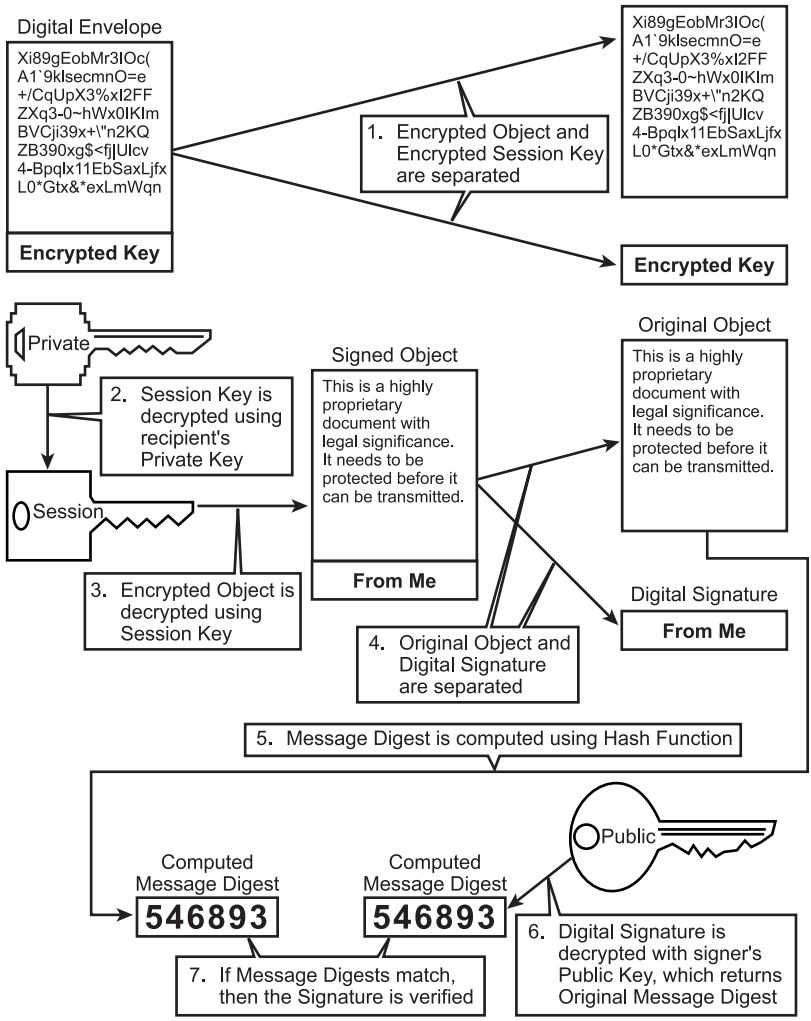


Exhibit 17-3. Decrypting and verifying a signed object.

CONCLUSION

This chapter is just a brief introduction to a fascinating and complex subject. Familiarity with encryption concepts has become mandatory for those seeking a career involving Internet technology (see Exhibit 17-4). Many online and printed resources are available to provide more detailed information on encryption technology and application. The "Annotated Bibliography" contains suggestions for readers interested in a more in-depth approach to this subject.

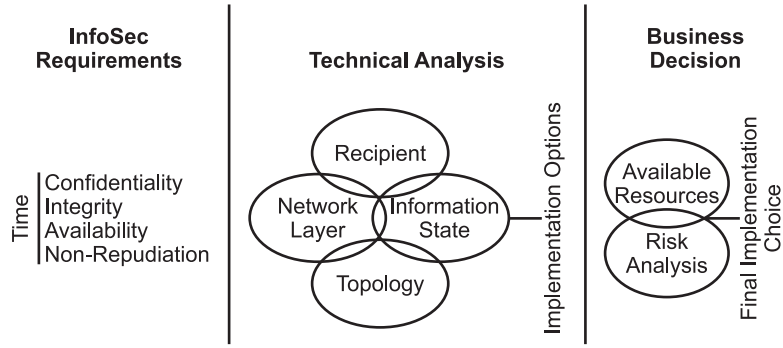


Exhibit 17-4. Encryption concepts.

Annotated Bibliography

Printed References

1. Dan and Lim, Eds., *Cryptography's Role in Securing the Information Society,* National Research Council. Although somewhat dated, this contains useful information not found in other sources on how specific industries apply encryption.
2. Diffie, W. and Hellman, M., New Directions in Cryptography, *IEEE Transactions on Information Theory,* November 1976. This is the first article on public key encryption to appear in an unclassified publication.
3. Kahn, David, *The Codebreakers; The Comprehensive History of Secret Communication from Ancient Times to the Internet.* Kahn's original 1969 tome was recently updated. It is an exhaustive reference that is considered the most authoritative historical guide to cryptography.
4. Marks, Leo, *Between Silk and Cyanide: A Codemaker's War 1941–1945.* A personal biography of a WWII British cryptographer. An entertaining book that should make crystal clear the importance of following proper procedures and maintaining good hygiene. The electronic cryptographic infrastructure can be broken down just like the manual infrastructure used in WWII for military and intelligence traffic. It dramatizes the dangers in making decisions about the use of encryption without properly understanding how it can be broken down.
5. Schneier, Bruce, *Applied Cryptography,* 2nd edition. Everyone involved in encryption in any fashion must have a copy of this comprehensive text. Schneier is brilliant not only in making complex mathematics accessible to the layperson, but he also has a tremendous grasp on the trust issues and the human social conventions replicated cryptographically in the virtual world.
6. Smith, Richard, *Internet Cryptography.* A basic text intended for non-programmers.
7. Stallings, William, *Cryptography and Network Security: Principles and Practice.* A comprehensive college textbook.

Online References

1. Anderson, Ross, Why Cryptosystems Fail, http://www.cl.cam.ac.uk/users/rja14/wcf.html.
2. Schneier, Bruce, Security Pitfalls in Cryptography, http://www.counterpane.com/pitfalls.html.
3. Schneier, Bruce, Why Cryptography Is Harder Than It Looks, http://www.counterpane.com/ whycrypto.html.

4. PKCS documentation, http://www.rsa.com/rsalabs/pubs/PKCS/.
5. Ellis, J., The Story of Non-decret Encryption, CESG Report, 1987, http://www.cesg.gov.uk/ellisint.htm.
6. Johnson, N., Steganography, http://patriot.net/~johnson/html/neil/stegdoc/stegdoc.html, 1997.
7. M. Blaze, W. Diffie, R. Rivest, B. Schneier, T. Shimomura, E. Thompson, and M. Weiner, Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security, http://www.counterpane.com/keylength.html.

115

Principles and Applications of Cryptographic Key Management

William Hugh Murray, CISSP

Introduction

The least appreciated of the (five) inventions that characterize modern cryptography is automated key management. This powerful mechanism enables us to overcome the lack of rigor and discipline that leads to the inevitable compromise of crypto systems. By permitting us to change keys frequently and safely, it overcomes the fundamental limitations of the algorithms that we use. It enables us to compensate for such human limitations as the inability to remember or transcribe long random numbers.

This chapter attempts to tell the information security professional the minimum that he needs to know about key management. It must presume that the professional already understands modern cryptography. This chapter defines key management, enumerates its fundamental principles, and describes its use. It will make recommendations on the key choices that confront the user and manager.

Context

First a little context. Cryptography is the use of secret codes to hide data and to authenticate its origin and content. Although public codes could be used to authenticate content, secret codes are necessary to authenticate origin. This use of cryptography emerged only in the latter half of the 20th century and has been surprising to all but a few.

Of all security mechanisms, cryptography is the one most suited to open and hostile environments, environments where control is otherwise limited, environments like the modern, open, flat, broadcast, packet-switched, heterogeneous networks.

It is broadly applicable. In the presence of cheap computing power, its uses are limited only by our imaginations. Given that most of the power of our computers goes unused, we could, if we wished, use secret codes by default, converting into public codes only for use. Indeed, modern distributed computing systems and applications would be impossible without it.

It is portable; the necessary software to encode or decode the information can be distributed at or near the time of use in the same package and channel. Within minor limits, it is composable; we can put together different functions and algorithms without losing any strength. One can put together mechanisms in such a way as to emulate any environmental or media-based control that we have ever had.

Not only is cryptography effective, it is efficient. That is to say, it is usually the cheapest way to achieve a specified degree of protection. The cost of cryptography is low. Not only is it low in absolute terms, it is low

in terms of the security value it delivers. It is low compared to the value of the data it protects. It is low compared to the alternative ways of achieving the same degree of security by such alternative means as custody, supervision, or automated access control.

Its low cost is the result in part of the low cost of the modern computer, and it is falling with the cost of that computing. The cost of a single cryptographic operation today is one ten thousandth of what it was as recently as 20 years ago and can be expected to continue to fall.

Another way of looking at it is that its relative strength is rising when cost is held constant; the cost to the user is falling relative to the cost to the attacker. As we will see, automated key management is one mechanism that permits us to trade the increasing power of computing for increased security.

Modern cryptography is arbitrarily strong; that is, it is as strong as we need it to be. If one knows what data he wishes to protect, for how long, and from whom, then it is possible to use modern cryptography to achieve the desired protection. There are limitations; if one wanted to encrypt tens of gigabytes of data for centuries, it is hard to know how to achieve that. However, this is a theoretical rather than a practical problem. In practice, there are no such applications or problems.

Cryptography is significantly stronger than other security mechanisms. Almost never will cryptography be the weak link in the security chain. However, in practice its strength is limited by the other links in the chain, for example, key management. As it is not efficient to make one link in a chain significantly stronger than another, so it is not necessary for cryptography to be more than a few hundred times stronger than the other mechanisms on which the safety of the data depends.

The cryptography component of a security solution is robust and resilient, not likely to break. While history suggests that advances in technology may lower the cost of attack against a particular cryptographic mechanism, it also suggests that the cost does not drop suddenly or precipitously. It is very unlikely to collapse. Given the relative effectiveness and efficiency of cryptography relative to other security measures, changes in the cost of attack against cryptography are unlikely to put security at risk. The impact is obvious, and there is sufficient opportunity to compensate.

Changes in technology reduce the cost to both the user of cryptography and the attacker. Because the attacker enjoys economies of scale, historically, advances such as the computer have favored him first and the user second. However, that probably changed forever when both the scale and the cost of the computer fell to within the discretion of an individual. Further advances in technology are likely to favor the cryptographer.

As we will see, as the cost of attack falls, the user will spend a little money to compensate. However, it is in the nature of cryptography that as his costs rise linearly, the costs to the attacker rise exponentially. For example, the cost of attack against the Data Encryption Standard (DES) has fallen to roughly a million MIPS years. Although this is still adequate for most applications, some users have begun to use Triple DES-112. This may quadruple their cost but double the cost of a brute-force attack.

One way of looking at cryptography is that it changes the problem of maintaining the secrecy of the message to one of maintaining the secrecy of the keys. How we do that is called *key management.*

Key Management Defined

Key management can be defined as the generation, recording, transcription, distribution, installation, storage, change, disposition, and control of cryptographic keys. History suggests that key management is very important. It suggests that each of these steps is an opportunity to compromise the cryptographic system. Further, it suggests that attacks against keys and key management are far more likely and efficient than attacks against algorithms.

Key management is not obvious or intuitive. It is very easy to get it wrong. For example, students found that a recent release of Netscape's SSL (Secure Sockets Layer) implementation chose the key from a recognizable subspace of the total keyspace. Although the total space would have been prohibitively expensive to exhaust, the subspace was quite easy. Key management provides all kinds of opportunities for these kinds of errors.

As a consequence, key management must be rigorous and disciplined. History tells us that this is extremely difficult to accomplish. The most productive cryptanalytic attacks in history, such as ULTRA, have exploited poor key management. Modern automated key management attempts to use the computer to provide the necessary rigor and discipline. Moreover, it can be used to compensate for the inherent limitations in the algorithms we use.

Key Management Functions

This section addresses the functions that define key management in more detail. It identifies the issues around each of these functions that the manager needs to be aware of.

Key Generation

Key generation is the selection of the number that is going to be used to tailor an encryption mechanism to a particular use. The use may be a sender and receiver pair, a domain, an application, a device, or a data object. The key must be chosen in such a way that it is not predictable and that knowledge of it is not leaked in the process.

It is necessary but not sufficient that the key be randomly chosen. In an early implementation of the SSL protocol, Netscape chose the key in such a manner that it would, perforce, be chosen from a small subset of the total set of possible keys. Thus, an otherwise secure algorithm and secure protocol was weakened to the strength of a toy. Students, having examined how the keys were chosen, found that they could find the keys chosen by examining a very small set of possible keys.

In addition to choosing keys randomly, it is also important that the chosen key not be disclosed at the time of the selection. Although a key may be stored securely after its generation, it may be vulnerable to disclosure at the time of its generation when it may appear in the clear. Alternatively, information that is used in the generation of the key may be recorded at the time it is collected, thus making the key more predictable than might otherwise be concluded by the size of the keyspace. For example, some key-generation routines, requiring random numbers, ask the user for noisy data. They may ask the user to run his hands over the key board. While knowledge of the result of this action might not enable an attacker to predict the key, it might dramatically reduce the set of keys that the attacker must search.

Distribution

Key distribution is the process of getting a key from the point of its generation to the point of its intended use. This problem is more difficult in symmetric key algorithms, where it is necessary to protect the key from disclosure in the process. This step must be performed in a channel separate from the one that the traffic moves in.

During the World War II, the Germans used a different key each day in their Enigma Machine but distributed the keys in advance. In at least one instance, the table of future keys, recorded on water-soluble paper, was captured from a sinking submarine.

Installation

Key installation is the process of getting the key into the storage of the device or process that is going to use it. Traditionally this step has involved some manual operations. Such operations might result in leakage of information about the key, error in its transcription, or it might be so cumbersome as to discourage its use.

The German Enigma Machine had two mechanisms for installing keys. One was a set of three (later four) rotors. The other was a set of plug wires. In one instance, the British succeeded in inserting a listening device in a code room in Vichy, France. The clicking of the rotors leaked information about the delta between key n and key n + 1.

The plugging of the wires was so cumbersome and error prone as to discourage its routine use. The British found that the assumption that today's plug setting was the same as yesterday's was usually valid.

Storage

Keys may be protected by the integrity of the storage mechanism itself. For example, the mechanism may be designed so that once the key is installed, it cannot be observed from outside the encryption machine itself. Indeed, some key-storage devices are designed to self-destruct when subjected to forces that might disclose the key or that are evidence that the key device is being tampered with.

Alternatively, the key may be stored in an encrypted form so that knowledge of the stored form does not disclose information about the behavior of the device under the key.

Visual observation of the Enigma Machine was sufficient to disclose the rotor setting and might disclose some information about the plug-board setting.

Change

Key change is ending the use of one key and beginning that of another. This is determined by convention or protocol. Traditionally, the time at which information about the key was most likely to leak was at key-change time. Thus, there was value to key stability. On the other hand, the longer the key is in use, the more traffic that is encrypted under it, the higher the probability that it will be discovered and the more traffic that will be compromised. Thus, there is value to changing the key.

The Germans changed the key every day but used it for all of the traffic in an entire theatre of operations for that day. Thus, the compromise of the key resulted in the compromise of a large quantity of traffic and a large amount of information or intelligence.

Control

Control of the key is the ability to exercise a directing or restraining influence over its content or use. For example, selecting which key from a set of keys is to be used for a particular application or party is part of key control. Ensuring that a key that is intended for encrypting keys cannot be used for data is part of key control. This is such a subtle concept that its existence is often overlooked. On the other hand, it is usually essential to the proper functioning of a system.

The inventors of modern key management believe that this concept of key control and the mechanism that they invented for it, which they call the *control vector*, is one of their biggest contributions.

Disposal

Keys must be disposed of in such a way as to resist disclosure. This was more of a problem when keys were used for a long time and when they were distributed in persistent storage media than it is now. For example, Enigma keys for submarines were distributed in books with the keys for the future. In at least one instance, such a book was captured.

Modern Key Management

Modern key management was invented by an IBM team in the 1970s.[1] It was described in the *IBM Systems Journal*[2] at the same time as the publication of the Data Encryption Standard (DES). However, although the DES has inspired great notice, comment, and research, key management has not gotten the recognition it deserves. While commentators were complaining about the length of the DES key, IBM was treating it as a solved problem; they always knew how they would compensate for fixed key length and believed that they had told the world.

Modern key management is fully automated; manual steps are neither required nor permitted. Users do not select, communicate, or transcribe keys. Not only would such steps require the user to know the key and permit him to disclose it, accidentally or deliberately, they would also be very prone to error.

Modern key management permits and facilitates frequent key changes. For example, most modern systems provide that a different key will be used for each object, e.g., file, session, message, or transaction, to be encrypted. These keys are generated at the time of the application of encryption to the object and specifically for that object. Its life is no longer than the life of the object itself. The most obvious example is a session key. It is created at the time of the session, exchanged under a key-encrypting key, and automatically discarded at the end of the session. (Because of the persistence of TCP sessions, even this may result in too much traffic under a single key. The IBM proposal for secure-IP is to run two channels [TCP sessions], one for data and one for keys. The data key might change many times per session.)

One can compare the idea of changing the key for each object or method with the practices used during World War II. The Germans used the same key across all traffic for a service or theater for an entire day. Since the British were recording all traffic, the discovery of one key resulted in the recovery of a large amount of traffic.

Manual systems of key management were always in a difficult bind; the more frequently one changed the key, the greater the opportunity for error and compromise. On the other hand, the more data encrypted under a single key, the easier the attack against that key and the more data that might be compromised with that key. To change or not to change? How to decide?

Automating the system changes the balance. It permits frequent secure key changes that raise the cost of attack to the cryptanalyst. The more keys that are used for a given amount of data, the higher the cost of attack (the more keys to be found), and the lower the value of success (the less data for each key). As the number of keys increases, the cost of attack approaches infinity and the value of success approaches zero. The cost of changing keys increases the cost of encryption linearly, but it increases the cost of attack exponentially. All other things being equal, changing keys increases the effective key length of an algorithm.

Because many algorithms employ a fixed-length key, and one can almost always find the key in use by exhausting the finite set of keys, and because the falling cost and increasing speed of computers is always lowering the cost and elapsed time for such an attack, the finite length of the key might be a serious limitation on the effectiveness of the algorithm. In the world of the Internet, in which thousands of computers have been used simultaneously to find one key, it is at least conceivable that one might find the key within its useful life. Automatic key change compensates for this limit.

A recent challenge key[3] was found using more than 10,000 computers for months at the rate of billions of keys per second. The value of success was only $10,000. By definition, the life of a challenge key is equal to the duration of the attack. Automated key management enables us to keep the life of most keys to minutes to days rather than days to months.

However, modern key management has other advantages in addition to greater effective key length and shorter life. It can be used to ensure the involvement of multiple people in sensitive duties. For example, the Visa master key is stored in San Francisco inside a box called the BBN SafeKeyper. It was created inside that box and no one knows what it is. Beneficial use of the key requires possession of the box and its three physical keys. Because it is at least conceivable that the box could be destroyed, it has exported information about the key. Five trustees share that information in such a way that any three of them, using another SafeKeyper box, could reconstruct the key.

Key management can also be used to reduce the risk associated with a lost or damaged key. Although in a communication application there is no need to worry about lost keys, in a file encryption application, a lost key might be the equivalent of loss of the data. Key management can protect against that. For example, one of my colleagues has information about one of my keys that would enable him to recover it if anything should happen to me. In this case he can recover the key all by himself. Because a copy of a key halves its security, the implementation that we are using permits me to compensate by specifying how many people must participate in recovering the key.

Key management may be a stand-alone computer application or it can be integrated into another application. IBM markets a product that banks can use to manage keys across banks and applications. The Netscape Navigator and Lotus Notes have key management built in.

Key management must provide for the protection of keys in storage and during exchange. Smart cards may be used to accomplish this. For example, if one wishes to exchange a key with another, one can put it in a smart card and mail it. It would be useless to anyone who took it from the mail.

Principles of Key Management

A number of principles guide the use and implementation of key management. These are necessary, but may not be sufficient, for safe implementation. That is, even implementations that adhere to these principles may be weak, but all implementations that do not adhere to these principles are weak.

First, *Key* management must be fully automated. There may not be any manual operations. This principle is necessary both for discipline and for the secrecy of the keys.

Second, *No* key may ever appear in the clear outside a cryptographic device. This principle is necessary for the secrecy of the keys. It also resists known plain-text attacks against keys.

Keys must be randomly chosen from the entire keyspace. If there is any pattern to the manner in which keys are chosen, this pattern can be exploited by an attacker to reduce his work. If the keys are drawn in such a way that all possible keys do not have an equal opportunity to be drawn, then the work of the attacker is reduced. For example, if keys are chosen so as to correspond to natural language words, then only keys that have such a correspondence, rather than the whole space, must be searched.

Key-encrypting keys must be separate from data keys. Keys that are used to encrypt other keys must not be used to encrypt data, and vice versa. Nothing that has ever appeared in the clear may be encrypted under a key-encrypting key. If keys are truly randomly chosen and are never used to encrypt anything that has appeared in the clear, then they are not vulnerable to an exhaustive or brute-force attack. In order to understand this, it is necessary to understand how a brute-force attack works.

In a brute-force attack, one tries keys one after another until one finds the key in use. The problem that the attacker has is that he must be able to recognize the correct key when he tries it. There are two ways to do this, corresponding clear- and cipher-text attacks, and cipher-text-only attacks. In the former, the attacker keeps trying keys on the cipher text until he finds the one that produces the expected clear text.

At a minimum, the attacker must have a copy of the algorithm and a copy of the cryptogram. In modern cryptography, the algorithm is assumed to be public. Encrypted keys will sometimes appear in the environment, and encrypted data, cipher text, is expected to appear there.

For the first attack, the attacker must have corresponding clear and cipher text. In historical cryptography, when keys were used widely or for an extended period of time, the attacker could get corresponding clear and cipher text by duping the cryptographer into encrypting a message that he already knew. In modern cryptography, where a key is used only once and then discarded, this is much more difficult to do.

In the cipher-text-only attack, the attacker tries a key on the cipher text until it produces recognizable clear text. Clear text may be recognized because it is not random. In the recent RSA DES Key Challenge, the correct clear-text message could be recognized because the message was known to begin with the words, "The correct message is...." However, even if this had not been the case, the message would have been recognizable because it was encoded in ASCII.

To resist cipher-text-only attacks, good practice requires that all such patterns as format, e.g., file or e-mail message, language (e.g., English), alphabet (e.g., Roman), and public code (e.g., ASCII or EBCDIC) in the clear text object must be disguised before the object is encrypted.

Note that neither of these attacks will work on a key-encrypting key if the principles of key management are adhered to. The first one cannot be made to work because the crypto engine cannot be duped into encrypting a known value under a key-encrypting key. The only thing that it will encrypt under a key-encrypting key is a random value which it produced inside itself. The cipher-text-only attack cannot be made to work because there is no information in the clear text key that will allow the attacker to recognize it. That is, the clear text key is, by definition, totally random, without recognizable pattern, information, or entropy.

Keys with a long life must be sparsely used. There are keys, such as the Visa master key mentioned earlier, whose application is such that a very long life is desirable. As we have already noted, the more a key is used, the more likely is a successful attack and the greater the consequences of its compromise. Therefore, we compensate by using this key very sparsely and only for a few other keys. There is so little data encrypted under this key and that data is so narrowly held that a successful attack is unlikely. Because only this limited number of keys is encrypted under this key, changing it is not prohibitively expensive.

Asymmetric Key Cryptography

In traditional and conventional cryptography, the key used for encrypting and the one used for decrypting have the same value; that is to say that the relationship between them is one of symmetry or equality. In 1976, Whitfield Diffie and Martin Hellman pointed out that although the relationship between these two numbers must be fixed, it need not be equality. Other relationships could serve. Thus was born the idea of asymmetric key cryptography.

In this kind of cryptography the key has two parts; the parts are mathematically related to each other in such a way that what is encrypted with one part can only be decrypted by the other. The value of one of the keys does not necessarily imply the other; one cannot easily calculate one from the other. However, one of the keys, plus a message encrypted under it, does imply the other key. From a message and one part of the key, it is mathematically possible to calculate the other but it is not computationally feasible to do so.

Only one part, called the *private key*, need be kept secret. The other part, the *public key*, is published to the world. Anyone can use the public key to encrypt a message that can only be decrypted and read by the owner of the private key. Conversely, anyone can read a message encrypted with the private key, but only the person with beneficial use of that key could have encrypted it.

Note that if A and B share a symmetric key, then either knows that a message encrypted under that key originated with the other. Because a change in as little as one bit of the message will cause it to decode to garbage, the receiver of a good message knows that the message has not been tampered with. However, because each party has beneficial use of the key and could have created the cryptogram, they cannot demonstrate that it originated with the other. In asymmetric key cryptography only the possessor of the private key can have created the cryptogram. Any message that will decrypt with the public key is therefore known to all to have originated with the person who published it. This mechanism provides us with a digital signature capability that is independent of medium and far more resistant to forgery than marks on paper.

Although key management can be accomplished using only symmetric key cryptography, it requires secret key exchange, a closed population, some prearrangement, and it benefits greatly from trusted hardware. Asymmetric key cryptography enables us to do key management without secret key exchange, in an open population, with a minimum of prearrangement. It reduces the need for trusted hardware for key distribution though it is still desirable for key storage and transcription.

However, when otherwise compared to symmetric key cryptography, asymmetric key cryptography comes up short. Exhibit 115.1 compares a symmetric key algorithm, DES, to an asymmetric key algorithm, RSA. Exhibit 115.1 shows that the asymmetric key algorithm requires much longer keys to achieve the same computational resistance to attack (i.e., to achieve the same security). It takes much longer to generate a key. It is much slower in operation, and its cost goes up faster than the size of the object to be encrypted.

However, for keys that are to be used for a long period of time, the time required to generate a key is not an issue. For short objects to be encrypted, performance is not an issue. Therefore, asymmetric key cryptography is well suited to key management applications, and in practice its use is limited to that role. Most products use symmetric key cryptography to encrypt files, messages, sessions, and other objects, but use asymmetric key cryptography to exchange and protect keys.

Hybrid Cryptography

If one reads the popular literature, he is likely to be gulled into believing that he has to make a choice between symmetric and asymmetric key cryptography. In fact and in practice, this is not the case. In practice we use a hybrid of the two that enables us to enjoy the benefits of each. In this style of use, a symmetric key algorithm is used to hide the object, while an asymmetric key mechanism is used to manage the keys of this symmetric algorithm.

The symmetric key algorithm is well suited for hiding the data object. It is fast and secure, even with a short key. Because keys are easily chosen, they can be changed for each object. The asymmetric key algorithm would not be suitable for this purpose because it is slow and requires a long key that is expensive to choose.

On the other hand, the asymmetric algorithm is well suited to managing keys. Because symmetric keys are short, one need not worry about the speed of encrypting them. Because key management keys are relatively stable, one need not worry about the cost of finding them.

Exhibit 115.2 illustrates a simple implementation of hybrid cryptography. A randomly selected 56-bit key is used to encrypt a message using the DES algorithm. This key is then encrypted using Jane's public key. The encrypted message along with its encrypted key are now broadcast. Everyone can see these; however, their meaning is hidden from all but Jane. Jane uses her private key to recover the message key and the message key to recover the message.

EXHIBIT 115.1 DES versus RSA

Characteristic	DES	RSA
Relative speed	Fast	Slow
Functions used	Transportation, Substitution	Multiplication
Key length	56 bits	400–800 bits
Least-cost attack	Exhaustion	Factoring
Cost of attack	Centuries	Centuries
Time to generate a key	Microseconds	Tens of seconds
Key type	Symmetric	Asymmetric

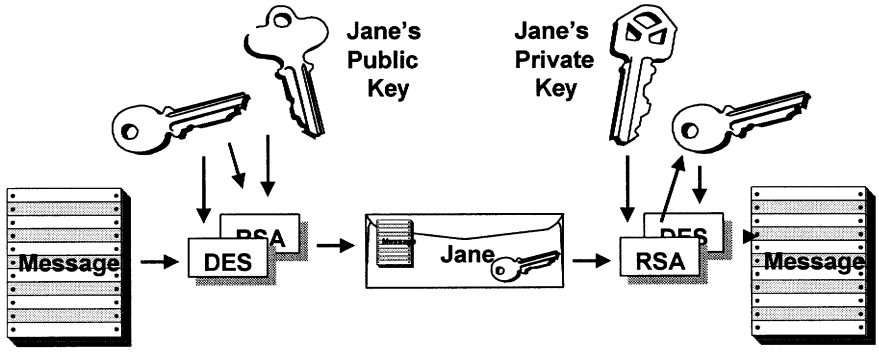


EXHIBIT 115.2 Hybrid cryptography.

Public Key Certificates

As we have noted, by definition, there is no need to keep public keys secret. However, it is necessary to ensure that one is using the correct public key. One must obtain the key in such a way as to preserve confidence that it is the right key. Also, as already noted, the best way to do that is to obtain the key directly from the party. However, in practice we will get public keys at the time of use and in the most expeditious manner.

As we do with traditional signatures, we may rely on a trusted third party to vouch for the association between a particular key and a particular person or institution. For example, the state issues credentials that vouch for the bind between a photo, name and address, and a signature. This may be a driver's license or a passport. Similar credentials, called *public key certificates*, will be issued for public keys by the same kinds of institutions that issue credentials today: employers, banks, credit card companies, telephone companies, state departments of motor vehicles, health insurers, and nation-states.

A public key certificate is a credential that vouches for the bind or join between a key pair and the identity of the owner of the key. Most certificates will vouch for the bind between the key pair and a legal person. It contains the identifiers of the key pair owner and the public half of the key pair. It is signed by the private key of the issuing authority and can be checked using the authority's public key. In addition to the identifiers of the owner and the key, it may also contain the start and end dates of its validity, and its intended purpose, use, and limitations. Like other credentials, it is revocable at the discretion of the issuer and used or not at the discretion of the key owner. Like other credentials, it is likely to be one of several and, for some purposes, may be used in combination with others.

Credential issuers or certification authorities (CAs) are legal persons trusted by others to vouch for the bind, join, or association between a public key and another person or entity. The CA may be a principal, such as the management of a company, a bank, or a credit card company. It may be the secretary of a "club" or other voluntary association, such as a bank clearing house association. It may be a government agency or designee, such as the post office or a notary public. It may be an independent third party operating as a fiduciary and for a profit.

The principal requirement for a certification authority is that it must be trusted by those who will use the certificate and for the purpose for which the certificate is intended. The necessary trust may come from its role, independence, affinity, reputation, contract, or other legal obligation.

Use of Certificates for Managing Keys

In one-to-one relationships, one knows that one is using the correct public key because one obtains it directly and personally from one's correspondent. However, for large populations and most applications, this is not feasible. In most such cases, it is desirable to obtain the key automatically and late, that is, at or near the time of use.

In a typical messaging application, one might look up one's correspondent in a public directory, using his name as a search argument. As a function, one would get an e-mail address, a public key, and a certificate that bound the key to the name and address.

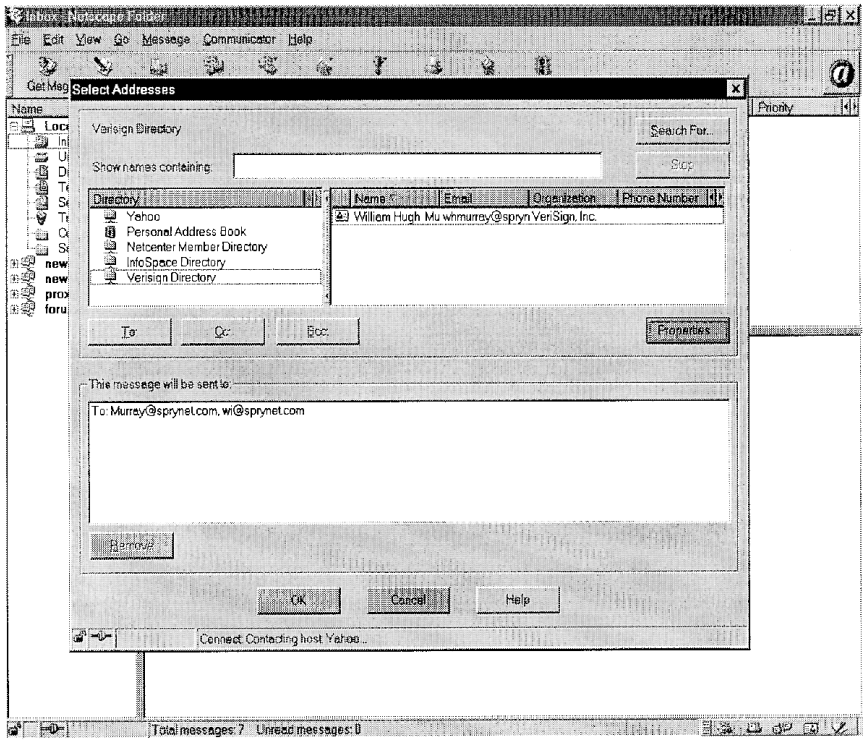


EXHIBIT 115.3

Exhibit 115.3 illustrates looking up the address whmurray@sprynet.com in the public directory operated by VeriSign, Inc. In addition to the address, the directory returns a public key that goes with that name and address. It also returns a certificate for that key. As a rule, the user will never see nor care about the key or the certificate. They will be handled automatically by the application. However, if one clicked on the <properties> button, one would see the certificate shown in Exhibit 115.4.

If one now clicks <Encrypt> on the message options, the message will now be encrypted using this key. If one signs a message using a private key, the corresponding public key and its certificate will automatically be attached to the message. Other applications work in a similar manner. Tool kits can be purchased to incorporate these functions into enterprise-developed applications.

Implementations

To illustrate the power, use, and limitations of modern key management, this section discusses a number of implementations or products. Because the purpose of this discussion is to make points about key management, it will not provide a complete discussion of any of the products. The products are used only for their value as examples of key management. The order of presentation is chosen for illustrative purposes rather than to imply importance.

Kerberos Key Distribution Center

The Kerberos key distribution center (KDC) is a trusted server to permit any two processes that it knows about to obtain trusted copies of a key-session key. Kerberos shares a secret with every process or principal in the population. When A wants to talk to B, it requests a key from the KDC. The KDC takes a random number and encrypts it under the secret it shares with B, appends a second copy of the key, and encrypts the result under the secret that it shares with A. It broadcasts the result into the network addressed to A.

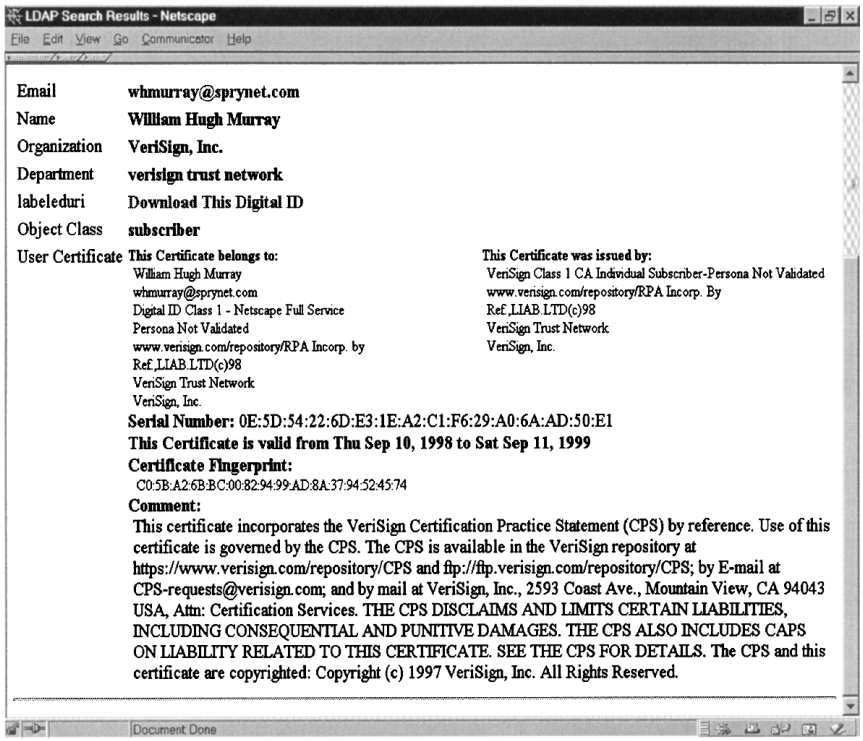


EXHIBIT 115.4

A uses the secret it shares with the KDC to recover its copy of the key and B's copy (encrypted under the secret that B shares with the KDC). It broadcasts B's copy into the network addressed to B. Although everyone in the network can see the messages, only A and B can use them. B uses its secret to recover its copy of the key. Now A and B share a key that they can use to talk securely to each other.

This process requires that the KDC be fully trusted to vouch for the identity of A and B, but not to divulge the secrets or the key to other processes or even to use it itself. If the KDC is compromised, all of the secrets will have to be changed, i.e., the principals must all be reenrolled. These limitations could be reduced if, instead of keeping a copy of the secret shared with the principals, the KDC kept only its public key. Then whatever other remedies might be necessary if the KDC were compromised, there would be no secrets to change.

PGP

PGP stands for Phil's "Pretty Good Privacy." Phil Zimmerman, its author, has received honors and awards for this product, not so much because of its elegant design and implementation, as for the fact that it brought the power of encryption to the masses. It is the encryption mechanism of choice for confidential communication among individuals.

PGP is implemented exclusively in software. It is available in source code, and implementations are available for all popular personal computers. It is available for download from servers all over the world and is free for private use. It is used to encrypt files for protection on the storage of the local system and to encrypt messages to be sent across a distance.

It uses a block cipher, IDEA, with a 128-bit key to encrypt files or messages. It automatically generates a new block-cipher key for each file or message to be encrypted. It uses an asymmetric key algorithm, Rivest–Shamir–Adelman (RSA), to safely exchange this key with the intended recipient by encrypting it using the recipient's public key. Only the intended recipient, by definition the person who has beneficial use of the mathematically corresponding private key, can recover the symmetric key and read the message.

Because the principles of key management require that this key not be stored in the clear, it is stored encrypted under the block cipher. The key for this step is not stored but is generated every time it is needed

by compressing to 128 bits an arbitrarily long passphrase chosen by the owner of the private key. Thus, beneficial use of the private key requires both a copy of the encrypted key and knowledge of the passphrase.

Of course, while PGP does not require secret exchange of a key in advance, it does require that the public key be securely acquired. That is, it must be obtained in a manner that preserves confidence that it is the key of the intended recipient. The easiest way to do this is to obtain it directly, hand-to-hand, from that recipient. However, PGP has features to preserve confidence while passing the public key via e-mail, public servers, or third parties.

Note that if the passphrase is forgotten, the legitimate owner will have lost beneficial use of the private key and all message or file keys that were hidden using the public key. For communication encryption the remedy is simply to generate a new key-pair, publish the new public key, and have the originator resend the message using the new key. However, for file encryption, access to the file is lost. As we will see, commercial products use key management to provide a remedy for this contingency.

PGP stores keys in files called *key rings*. These files associate user identifiers with their keys. It provides a number of mechanisms for ensuring that one is using the correct and intended public key for a correspondent. One of these is called the *key fingerprint*. This is a relatively short hash of the key that can be exchanged out of channel and used to check the identity of a key. Alice sends a key to Bob. On receiving the key, Bob computes the fingerprint and checks it with Alice. Note that although fingerprints are information about the public key, they contain even less information about the private key than does the public key itself. Therefore, the fingerprint need not be kept secret.

PGP also provides a record of the level of trust that was attributed to the source of the key when it was obtained. This information is available whenever the key is used. Of course, the existence of this mechanism suggests that all sources are not trusted equally nor equally trustworthy. In practice, entire key rings are often exchanged and then passed on to others. In the process, the provenance of and confidence in a key may be obscured; indeed, the confidence in a key is often no better than hearsay. The documentation of PGP suggests that the potential for duping someone into using the wrong key is one of the greatest limitations to the security of PGP.

ViaCrypt PGP, Business Edition

ViaCrypt PGP, Business Edition, is licensed for business or commercial use and includes emergency key recovery features to address some of the limitations of PGP noted above. Instead of encrypting the private key under a key generated on-the-fly from the passphrase, it introduces another level of key. This key will be used to encrypt the private key and will itself be hidden using the passphrases of the "owners" of the private key. This may be the sole user or it may be an employee and manager representing his employer. In the latter case, the employee is protected from management abuse of the private key by the fact that he has possession of it, and management only has possession of a copy of the key used to hide it. However, both the employee and management are protected from the consequences of loss of a single passphrase.

RSA SecurePC

RSA SecurePC is an add-in to the Windows file manager that is used for file encryption. It has features that extend the ideas in PGP BE and illustrate some other uses of key management. It encrypts specified files, directories, or folders, on command, that is, by marking and clicking; or by default, by marking a file or directory and indicating that everything in it is always to be encrypted. Marking the root of a drive would result in all files on the drive, except executables, always being stored in encrypted form.

The object of encryption is always the individual file rather than the drive or the directory. When a file is initially encrypted, the system generates a 64-bit block-cipher key to be used to encrypt the file. This file key is then encrypted using the public key of the system and is stored with the file.

The private key for the system is stored encrypted using a two-level key system and passphrase as in PGP BE. In order for a user to read an encrypted file, he must have the file key in the clear. To get that, he must have the private key in the clear. Therefore, when he opens a file, the system looks to see if the private key is in the clear in its memory. If not, then the user is prompted for his passphrase so that the private key can be recovered. At the time of this prompt, the user is asked to confirm or set the length of time that the private key is to be kept in the clear in system memory. The default is five minutes. Setting it to zero means that the user will be prompted for a second use. The maximum is 8 hours. The lower the user sets the time that the

key may remain in memory, the more secure it is; the higher he sets it, the less often he will be prompted for the passphrase.

RSA SecurePC also implements emergency key-recovery features. These features go beyond those described above in that management may specify that multiple parties must be involved in recovering the private key. These features not only permit management to specify the minimum number of parties that must be involved but also permits them to specify a larger set from which the minimum may be chosen. Multiparty emergency key recovery provides both the user and management with greater protection against abuse.

BBN SafeKeyper

BBN SafeKeyper is a book-size hardware box for generating and protecting private keys. It generates a private-key/public-key pair. The private key cannot be removed from the box. Beneficial use of the key requires possession of the box and its three physical keys. SafeKeyper is intended for the root key for institutions.

The box has a unique identity and a public key belonging to BBN. After it generates its key pair, it encrypts its public key and its identity under the public key of BBN and broadcasts it into the network addressed to BBN. When BBN recovers the key, it uses its own private key to create a "certificate" for the SafeKeyper that vouches for the bind between the public key and the identity of the person or institution to whom BBN sold the box.

Although the SafeKeyper box is very robust, it is still conceivable that it could be destroyed and its key lost. Therefore, it implements emergency key recovery. Although it is not possible to make an arbitrary copy of its key, it will publish information about its key sufficient to enable another SafeKeyper box to recreate it. For example, information about the Visa master key is held by five people. Any three of them acting in concert can reproduce this key.

Secure Sockets Layer (SSL)

SSL is both an API and a protocol intended for end-to-end encryption in client–server applications across an arbitrary network. The protocol was developed by Netscape, and the Navigator browser is its reference implementation. It uses public key certificates to authenticate the server to the client and, optionally, the client to the server.

When the browser connects to the secure server, the server sends its public key along with a certificate issued by a public certification authority. The browser automatically uses the issuer's public key to check the certificate, and manifests this by setting the URL to that of the server. It then uses the server's public key to negotiate a session key to be used for the session. It manifests this by setting a solid key icon in the lower left-hand corner of the screen.

Optionally, the client can send its public key and a certificate for that key issued by the management of the server or a certification authority trusted by the management of the server.

Recommendations for Key Management

To ensure rigor and discipline, automate all encryption, particularly including key management; hide all encryption from users.

To resist disclosure or arbitrary copies of a key, prefer trusted hardware for key storage. Prefer evaluated (FIPS-140)[4] hardware, dedicated single-application-only machines (such as those from Atalla, BBN, Cylink, and Zergo), smart cards, PCMCIA cards, laptops, diskettes, and trusted desktops, in that order. As a general rule, one should discourage the use of multi-user systems for key storage except for keys that are the property of the system owner or manager (e.g., payroll manager key).

Prefer one copy of a key; avoid strategies that require multiple copies of a key. Every copy of a key increases the potential for disclosure. For example, rather than replicating a single key across multiple servers, use different keys on each server with a certificate from a common source.

Change keys for each file, message, session, or other object.

Prefer one key per use or application rather than sharing a key across multiple uses. The more data that is encrypted under a single key, the greater the potential for successful cryptanalysis and the more damaging the consequences. With modern key management, keys are cheap.

To reduce the consequences of forgotten passphrases, use emergency key recovery for file encryption applications. Do not use emergency key recovery for communication encryption; change the key and resend the message.

Employ multiparty control for emergency key recovery; this reduces the potential for abuse, improves accountability, and increases trust all around. Consider requiring that the parties come from different levels of management and from different business or staff functions.

To ensure that keys are randomly selected from the entire keyspace, prefer closed and trusted processes for key generation. Avoid any manual operations in key selection.

Prefer encryption and key management that are integrated into the application. The easiest way to hide encryption from the user and to avoid errors is to integrate the encryption into the application.

Similarly, prefer applications with integrated encryption and key management. No serious business applications can be done in the modern network environment without encryption. Integrated encryption is a mark of good application design.

Finally, buy key management code from competent laboratories; do not attempt to write your own.

Notes:

1. Dr. Dorothy Denning has told me privately that she believes that automated key management was invented by the National Security Agency prior to IBM. Whether or not that is true is classified. In the absence of contemporaneous publication, it is unknowable. However, even if it is true, their invention did not ever make a difference; as far as we know, it never appeared in a system or an implementation. The IBM team actually implemented theirs, and it has made a huge difference. I remember being told by a member of the IBM team about the reaction of NSA to IBM's discussion of key management. He indicated that the reaction was as to a novel concept.
2. R. Elander et al., *Systems Journal*, 1977; IBM pub G321-5066, *A Cryptographic Key Management Scheme.*
3. RSA $10,000 Challenge, http://www.frii.com/~rcv/deschall.htm.
4. Federal Information Processing Standard 140, http://csrc.ncsl.nist.gov/fips/fips1401.htm.

Getting Started with PKI

Harry DeMaio

In the recent history of information protection there has been an ongoing parade of technologies that loudly promises new and total solutions but frequently does not make it past the reviewing stand. In some cases, it breaks down completely at the start of the march. In others, it ends up turning down a side street. Is Public-Key Infrastructure (PKI) just another gaudy float behind more brass bands, or is there sufficient rationale to believe that this one might make it? There are some very good reasons for optimism in this case, but optimism has been high before.

To examine PKI, one needs to know more than just the design principles. Many a slick and sophisticated design has turned embarrassingly sour when implemented and put into application and operational contexts. There are also the questions of economics, market readiness, and operational/technological prerequisites, all of which can march a brilliant idea into a blind alley.

APPROACH AND PRELIMINARY DISCUSSION

We'll start with a short review of the changing requirements for security. Is there really a need, especially in networking, that didn't exist before for new security technologies and approaches?

- We'll (very) briefly describe encryption, public-key encryption and PKI.
- We'll see how well PKI satisfies today's needs from a design standpoint.
- We'll look at what's involved in actually making PKI a cost-effective reality.
- Finally, we'll ask whether PKI is an exceptional approach or just one of many alternatives worth looking at.

THE CHANGING WORLD OF NETWORKED SYSTEMS

First a few characteristics of yesterday's and today's network-based information processing need to be considered. If the differences can be summed up in a single phrase, it is "accelerated dynamics." The structure and components of most major networks are in a constant state of flux — as are the applications, transactions, and users that traverse its pathways. This has a profound influence on the nature, location, scope, and effectiveness of protective mechanisms.

Exhibit 22.1 illustrates some of the fundamental differences between traditional closed systems and open (often Internet-based) environments. These differences do much to explain the significant upsurge in interest in encryption technologies.

	LEGACY/CLOSED NETWORK	MODERN OPEN NETWORK
User Environments	Known and stable	Mobile/variable
End Points	Established	Dynamic/open
Network Structure	Established/known	Dynamic/open
Processing	Mainframe/internally distributed	Multisite/Multienterprise
Data Objects	Linked to defined process	Often independent

Exhibit 22.1. Open vs. Closed Networks

Clearly, each network is unique, and most display a mix of the above characteristics. But the trends toward openness and variability are clear. The implications for security can be profound. Security embedded in or "hard-wired" to the system and network infrastructure cannot carry the entire load in many of the more mobile and open environments, especially where dial-up is dominant. A more flexible mode that addresses the infrastructure, user, work station, environment, and data objects is required.

An example: Envision the following differences:

- A route salesperson who returns to the office work station in the evening to enter the day's orders (online batch)
- That same worker now entering on a laptop through a radio or dial-up phone link those same orders as they are being taken at the customer's premises (dial-up interactive)
- Third-party operators taking orders at an 800/888 call center
- Those same orders being entered by the customer on a Web site
- A combination of the above

The application is still the same: order entry. But the process is dramatically different, ranging from batch entry to Web-based electronic commerce.

In the first case, the infrastructure, environment, process, and user are known, stable, and can be well controlled. The classic access control facility or security server generally carries the load.

In the second (interactive dial-up) instance, the employee is still directly involved. However, now there is a portable device and its on-board functions and data, the dial-up connections, the network, the points of entry to the enterprise, and the enterprise processes to protect if the level of control that existed in the first instance is to be achieved.

The third instance involves a third party, and the network connection may be closed or open.

The fourth (Web-based) approach adds the unknowns created by the customer's direct involvement and linkage through the Internet to the company's system.

The fifth, hybrid scenario calls for significant compatibility adjustments on top of the other considerations. By the way, this scenario is not unlikely. A fallacious assumption in promoting Web-based services is that one can readily discontinue the other service modes. It seldom happens.

Consider the changes to identification, authentication, and authorization targets and processes in each instance. Consider monitoring and the audit trail. Then consider the integrity and availability issues. Finally, the potential for repudiation begins to rear its ugly head. The differences are real and significant.

THE EVOLVING BUSINESS NETWORK

Remember, too, that most network-based systems in operation today have evolved, or in many cases, accreted into their current state — adding infrastructures and applications on demand and using the technology available at the time. Darwin notwithstanding, some of the currently surviving networks are not necessarily the fittest. In most of the literature, networks are characterized as examples of a specific class — open–closed; intranet–extranet; LAN–WAN–Internet; protocol-X or protocol-Y. Although these necessary and valuable distinctions can be used to describe physical and logical infrastructures, remember that when viewed from the business processes they support supply chain, order entry, funds transfer, and patient record processing. Most "business process" networks are technological and structural hybrids.

The important point is that today security strategy and architecture decisions are being driven increasingly by specific business requirements, not just technology. This is especially true in the application of encryption-related techniques such as PKI. Looking again at the order entry example above, the application of consistent protective mechanisms for a hybrid order entry scenario will undoubtedly require compatibility and interoperability across platform and network types unless the entire system is rebuilt to one specification. This seldom happens unless the enterprise is embarking on a massive reengineering effort or deploying major application suites such as the SAP AG R/3 or PeopleSoft.

The Disintegration and Reintegration of Security Mechanisms

To be effective, a protective mechanism must appropriately bind with the object and the environment requiring protection. In open networks, the connection, structure, and relationship of the components are more loosely defined and variable. Therefore, the protective mechanisms must be more granular, focused, and more directly linked to the object or process to be protected than was the case with legacy systems. Formerly, protection processes operated primarily at a "subterranean plumbing" level, surfacing only in password and authorization administration and log-ons. Now the castle moat is being supplemented with "no-go" zones, personal bodyguards posted at strategic spots, food tasters, and trusted messengers.

Encryption mechanisms fit this direct, granular requirement often ideally, since they can protect individual files, data elements (including passwords), paths (tunneling and Virtual Private Networks) and manage access management requirements. (Identification and authentication through encryption is easier than authorization.) But saying that encryption is granular is not the same as saying that a PKI system is interoperable, portable, or scalable. In fact, it means that most encryption-related systems today are still piece parts, although some effective suites such as Entrust are in the market and several others, such as IBM SecureWay and RSA/SD Keon, are just entering.

This "disintegrated" and specialized approach to providing security function creates a frustrating problem for security professionals accustomed to integrated suites. Now the user becomes the integrator or must use a third-party integrator. The products may not integrate well or even be able to interface with one another. At the 1999 RSA Conference in San Jose, CA, the clarion call for security suites was loud and clear.

Encryption Defined

Encryption is a process for making intelligible information unintelligible through the application of sophisticated mathematical conversion

techniques. Obviously, to be useful the process must be reversible (decryption). The three major components of the encryption/decryption process are as follows:

1. *The information stream in clear or encrypted form.*
2. *The mathematical encryption process* — the algorithm. Interestingly, most commercial algorithms are publicly available and are not secret. What turns a public process into a uniquely secret one is the encryption key.
3. *The encryption key.* The encryption key is a data string that is mathematically combined with the information (clear or encrypted) by the algorithm to produce the opposite version of the data (encrypted or clear). Remember that all data on computers is represented in binary number coding. Binary numbers can be operated upon by the same arithmetic functions as those that apply to decimal numbers. So by combining complex arithmetic operations, the data and key are converted into an encrypted message form and decrypted using the same process and *same key — with one critical exception.*

Before explaining the exception, one more definition is required. The process that uses the *same key* to decrypt and encrypt is called *symmetric* cryptography. It has several advantages, including exceptional speed on computers. It has a serious drawback. In any population of communicating users (n), in order to have *individually unique* links between each pair of users, the total number of keys required is n (n + 1)/2. Try it with a small number and round up. If the population of users gets large enough, the number of individual keys required rapidly becomes unmanageable. This is one (but not the only) reason why symmetric cryptography has not had a great reception in the commercial marketplace in the last 20 years.

The salvation of cryptography for practical business use has been the application of a different class of cryptographic algorithms using *asymmetric* key pairs. The mathematics is complex and is not intuitively obvious, but the result is a *pair of linked keys* that must be used together. However, only one of the pair, the private key, must be kept secret by the key owner. The other half of the pair — the public key — can be openly distributed to anyone wishing to communicate with the key owner. A partial analogy is the cash depository in which all customers have the same key for depositing through a one-way door, but only the bank official has a key to open the door to extract the cash. This technique vastly reduces the number of keys required for the same population to communicate safely and uniquely.

ENTER PKI

If the public key is distributed openly, how do you know that it is valid and belongs with the appropriate secret key and the key owner? How do you manage the creation, use, and termination of these key pairs. That is the foundation of PKI. Several definitions follow:

> The comprehensive system required to provide public-key encryption and digital signature services is known as the *public-key infrastructure* (PKI). The purpose of a public-key infrastructure is to manage keys and certificates.
>
> *Entrust Inc.*

> A public-key infrastructure (PKI) consists of the programs, data formats, communications protocols, institutional policies, and procedures required for enterprise use of public-key cryptography.
>
> *Office of Information Technology, University of Minnesota*

> In its most simple form, a PKI is a system for publishing the public-key values used in public-key cryptography. There are two basic operations common to all PKIs:
>
> 1. Certification is the process of binding a public-key value to an individual organization or other entity, or even to some other piece of information such as a permission or credential.
> 2. Validation is the process of verifying that a certificate is still valid.
>
> How these two operations are implemented is the basic defining characteristic of all PKIs.
>
> *Marc Branchaud*

The Digital Certificate and Certificate Authorities

Obviously, from these definitions, a digital certificate is the focal point of the PKI process. What is it? In simplest terms, a digital certificate is a credential (in digital form) in which the public key of the individual is embedded along with other identifying data. That credential is encrypted (signed) by a trusted third party or certificate authority (CA) who has established the identity of the key owner (similar to but more rigorous than notarization). The "signing key" ties the certificate back to the CA and ultimately to the process that bound the certificate holder to his or her credentials and identity proof process.

By "signing" the certificate, the CA establishes and takes liability for the authenticity of the public key contained in the certificate and the fact that it is bound to the named user. Now total strangers who know or at least trust a common CA can use encryption not just to *conceal* the data but also to *authenticate* the other party. The *integrity* of the message is also ensured.

If you change it once encrypted, it will not decrypt. The message *cannot be repudiated* because it has been encrypted using the sender's certificate.

Who are CAs? Some large institutions are their own CAs, especially banks (private CAs). There are some independent services (public CAs) developing, and government, using the licensing model as a take off point, is moving into this environment. It may become a new security industry. In The Netherlands, KNB, the Dutch notary service, supplies digital certificates.

As you would expect, there has been a move by some security professionals to include more information in the certificate, making it a multipurpose "document." There is one major problem with this. Consider a driver's license, which is printed on special watermarked paper, includes the driver's picture and is encapsulated in plastic. If one wished to maintain more volatile information on it, such as current make of car(s), doctor's name and address, or next of kin, the person would have to get a new license for each change.

The same is true for a certificate. The user would have to go back to the CA for a new certificate each time he made a change. For a small and readily accessible population, this may be reasonable. However, PKI is usually justified based on large populations in open environments, often across multiple enterprises. The cost and administrative logjam can build up with the addition of authorization updates *embedded in the certificate*. This is why relatively changeable authorization data (permissions) are seldom embedded in the certificate but rather attached. There are several certificate structures that allow attachments or permissions that can be changed independently of the certificate itself.

To review, the certificate is the heart of the PKI system. A given population of users who wish to intercommunicate selects or is required to use a specific CA to obtain a certificate. That certificate contains the public-key half of an asymmetric key pair as well as other indicative information about the target individual. This individual is referred to as the "distinguished name" — implying that there can be no ambiguities in certificate-based identification — all Smiths must be separately distinguished by ancillary data.

Where are Certificates Used?

Certificates are used primarily in open environments in which closed network security techniques are inappropriate or insufficient for any or all of the following:

- Identification/authentication
- Confidentiality

- Message/transaction integrity
- Nonrepudiation

Not all PKI systems serve the same purposes or have the same protective priorities. This is important to understand when one is trying to justify a PKI system for a specific business environment.

How Does PKI Satisfy Those Business Environment Needs?

Market Expectation. As PKI becomes interoperable, scalable, and generally accepted, companies will begin to accept the wide use of encryption-related products. Large enterprises such as government, banks, and large commercial firms will develop trust models to easily incorporate PKI into everyday business use.

Current Reality. It is not that easy. Thus far, a significant number of PKI projects have been curtailed, revised, or temporarily shelved for reevaluation. The reasons most often given include the following:

- Immature technology
- Insufficient planning and preparation
- Underestimated scope
- Infrastructure and procedural costs
- Operational and technical incompatibilities
- Unclear cost-benefits

Apparent Conclusions about the Marketplace

PKI has compelling justifications for many enterprises, but there are usually more variables and pitfalls than anticipated. Broadside implementation, though sometimes necessary, has not been as cost-effective. Pilots and test beds are strongly recommended.

A properly designed CA/RA administrative function is always a critical success factor.

CERTIFICATES, CERTIFICATE AUTHORITIES (CA), AND REGISTRATION AUTHORITIES (RA)

How do they work and how are they related?

First look at the PKI certificate lifecycle. It is more involved than one may think. A digital certificate is a secure and trustworthy credential, and the process of its creation, use, and termination must be appropriately controlled.

Not all certificates are considered equally secure and trustworthy, and this is an active subject of standards and industry discussion. The strength

of the cryptography supporting the certificate is only one discriminating factor. The degree to which the certificate complies with a given standard, X.509, for example, is another criterion for trustworthiness. The standards cover a wide range of requirements, including content, configuration, and process. The following is hardly an exhaustive list, but it will provide some insight into some of the basic requirements of process.

- *Application* — How do the "certificate owners to be" apply for a certificate? To whom do they apply? What supporting materials are required? Must a face-to-face interview be conducted, or can a surrogate act for the subject? What sanctions are imposed for false, incomplete, or misleading statements? How is the application stored and protected, etc?
- *Validation* — How is the applicant's identity validated? By what instruments? By what agencies? For what period of time?
- *Issuance* — Assuming the application meets the criteria and the validation is successful, how is the certificate actually issued? Are third parties involved? Is the certificate sent to the individual or, in the case of an organization, some officer of that organization? How is issuance recorded? How are those records maintained and protected?
- *Acceptance* — How does the applicant indicate acceptance of the certificate? To whom? Is nonrepudiation of acceptance eliminated?
- *Use* — What are the conditions of use? Environments, systems, and applications?
- *Suspension or Revocation* — In the event of compromise or suspension, who must be notified? How? How soon after the event? How is the notice of revocation published?
- *Expiration and Renewal* — Terms, process, and authority?

Who and What Are the PKI Functional Entities That Must Be Considered?

Certification Authority (CAs)

- A person or institution who is trusted and can vouch for the authenticity of a public key
- May be a principal (e.g., management, bank, credit card issuer)
- May be a secretary of a "club" (e.g., bank clearing house)
- May be a government agency or designee (e.g., notary public, Department of Motor Vehicles, or post office)
- May be an independent third party operating for a profit (e.g., VeriSign)
- Makes a decision on evidence or knowledge after due diligence
- Records the decision by signing a certificate with its private key
- Authorizes issuance of certificate

Registration Authority (RA)

- Manages certificate life cycle, including Certificate Directory maintenance and Certificate Revocation List (s) maintenance and publication
- Thus can be a critical choke point in PKI process and a critical liability point, especially as it relates to CRLs
- An RA may or may not be CA

Other Entities

- *Other Trusted Third Parties* — These may be service organizations that manage the PKI process, brokers who procure certificates from certificate suppliers, or independent audit or consulting groups that evaluate the security of the PKI procedure
- *Individual Subscribers*
- *Business Subscribers* — In many large organizations, two additional constructs are used:
    1. *The Responsible Individual* (RI) — The enterprise certificate administrator
    2. *The Responsible Officer* (RO) — The enterprise officer who legally assures the company's commitment to the certificate. In many business instances, it is more important to know that this certificate is backed by a viable organization that will accept liability than to be able to fully identify the actual certificate holder. In a business transaction, the fact that a person can prove he or she is a partner in Deloitte & Touche LLP who is empowered to commit the firm usually means more than who that person is personally.

PKI policies and related statements include the following:

- Certificate policy
- Named set of rules governing certificate usage with common security requirements tailored to the operating environment within the enterprise
- Certificate practices statement (CPS)
- Detailed set of rules governing the Certificate Authority's operations
- Technical and administrative security controls
- Audit
- Key management
- Liability, financial stability, due diligence
- CA contractual requirements and documents
- Subscriber enrollment and termination processes

The Certificate Revocation List (CRL)

Of all the administrative and control mechanisms required by a PKI, the CRL function can be one of the more complex and subtle activities. The CRL is an important index of the overall trustworthiness of the specific PKI environment. Normally it is considered part of the RA's duties. Essentially the CRL is the instrument for checking the continued validity of the certificates for which the RA has responsibility. If a certificate is compromised, if the holder is no longer authorized to use the certificate or if there is a fault in the binding of the certificate to the holder, it must be revoked and taken out of circulation as rapidly as possible. All parties in the trust relationship must be informed. The CRL is usually a highly controlled online database (it may take any number of graphic forms) at which subscribers and administrators may determine the currency of a target partner's certificate. This process can vary dramatically by the following:

- *Timing/frequency of update*. Be careful of the language here. Many RAs claim a 24-hour update. That means the CRL is refreshed every 24 hours. It does not necessarily mean that the total cycle time for a particular revocation to be posted is 24 hours. It may be longer.
- *Push-pull*. This refers to the way in which subscribers can get updates from the CRL. Most CRLs require subscribers to pull the current update. A few private RAs (see below) employ a push methodology. There is a significant difference in cost and complexity and most important the line of demarcation between an RA's and subscriber's responsibility and liability. For lessened liability alone, most RAs prefer the pull mode.
- *Up link/down link*. There are two transmissions in the CRL process. The link from the revoking agent to the CRL and the distribution by the CRL to the subscribing universe. Much work has been exerted by RAs to increase the efficiency of the latter process, but because it depends on the revoking agency, the up link is often an Achilles' heel. Obviously, the overall time is a combination of both processes, plus file update time.
- *Cross domain*. The world of certificates may involve multiple domains and hierarchies. Each domain has a need to know the validity status of all certificates that are used within its bounds. In some large extranet environments, this may involve multiple and multilayer RA and CRL structures. Think this one through very carefully and be aware that the relationships may change each time the network encompasses a new environment.
- *Integrity*. One major way to undermine the trustworthiness of a PKI environment is to compromise the integrity of the CRL process. If the continued validity of the certificate population cannot be assured, the whole system is at risk.

- *Archiving*. How long should individual CRLs be kept and for what purposes?
- *Liabilities and commitments*. These should be clearly, unambiguously, and completely stated by all parties involved. In any case of message or transaction compromise traceable to faulty PKI process, the RA is invariably going to be involved. Make very sure you have a common understanding.

As you might expect, CAs and RAs come in a variety of types. Some of the more common include the following:

- *Full-service public CA* providing RA, certificate generation, issuance, and life-cycle management. Examples: VeriSign, U.S. Postal Service, TradeWave
- *Branded public CA* providing RA, certificate issuance and lifecycle management
- *Certificates generated by a trusted party*, e.g., VeriSign, GTE CyberTrust. Examples: IDMetrix/*GTE CyberTrust*, Sumitomo Bank/*VeriSign*
- *Private CAs* using CA turn-key system solutions internally. Examples: ScotiaBank (*Entrust*), Lexis–Nexis (*VeriSign On-Site*)
- *IBM Vault Registry*

There are also wide variations in trust structure models. This is driven by the business process and network architecture:

- Hierarchical trust (a classical hierarchy that may involve multiple levels and a large number of individual domains)
- VeriSign, Entrust
- X.509v3 certificates
- One-to-one binding of certificate and public key
- Web of Trust (a variation on peer relationships between domains)
- PGP
- Many-to-one binding of certificates and public key
- Constrained or Lattice of Trust structures
- Hybrid of hierarchical and Web models
- Xcert

There are several standards, guidelines, and practices that are applicable to PKI. This is both a blessing and a curse. The most common are listed below. Individual explanations can be found at several Web sites. Start at the following site, which has a very comprehensive set of PKI links — http://www.cert.dfn.de/eng/team/ske/pem-dok.html. This is one of the best PKI link sites available.

- X.500 Directory Services and X.509 Authentication
- Common Criteria (CC)
- ANSI X9 series

- Department of Defense Standards
- TCSEC, TSDM, SEI CMM
- IETF RFC — PKIX, PGP
- S/MIME, SSL, IPSEC
- SET
- ABA Guidelines
- Digital Signatures, Certification Practices
- FIPS Publications 46, 140-1, 180-1, 186

CA/RA Targets of Evaluation. To comprehensively assess the trustworthiness of the individual CA/RA and the associated processes, Deloitte & Touche has developed the following list of required evaluation targets:

- System level (in support of the CA/RA process and certificate usage if applicable)
- System components comprising an CA/RA environment
- Network devices
- Firewalls, routers, and switches
- Network servers
- IP addresses of all devices
- Client work stations
- Operating systems and application software
- Cryptographic devices
- Physical security, monitoring, and authentication capabilities
- Data object level (in support of the CA/RA process and certificate usage)
- Data structures used
- Critical information flows
- Configuration management of critical data items
- Cryptographic data
- Sensitive software applications
- Audit records
- Subscriber and certificate data
- CRLs
- Standards compliance where appropriate
- Application and operational level (repeated from above)
- Certificate policy
- Named set of rules governing certificate usage with common security requirements tailored to the operating environment within the enterprise
- Certificate practices statement (CPS)
- Detailed set of rules governing the CA operations
- Technical and administrative security controls
- Audit
- Key management

- Liability, financial stability, and due diligence
- CA contractual requirements and documents
- Subscriber enrollment and termination processes

How Well Does PKI Satisfy Today's Open Systems Security Needs?

In a nutshell, PKI is an evolving process. It has the fundamental strength, granularity, and flexibility required to support the security requirements outlined. In that respect, it is the best available alternative. But wholesale adoption of PKI as the best, final, and global solution for security needs is naïve and dangerous. It should be examined selectively by business process or application to determine whether there is sufficient "value-added" to justify the direct and indirect cost associated with deployment. As suites such as Entrust become more adaptive and rich interfaces to ERP systems such as the SAP R/3 become more commonplace, PKI will be the security technology of choice for major, high-value processes. It will never be the only game in town. Uncomfortable or disillusioning as it may be, the security world will be a multisolution environment for quite a while.

What Is Involved in Making PKI a Cost-Effective Reality?

The most common approach to launching PKI is a pilot environment. Get your feet wet. Map the due diligence and procedural requirements against the culture of the organization. Look at the volatility of the certificates that will be issued. What is their life expectancy and need for modification? Check the interface issues. What is the prospective growth curve for certificate use? How many entities will be involved? Is cross-certification necessary? Above all else, examine the authorization process requirements that must co-exist with PKI. PKI is not a full-function access-control process. Look into the standards and regulations that affect your industry. Are there export control issues associated with the PKI solution being deployed? Is interoperability a major requirement? If so, how flexible is the design of the solutions being considered?

CA PILOT CONSIDERATIONS

Type of Pilot

- *Proof of concept* — May be a test bed or an actual production environment
- *Operational* — A total but carefully scoped environment. Be sure to have a clear statement of expectations against which to measure functional and business results.
- *Interenterprise* — Avoid this as a start-up if possible. But sometimes it is the real justification for adopting PKI. If so, spend considerable time and effort getting a set of procedures and objectives agreed upon by

all of the partners involved. An objective third-party evaluation can be very helpful.
- Examine standards alternatives and requirements carefully — especially in a regulated industry.
- Check product and package compatibility, interoperability, and scalability *very carefully.*
- Develop alternative compatible product scenarios. At this stage of market maturity, a Plan B is essential. Obviously not all products are universally interchangeable. Develop a backup suite and do some preliminary testing on it.
- Investigate outsourced support as an initial step into the environment. Although a company's philosophy may dictate an internally developed solution, the first round may be better deployed using outside resources.
- What are the service levels explicitly or implicitly required?
- Start internally with a friendly environment. You need all the support you can get, especially from business process owners.
- Provide sufficient time and resources for procedural infrastructure development, including CA policy, CPS, and training
- Do not promise more than you can deliver.

Is PKI an Exceptional Approach or Just One of Many Alternatives Worth Looking At?

The answer depends largely on the security objectives of the organization. PKI is ideal (but potentially expensive) for extranets and environments in which more traditional identification and authentication are insufficient. Tempting as it may be, resist the urge to find the *single solution.* Most networked-based environments and the associated enterprises are too complex for one global solution. Examine the potential for SSL, SMIME, Kerberos, single sign-on, and VPNs. If you can make the technical, operational and cost-justification case for a single, PKI-based security approach, do so. PKI is a powerful structure, but it is not a religious icon. Leave yourself room for tailored multi-solution environments.

Harry DeMaio is president of Deloitte & Touche Security Services LLC, (DTS) a wholly owned subsidiary of Deloitte & Touche LLP, Deerfield, IL. In addition to his current assignment, he is a director in Deloitte & Touche (D&T) Enterprise Risk Services, delivering the D&T family of information security and continuity planning services to major clients globally.

Mitigating E-business Security Risks: Public Key Infrastructures in the Real World

Douglas C. Merrill
Eran Feigenbaum

MANY ORGANIZATIONS WANT TO GET INVOLVED WITH ELECTRONIC COMMERCE — OR ARE BEING FORCED TO BECOME AN E-BUSINESS BY THEIR COMPETITORS. The goal of this business decision is to realize bottom-line benefits from their information technology investment, such as more efficient vendor interactions and improved asset management. Such benefits have indeed been realized by organizations, but so have the associated risks, especially those related to information security. Managed risk is a good thing, but risk for its own sake, without proper management, can drive a company out of existence. More and more corporate management teams — even up to the board of directors level — are requiring evidence that security risks are being managed. In fact, when asked about the major stumbling blocks to widespread adoption of electronic business, upper management pointed to a lack of security as a primary source of hesitation.

An enterprisewide security architecture, including technology, appropriate security policies, and audit trails, can provide reasonable measures of risk management to address senior management concerns about E-business opportunities. One technology involved in enterprisewide security architectures is public key cryptography, often implemented in the form of a public key infrastructure (PKI). This chapter describes several hands-on

examples of PKI, including business cases and implementation plans. The authors attempt to present detail from a very practical, hands-on approach, based on their experience implementing PKI and providing large-scale systems integration services. Several shortcuts are taken in the technical discussions to simplify or clarify points, while endeavoring to ensure that these did not detract from the overall message.

Although this chapter focuses on a technology — PKI — it is important to realize that large implementations involve organizational transformation. Many nontechnical aspects are integral to the success of a PKI implementation, including organizational governance, performance monitoring, stakeholder management, and process adjustment. Failing to consider these aspects greatly increases the risk of project failure, although many of these factors are outside the domain of information security. In the authors' experience, successful PKI implementations involve not only information security personnel, but also business unit leaders and senior executives to ensure that these nontechnical aspects are handled appropriately.

NETWORK SECURITY: THE PROBLEM

As more and more data is made network-accessible, security mechanisms must be put in place to ensure only authorized users access the data. An organization does not want its competitor to read, for example, its internal pricing and availability information. Security breaches often arise through failures in authentication. Authentication is the process of identifying an individual so that one can determine the individual's access privileges. To start my car, I must authenticate myself to my car. When I start my car, I have to "prove" that I have the required token — the car key — before my car will start. Without a key, it is difficult to start my car. However, a car key is a poor authentication mechanism — it is not that difficult to get my car keys, and hence be me, at least as far as my car is concerned. In the everyday world, there are several stronger authentication mechanisms, such as presenting one's driver's license with a picture. People are asked to present their driver's licenses at events ranging from getting on a plane to withdrawing large amounts of money from a bank. Each of these uses involves comparing the image on the license to one's appearance. This strengthens the authentication process by requiring two-factor authentication — an attacker must not only have my license, but he must also resemble me. In the electronic world, it is far more difficult to get strong authentication: a computer cannot, in general, check to be sure a person looks like the picture on their driver's license. Typically, a user is required to memorize a username and password. These username and password pairs must be stored in operating system-specific files, application tables, and the user's head (or desk). Any individual sitting at a keyboard that can produce a user's password is assumed to be that user.

Traditional implementations of this model, although useful, have several significant problems. When a new user is added, a new username must be generated and a new password stored on each of the relevant machines. This can be a significant effort. Additionally, when a user leaves the company, that user's access must be terminated. If there are several machines and databases, ensuring that users are completely removed is not easy. The authors' experience with PricewaterhouseCoopers LLP (PricewaterhouseCoopers) assessing security of large corporations suggests that users are often not removed when they leave, creating significant security vulnerabilities.

Additionally, many studies have shown that users pick amazingly poor passwords, especially when constrained to use a maximum of eight characters, as is often the case in operating system authentication. For example, a recent assessment of a FORTUNE 50 company found that almost 10 percent of users chose a variant of the company's logo as their password. Such practices often make it possible for an intruder to simply guess a valid password for a user and hence obtain access to all the data that user could (legitimately) view or alter.

Finally, even if a strong password is selected, the mechanics of network transmission make the password vulnerable. When the user enters a username and password, there must be some mechanism for getting the identification materials to the server itself. This can be done in a variety of ways. The most common method is to simply transmit the username and password across the network. However, this information can be intercepted during transmission using commonly available tools called "sniffers." A sniffer reads data as it passes across a network — data such as one's username and password. After reading the information, the culprit could use the stolen credentials to masquerade as the legitimate user, attaining access to any information that the legitimate user could access. To prevent sniffing of passwords, many systems use cryptography to hide the plaintext of the password before sending it across the network. In this event, an attacker can still sniff the password off the network, but cannot simply read its plaintext; rather, the attacker sees only the encrypted version. The attacker is not entirely blocked, however. There are publicly available tools to attack the encrypted passwords using dictionary words or brute-force guessing to get the plaintext password from the encrypted password. These attacks exploit the use of unchanging passwords and functions. Although this requires substantial effort, many demonstrated examples of accounts being compromised through this sort of attack are known.

These concerns — lack of updates after users leave, poor password selection, and the capability to sniff passwords off networks — make reliance on username and password pairs for remote identification to business-critical information unsatisfactory.

WHY CRYPTOGRAPHY IS USEFUL

Cryptography (from the Greek for "secret writing") provides techniques for ensuring data integrity and confidentiality during transport and for lessening the threat associated with traditional passwords. These techniques include codes, ciphers, and steganography. This chapter only considers ciphers; for information on other types of cryptography, one could read Bruce Schneier's *Applied Cryptography* or David Kahn's *The Codebreakers*. Ciphers use mathematics to transform plaintext into "ciphertext." It is very difficult to transform ciphertext back into plaintext without a special key. The key is distributed only to select individuals. Anyone who does not have the key cannot read or alter the data without significant effort. Hence, authentication becomes the question, "does this person have the expected key?" Additionally, the property that only a certain person (or set of people) has access to a key implies that only those individuals could have done anything to an object encrypted with that key. This so-called "nonrepudiation" provides assurance about an action that was performed, such as that the action was performed by John Doe, or at a certain time, etc.

There are two types of ciphers. The first method is called secret key cryptography. In secret key cryptography, a secret — a password — must be shared between sender and recipient in order for the recipient to decrypt the object. The best-known secret key cryptographic algorithm is the Data Encryption Standard (DES). Other methods include IDEA, RC4, Blowfish, and CAST. Secret key cryptography methods are, in general, very fast, because they use fairly simple mathematics, such as binary additions, bit shifts, and table lookups.

However, transporting the secret key from sender to recipient — or recipients — is very difficult. If four people must all have access to a particular encrypted object, the creator of the object must get the same key to each person in a safe manner. This is difficult enough. However, an even more difficult situation occurs when each of the four people must be able to communicate with each of the others without the remaining individuals being able to read the communication (see Exhibit 16-1). In this event, each pair of people must share a secret key known only to those two individuals. To accomplish this with four people requires that six keys be created and distributed. With ten people, the situation requires 45 key exchanges (see Exhibit 16-2). Also, if keys were compromised — such as would happen when a previously authorized person leaves the company — all the keys known to the departing employee must be changed. Again, in the four-person case, the departure requires three new key exchanges; nine are required in the ten-person case. Clearly, this will not work for large organizations with hundreds or thousands of employees.

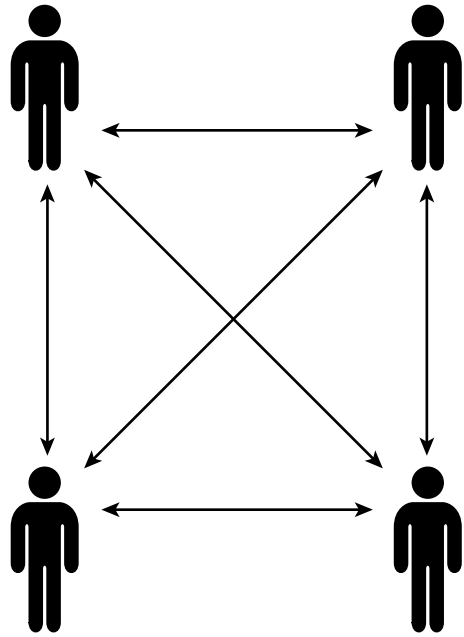


Exhibit 16-1. Four people require six keys.

In short, secret key cryptography has great power, employs fairly simple mathematics, and can quickly encrypt large volumes of data. However, its Achilles heel is the problem of key distribution and maintenance.

This Achilles heel led a group of mathematicians to develop a new paradigm for cryptography — asymmetric cryptography, also known as public key cryptography. Public key cryptography lessens the key distribution problem by splitting the encryption key into a public portion — which is given out to anyone — and a secret component that must be controlled by the user. The public and private keys, which jointly are called a key pair, are generated together and are related through complex mathematics. In the public key model, a sender looks up the recipient's public keys, typically stored in certificates, and encrypts the document using those public keys. No previous connection between sender and recipient is required, because only the recipient's public key is needed for secure transmission, and the certificates are stored in public databases. Only the private key that is associated with the public key can decrypt the document. The public and private keys can be stored as files, as entries in a database, or on a piece of hardware called a token. These tokens are often smart cards that look like credit cards but store user keys and are able to perform cryptographic computations far more quickly than general-purpose CPUs.

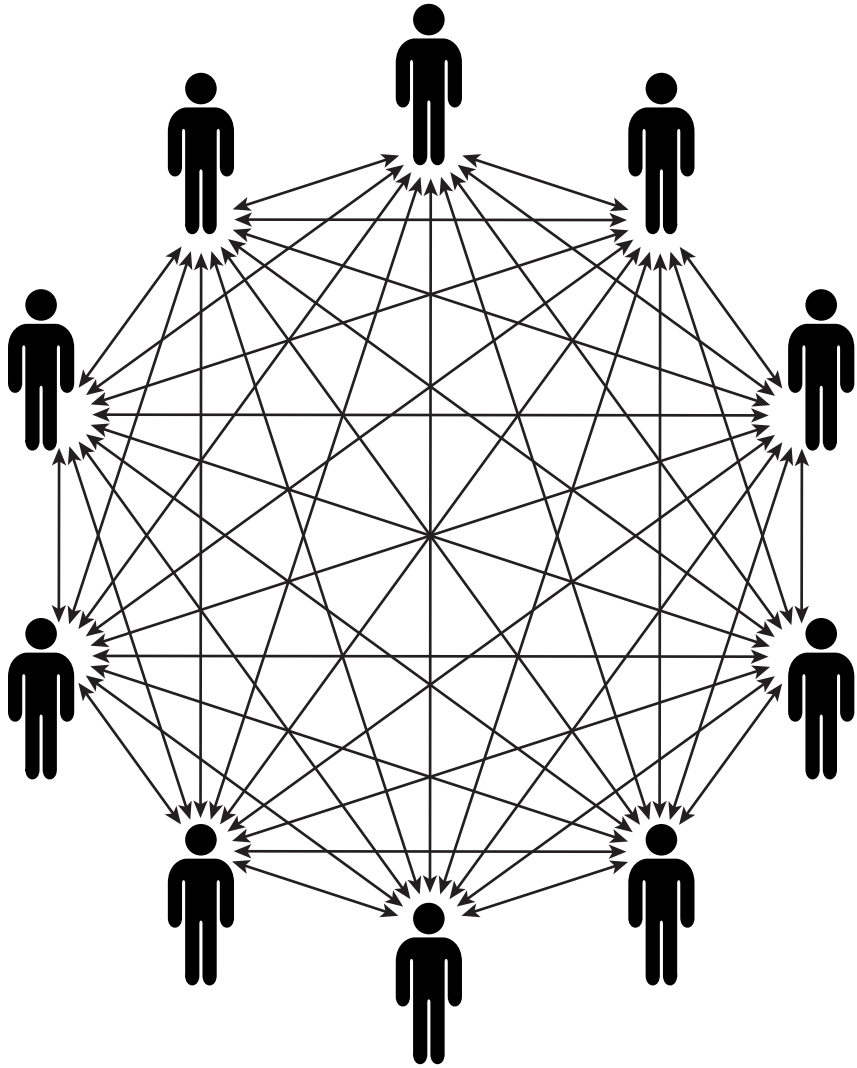


Exhibit 16-2. Ten people require 45 keys.

There are several public key cryptographic algorithms, including RSA, Diffie-Hellman, and Elliptic Curve cryptography. These algorithms relay on the assumption that there are mathematical problems that are easy to perform but difficult to do in reverse. To demonstrate this to yourself, calculate 11 squared (11^2). Now calculate the square root of 160. The square root is a bit more difficult, right? This is the extremely simplified idea behind public key cryptography. Encrypting a document to someone is akin to squaring a number, while decrypting it without the private key is somewhat

like taking the square root. Each of the public key algorithms uses a different type of problem, but all rely on the assumption that the particular problem chosen is difficult to perform in reverse without the key.

Most public key algorithms have associated "signature" algorithms that can be used to ensure that a piece of data was sent by the owner of a private key and was unchanged in transit. These digital signature algorithms are commonly employed to ensure data integrity, but do not, in and of themselves, keep data confidential.

Public key cryptography can be employed to protect data confidentiality and integrity while it is being transported across the network. In fact, Secure Sockets Layer (SSL) is just that: a server's public key is used to create an encrypted tunnel across which World Wide Web (WWW) data is sent. SSL is commonly used for WWW sites that accept credit card information; in fact, the major browsers support SSL natively, as do most Web servers. Unfortunately, SSL does not address all the issues facing an organization that wants to open up its data to network access. By default, SSL authenticates only the server, not the client. However, an organization would want to provide its data only to the correct person; in other words, the whole point of this exercise is to ensure that the client is authenticated.

The SSL standards provide methods to authenticate not only the server, but also the client. Doing this requires having the client side generate a key pair and having the server check the client keys. However, how can the server know that the supposed client is not an imposter even if the client has a key pair? Additionally, even if a key does belong to a valid user, what happens when that user leaves the company, or when the user's key is compromised? Dealing with these situations requires a process called key revocation. Finally, if a user generates a key pair, and then uses that key pair to, for example, encrypt attachments to business-related electronic mail, the user's employer may be required by law to provide access to user data when served with a warrant. For an organization to be able to answer such a warrant, it must have "escrowed" a copy of the users' private keys — but how could the organization get a copy of the private key, since the user generated the pair?

Public key cryptography has a major advantage over secret key cryptography. Recall that secret key cryptography required that the sender and recipient share a secret key in advance. Public key cryptography does not require the sharing of a secret between sender and recipients, but is far slower than secret key cryptography, because the mathematics involved are far more difficult.

Although this simplifies key distribution, it does not solve the problem. Public key cryptography requires a way to ensure that John Doe's public key in fact belongs to him, not to an imposter. In other words, anyone could

generate a key pair and assert that the public key belongs to the President of the United States. However, if one were to want to communicate with the President securely, one would need to ensure that the key was in fact his. This assurance requires that a trusted third party assert a particular public key does, in fact, belong to the supposed user. Providing this assurance requires additional elements, which, together make up a public key infrastructure (PKI).

The next section describes a complete solution that can provide data confidentiality and integrity protection for remote access to applications. Subsequent sections point out other advantages yielded by the development of a full-fledged infrastructure.

USING A PKI TO AUTHENTICATE TO AN APPLICATION

Let us first describe, at a high level, how a WWW-based application might employ a PKI to authenticate its users (see Exhibit 16-3). The user directs her WWW browser to the (secured) WWW server that connects to the application. The WWW page uses the form of SSL that requires both server and client authentication. The user must unlock her private key; this is done by entering a password that decrypts the private key. The server asks for the identity of the user, and looks up her public key in a database. After retrieving her public key, the server checks to be sure that the user is still authorized to access the application system, by checking to be sure that the user's key has not been revoked. Meanwhile, the client accesses the key database to get the public key for the server and checks to be sure it has not been revoked. Assuming that the keys are still valid, the server and client engage in mutual authentication.

There are several methods for mutual authentication. Regardless of approach, mutual authentication requires several steps; the major difference between methods is the order in which the steps occur. Exhibit 16-4 presents a simple method for clarity. First, the server generates a piece of random data, encrypts it with the client's public key, and signs it with its own private key. This encrypted and signed data is sent to the client, who checks the signature using the server's public key and decrypts the data. Only the client could have decrypted the data, because only the client has access to the user's private key; and only the server could have signed the data, because to sign the encrypted data, the server requires access to the server's private key. Hence, if the client can produce the decrypted data, the server can believe that the client has access to the user's private key. Similarly, if the client verifies the signature using the server's public key, the client is assured that the server signed the data. After decrypting the data, the client takes it, along with another piece of unique data, and encrypts both with the server's public key. The client then signs this piece of encrypted data and sends it off to the server. The server checks the

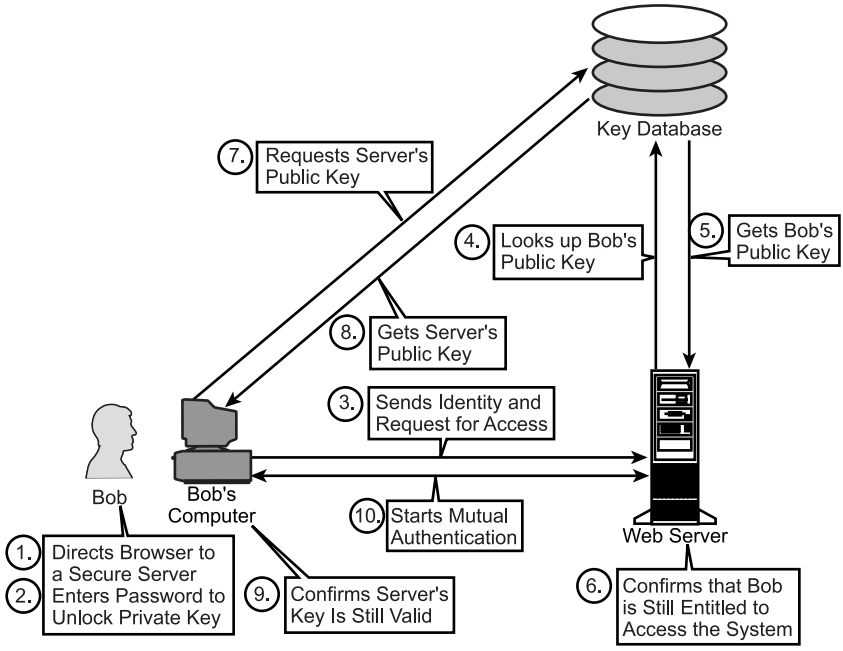


Exhibit 16-3. Using a PKI to authenticate users.

signature, decrypts the data, checks to be sure the first piece of data is the same as what the server sent off before, and gathers the new piece of data. The server generates another random number, takes this new number along with the decrypted data received from the client, and encrypts both together. After signing this new piece of data, the resulting data is sent off to the client. Only the client can decrypt this data, and only the server could have signed it. This series of steps guarantees the identity of each party. After mutual authentication, the server sends a notice to the log server, including information such as the identity of the user, client location, and time.

Recall that public key cryptography is relatively slow; the time required to encrypt and decrypt data could interfere with the user experience. However, if the application used a secret key algorithm to encrypt the data passing over the connection, after the initial public key authentication, the data would be kept confidential to the two participants, but with a lower overhead. This is the purpose of the additional piece of random data in the second message sent by the server. This additional piece of random data will be used as a session key — a secret shared by client and server. Both client and server will use the session key to encrypt all network transactions in the current network connection using a secret key algorithm such as DES, IDEA, or RC4. The secret key algorithm provides confidentiality and

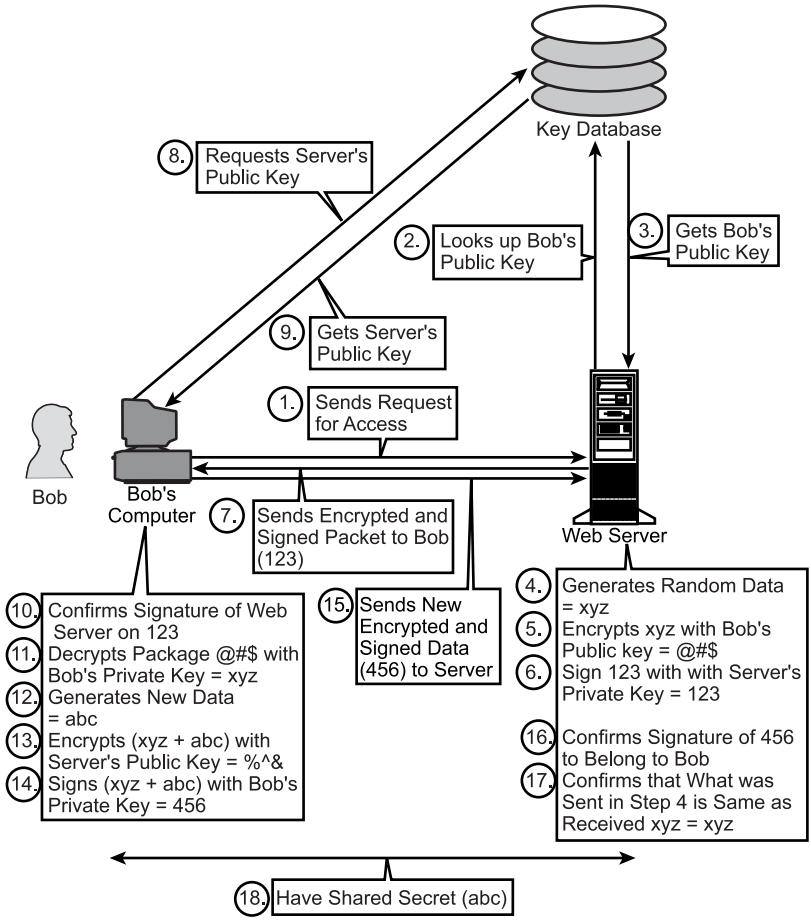


Exhibit 16-4. Mutual authentication.

integrity assurance for all data and queries as they traverse the network without the delay required by a public key algorithm. The public key algorithm handles key exchange and authentication. This combination of both a public key algorithm and a private key one offers the benefits of each.

How did these steps ensure that both client and server were authenticated? The client, after decrypting the data sent by the server, knows that the server was able to decrypt what the client sent, and hence knows that the server can access the server's private key. The server knows that the client has decrypted what it sent in the first step, and thus knows that the client has access to the user's private key. Both parties have authenticated the other, but no passwords have traversed the network, and no information that could be useful to an attacker has left the client or server machines.

Additionally, the server can pass the authentication through to the various application servers without resorting to insecure operating system-level trust relationships, as is often done in multi-system installations. In other words, a user might be able to leverage the public key authentication to not only the WWW-based application, but also other business applications. More details on this reduced sign-on functionality are provided in a later section.

COMPONENTS OF A PKI

The behavior described in the example above seemed very simple, but actually involved several different entities behind the scenes. As is so often the case, a lot of work must be done to make something seem simple. The entities involved here include a certificate authority, registration authorities, directory servers, various application programming interfaces and semi-custom development, third-party applications, and hardware. Some of these entities would be provided by a PKI vendor, such as the CA, RA, and a directory server, but other components would be acquired from other sources. Additionally, the policies that define the overall infrastructure and how the pieces interact with each other and the users are a central component. This section describes each component and tells why it is important to the overall desired behavior.

The basic element of a PKI is the certificate authority. One of the problems facing public key solutions is that anyone can generate a public key and claim to be anyone they like. For example, using publicly available tools, one can generate a public key belonging, supposedly, to the President of the United States. The public key will say that it belongs to the President, but it actually would belong to an imposter. It is important for a PKI to provide assurance that public keys actually belong to the person who is named in the public key. This is done via an external assurance link; to get a key pair, one demonstrates to a human that they are who they claim to be. For example, the user could, as part of the routine on the first day of employment, show his driver's license to the appropriate individual, known as a registration authority. The registration authority (RA) generates a key pair for the individual and tells the certificate authority (CA) to attest that the public key belongs to the individual. The CA does this attestation by signing the public key with the CA's private key. All users trust the CA. Because only the CA could access the CA's private key, and the private key is used to attest to the identity, all will believe that the user is in fact who the user claims to be. Thus, the CA (and associated RA) is required in order for the PKI to be useful, and any compromise of the CA's key is fatal for the entire PKI. CAs and RAs are usually part of the basic package bought from a PKI vendor. An abridged list of PKI vendors (in alphabetical order) includes Baltimore, Entrust Technologies, RSA Security, and Verisign.

When one user (or server) wants to send an encrypted object to another, the sender must get the recipient's public key. For large organizations, there can be thousands of public keys, stored as certificates signed by the CA. It does not make sense for every user to store all other certificates, due to storage constraints. Hence, a centralized storage site (or sites) must store the certificates. These sites are databases, usually accessed via the Lightweight Directory Access Protocol (LDAP), and normally called directory servers. A directory server will provide access throughout the enterprise to the certificates when an entity requires one. There are several vendors for LDAP directories, including Netscape, ICL, Novell, and Microsoft.

There are other roles for directory servers, including escrow of users' private keys. There are several reasons why an organization might need access to users' private keys. If an organization is served by a warrant, it may be required to provide access to encrypted objects. Achieving this usually involves having a separate copy of users' private keys; this copy is called an "escrowed" key. LDAP directories are usually used for escrow purposes. Obviously, these escrow databases must be extremely tightly secured, because access to a user's private key compromises all that user's correspondence and actions. Other reasons to store users' private keys include business continuity planning and compliance monitoring.

When a sender gets a recipient's public key, the sender cannot be sure that the recipient still works for the organization, and does not know if someone has somehow compromised that key pair. Human resources, however, will know that the recipient has left the organization and the user may know that the private key has been compromised. In either case, the certificate signed by the CA — and the associated private key — must be revoked. Key revocation is the process through which a key is declared invalid. Much as it makes little sense for clients to store all certificates, it is not sensible for clients to store all revoked certificates. Rather, a centralized database — called a certificate revocation list (CRL) — should be used to store revoked certificates. The CRL holds identifiers for all revoked certificates. Whenever an entity tries to use a certificate, it must check the CRL in order to ensure that the certificate is still valid; if an entity is presented a revoked certificate, it should log the event as a possible attack on the infrastructure. CRLs are often stored in LDAP databases, in data structures accessible through Online Certificate Status Processing (OCSP), or on centralized revocation servers, as in Valicert's Certificate Revocation Tree service. Some PKIs have ability to check CRLs, such as Entrust's Entelligence client, but most rely on custom software development to handle CRL checking. Additionally, even for PKIs supporting CRL checking, the capabilities do not provide access to other organization's CRLs — only a custom LDAP solution or a service such as, for example, Valicert's can provide this inter-organization (or inter-PKI) capability.

Off-the-shelf PKI tools are often insufficient to provide complete auditing, dual authentication, CRL checking, operating system integration, and application integration. To provide these services, custom development must be performed. Such development requires that the application and PKI both support application programming interfaces (APIs). The API is the language that the application talks and through which the application is extended. There are public APIs for directory servers, operating system authentication, CRL checking, and many more functions. It is very common for applications to support one or more APIs. Many PKI vendors have invested heavily in the creation of toolkits — notably, RSA Security, Entrust Technologies, and Baltimore.

For both performance and security reasons, hardware cryptographic support can be used as part of a PKI. The hardware support is used to generate and store keys and also to speed cryptographic operations. The CA and RAs will almost always require some sort of hardware support to generate and store keys. Potential devices include smart cards, PCMCIA cards, or external devices. An abridged list of manufactures includes Spyrus, BBN, Atalla, Schlumberger, and Rainbow. These devices can cost anywhere from a few dollars up to $5000, depending on model and functionality. They serve not only to increase the performance of CA encryption, but also to provide additional security for the CA private key, because it is difficult to extract the private key from a hardware device.

Normally, one would not employ a smart card on a CA but, if desired, user private keys can be stored on smart cards. Such smart cards may provide additional functionality, such as physical access to company premises. Employing a smart card provides higher security for the user's private key because there is (virtually) no way for the user's private key to be removed from the card, and all computations are performed on the card itself. The downside of smart cards is that each card user must be given both a card and a card reader. Note that additional readers are required anywhere a user wishes to employ the card. There are several card manufacturers, but only some cards work with some PKI selections. The card manufacturers include Spyrus, Litronic, Datakey, and GemPlus. In general, the cards cost approximately $100 per user, including both card and reader.

However, the most important element of a PKI is not a physical element at all, but rather the policies that guide design, implementation, and operation of the PKI. These policies are critical to the success of a PKI, yet are often given short shrift during implementation. The policies are called a "Certificate Practice Statement" (CPS). A CPS includes, among other things, direction about how users are to identify themselves to an RA in order to get their key pair; what the RA should do when a user loses his password (and hence cannot unlock his private key); and how keys should be escrowed, if at all. Additionally, the CPS covers areas such as backup

policies for the directory servers, CA, and RA machines. There are several good CPS examples that serve as the starting point for an implementation. A critical element of the security of the entire system is the sanctity of the CA itself — the root key material, the software that signs certificate requests, and the OS security itself. Extremely serious attention must be paid to the operational policies — how the system is administered, background checks on the administrators, multiple-person control, etc. — of the CA server.

The technology that underpins a PKI is little different from that of other enterprisewide systems. The same concerns that would apply to, for example, a mission-critical database system should be applied to the PKI components. These concerns include business continuity planning, stress and load modeling, service-level agreements with any outsourced providers or contract support, etc. The CA software often runs either on Windows NT or one of the UNIX variants, depending on the CA vendor. The RA software is often a Windows 9x client. There are different architectures for a PKI. These architectures vary on, among other things, the number and location of CA and RA servers, the location, hierarchy, and replication settings of directory servers, and the "chain of trust" that carries from sub-CA servers (if any) back to the root CA server. Latency, load requirements, and the overall security policy should dictate the particular architecture employed by the PKI.

OTHER PKI BENEFITS: REDUCED SIGN-ON

There are other benefits of a PKI implementation — especially the promise of reduced sign-on for users. Many applications require several authentication steps. For example, a user may employ one username and password pair to log on to his local desktop, others to log on to the servers, and yet more to access the application and data itself. This creates a user interaction nightmare; how many usernames and passwords can a user remember? A common solution to this problem is to employ "trust" relationships between the servers supporting an application. This reduces the number of logins a user must perform, because logging into one trusted host provides access to all others. However, it also creates a significant security vulnerability; if an attacker can access one trusted machine, the attacker has full access to all of them. This point has been exploited many times during PricewaterhouseCoopers attack and penetration exercises. The "attackers" find a development machine, because development machines typically are less secure than production machines, and attack it. After compromising the development machine, the trust relationships allow access to the production machines. Hence, the trust relationships mean that the security of the entire system is dependent not on the most secure systems — the production servers — but rather on the least secure ones.

Even using a trust relationship does not entirely solve the user interaction problem; the user still has at least one operating system username and password pair to remember and another application username and password. PKI systems offer a promising solution to this problem. The major PKI vendors have produced connecting software that replaces most operating system authentication processes with a process that is close to the PKI authentication system described above.

The operating system authentication uses access to the user's private key, which is unlocked with a password. After unlocking the private key, it can be used in the PKI authentication process described above. Once the private key is unlocked, it remains unlocked for a configurable period of time. The user would unlock the private key when first used, which would typically be when logging in to the user's desktop system. Hence, if the servers and applications use the PKI authentication mechanism, the users will not need to reenter a password — they need unlock the private key only once. Each system or application can, if it desires, engage in authentication with the user's machine, but the user need not interact, because the private key is already unlocked. From the user's perspective, this is single sign-on, but without the loss of security provided by other partial solutions (such as trust relationships).

There are other authentications involved in day-to-day business operations. For example, many of us deal with legacy systems. These legacy systems have their own, often proprietary, authentication mechanisms. Third-party products provide connections between a PKI and these legacy applications. A username and password pair is stored in a protected database. When the user attempts to access the legacy application, a "proxy" application requests PKI-based authentication. After successfully authenticating the user — which may not require reentry of the user's PKI password — the server passes the legacy application the appropriate username and password and connects the client to the legacy application. The users need not remember the username and password for the legacy application because they are stored in the database. Because the users need not remember the password, the password can be as complicated as the legacy application will accept, thus making security compromise of the legacy application more difficult while still minimizing user interaction headaches.

Finally, user keys, as mentioned above, can be stored as files or on tokens, often called smart cards. When using a smart card, the user inserts the card into a reader attached to the desktop and authenticates to the card, which unlocks the private key. From then on, the card will answer challenges sent to it and issue them in turn, taking the part of the client machine in the example above. Smart cards can contain more than simply the user keys, although this is their main function. For example, a person's picture can be printed onto the smart card, thus providing a corporate

identification badge. Magnetic stripes can be put on the back of the smart card and encoded with normal magnetic information. Additionally, smart card manufacturers can build proximity transmitters into their smart card. These techniques allow the same card that authenticates the user to the systems to allow the user access to the physical premises of the office. In this model, the PKI provides not only secure access to the entity's systems and applications with single sign-on, but also to physically secured areas of the entity. Such benefits are driving the increase in the use of smart cards for cryptographic security.

PKI IN OPERATION

With the background of how a PKI works and descriptions of its components, one can now walk through an end-to-end example of how a hypothetical organization might operate its PKI.

Imagine a company, DCMEF, Inc., which has a few thousand employees located primarily in southern California. DCMEF, Inc. makes widgets used in the manufacture of automobile air bags. DCMEF uses an ERP system for manufacturing planning and scheduling as well as for its general ledger and payables. It uses a shop-floor data management system to track the manufacturing process, and has a legacy system to maintain human resource-related information. Employees are required to wear badges at all times when in the facility, and these same picture badges unlock the various secured doors at the facility near the elevators and at the entrances to the shop floor via badge readers.

DCMEF implemented its PKI in 1999, using commercial products for CA and directory services. The CA is located in a separately secured data center, with a warm standby machine locked in a disaster recovery site in the Midwest. The warm standby machine does not have keying material. The emergency backup CA key is stored in a safety deposit box that requires the presence of two corporate officers or directors to access. The CA is administered by a specially cleared operations staff member who does not have access to the logging server, which ensures that that operations person cannot ask the CA to do anything (such as create certificates) without a third person seeing the event. The RA clients are scattered through human resources, but are activated with separate keys, not the HR representatives' normal day-to-day keys.

When new employees are hired, they are first put through a two-day orientation course. At this course, the employees fill out their benefits forms, tax information, and also sign the data security policy form. After signing the form, each employee is given individual access to a machine that uses cryptographic hardware support to generate a key pair for that user. The public half of the key pair is submitted to the organization's CA for certification by

the human resources representative, who is serving as the RA, along with the new employee's role in the organization.

The CA checks to be sure that the certificate request is correctly formed and originated with the RA. Then, the CA creates and signs a certificate for the new employee, and returns the signed certificate to the human resources representative. The resulting certificate is stored on a smart card at that time, along with the private key. The private key is locked on the smart card with a PIN selected by the user (and known only to that user). DCMEF's CPS specifies a four-digit PIN, and prohibits use of common patterns like "1234" or "1111." Hence, each user selects four digits; those who select inappropriate PIN values are prompted to select again until their selection meets DCMEF policies.

A few last steps are required before the user is ready to go. First, a copy of each user's private key is encrypted with the public key of DCMEF's escrow agent and stored in the escrow database. Then, the HR representative activates the WWW-based program that stores the new employee's certificate in the directory server, along with the employee's phone number and other information, and adds the employee to the appropriate role entry in the authentication database server. After this step, other employees will be able to look up the new employee in the company electronic phone book, be able to encrypt e-mail to the new employee, and applications will be able to determine the information to which the employee should have access. After these few steps, the user is done generating key material.

The key generating machine is rebooted before the next new employee uses it. During this time, the new employee who is finished generating a key pair is taken over to a digital camera for an identification photograph. This photograph is printed onto the smart card, and the employee's identification number is stored on the magnetic strip on the back of the card to enable physical access to the appropriate parts of the building.

At this point, the new employees return to the orientation course, armed with their smart cards for building access loaded with credentials for authentication to the PKI. This entire process took less than 15 minutes per employee, with most of that spent typing in information.

The next portion of the orientation course is hands-on instruction on using the ERP modules. In a normal ERP implementation, users have to log on to their client workstation, to an ERP presentation server and, finally, to the application itself. In DCMEF, Inc., the users need only insert their smart cards into the readers attached to their workstations (via either the serial port or a USB port, in this case), and they are logged in transparently to their local machine and to every PKI-aware application — including the ERP system. When the employees insert their smart cards, they are

prompted for the PIN to unlock their secret key. The remainder of the authentication to the client workstation is done automatically, in roughly the manner described above. When the user starts the ERP front-end application, it expects to be given a valid certificate for authentication purposes, and expects to be able to look that certificate up in an authorization database to select which ERP data this user's role can access. Hence, after the authentication process between ERP application server and user (with the smart card providing the user's credentials) completes, the user has full access to the appropriate ERP data. The major ERP packages are PKI-enabled using vendor toolkits and internal application-level controls. However, it is not always so easy to PKI-enable a legacy application, such as DCMEF's shop-floor data manager. In this case, DCMEF could have chosen to leave the legacy application entirely alone, but that would have meant users would need to remember a different username and password pair to gain access to the shop-floor information, and corporate security would need to manage a second set of user credentials. Instead, DCMEF decided to use a gateway approach to the legacy application. All network access to the shop-floor data manager system was removed, to be replaced by a single gateway in or out. This gateway ran customized proxy software that uses certificates to authenticate users. However, the proxy issues usernames and passwords that match the user's role to the shop-floor data manager. There are fewer roles than users, so it is easier to maintain a database of role-password pairs, and the shop-floor data manager itself does not know that anything has changed. The proxy application must be carefully designed and implemented, because it is now a single point of failure for the entire application, and the gateway machine should be hardened against attack.

The user credentials issued by HR expire in 24 months — this period was selected based on the average length of employment at DCMEF, Inc. Hence, every two years, users must renew their certificates. This is done via an automatic process; users visit an intranet WWW site and ask for renewal. This request is routed to human resources, which verifies that the person is still employed and is still in the same role. If appropriate, the HR representative approves the request, and the CA issues a new certificate — with the same public key — to the employee, and adds the old certificate to DCMEF's revocation list. If an employee leaves the company, HR revokes the user's certificate (and hence their access to applications) by asking the CA to add the certificate to the public revocation list. In DCMEF's architecture, a promoted user needs no new certificate, but HR must change the permissions associated with that certificate in the authorization database.

This example is not futuristic at all — everything mentioned here is easily achievable using commercial tools. The difficult portions of this example are related to DCMEF itself. HR, manufacturing, planning, and accounting

use the PKI on a day-to-day basis. Each of these departments has its own needs and concerns that need to be addressed up-front, before implementation, and then training, user acceptance, and updates must include each department going forward. A successful PKI implementation will involve far more than corporate information security — it will involve all the stakeholders in the resulting product.

IMPLEMENTING A PKI: GETTING THERE FROM HERE

The technical component of building a PKI requires five logical steps:

1. The policies that govern the PKI, known as a Certificate Practice Statement (CPS), must be created.
2. The PKI that embodies the CPS must be initialized.
3. Users and administration staff must be trained.
4. Connections to secured systems that could circumvent the PKI must be ended.
5. Any other system integration work — such as integrating legacy applications with the PKI, using the PKI for operating system authentication, or connecting back-office systems including electronic mail or human resource systems to the PKI — must be done.

The fourth and fifth steps may not be appropriate for all organizations.

The times included here are based on the authors' experience in designing and building PKI systems, but will vary for each situation. Some of the variability comes from the size of clients; it requires more time to build a PKI for more users. Other variability derives from a lack of other standards; it is difficult to build a PKI if the organization supports neither Windows NT nor UNIX, for example. In any case, the numbers provided here offer a glimpse into the effort involved in implementing a PKI as part of an ERP implementation.

The first step is to create a CPS. Creating a CPS involves taking a commonly accepted framework, such as the National Automated Clearing House Association guidelines, PKIX-4, or the framework promulgated by Entrust Technologies, and adapting it to the needs of the particular organization. The adaptations involve modification of roles to fit organizational structures and differences in state and federal regulation. This step involves interviews and extensive study of the structure and the environment within which the organization falls. Additionally, the CPS specifies the vendor for the PKI as well as for any supporting hardware or software, such as smart cards or directories. Hence, building a CPS includes the analysis stage of the PKI selection. Building a CPS normally requires approximately three person-months, assuming that the organization has in place certain components, such as an electronic mail policy and Internet use policy, and results in a document that needs high-level approval, often including legal review.

The CPS drives the creation of the PKI, as described above. Once the CPS is complete, the selected PKI vendor and products must be acquired. This involves hardware acquisition for the CA, any RA stations, the directories, and secure logging servers, as well as any smart cards, readers, and other hardware cryptographic modules. Operating system and supporting software must be installed on all servers, along with current security-related operating system patches. The servers must all be hardened, as the security of the entire system will rely to some extent on their security. Additional traditional information security work, such as the creation of intrusion detection systems, is normally required in this phase. Many of the servers — especially the logging server — will require hardware support for the cryptographic operations they must perform; these cryptographic support modules must be installed on each server. Finally, with the pieces complete, the PKI can be installed.

Installing the PKI requires, first, generating a "root" key and using that root key to generate a CA key. This generation normally requires hardware support. The CA key is used to generate the RA keys that in turn generate all user public keys and associated private keys. The CA private key signs users' public keys, creating the certificates that are stored on the directory server. Additionally, the RA must generate certificates for each server that requires authentication. Each user and server certificate and the associated role — the user's job — must be entered into a directory server to support use of the PKI by, for example, secure electronic mail. The server keys must be installed in the hardware cryptographic support modules, where appropriate. Client-side software must be installed on each client to support use of the client-side certificates. Additionally, each client browser must be configured to accept the organization's CA key and to use the client's certificate. These steps, taken together, constitute the initialization of the PKI. The time required to initialize a PKI is largely driven by the number of certificates required. In a recent project involving 1000 certificates, ten applications, and widespread use of smart cards, the PKI initialization phase required approximately twelve person-months. Approximately two person-months of that time were spent solely on the installation of the smart cards and readers.

Training cannot be overlooked when installing large-scale systems such as a PKI. With the correct architecture, much of the PKI details are below users' awareness, which minimizes training requirements. However, the users have to be shown how to unlock their certificates, a process that replaces their login, and how to use any ancillary PKI services, such as secure e-mail and the directory. This training is usually done in groups of 15 to 30 and lasts approximately one to two hours, including hands-on time for the trainees.

After training is completed, users and system administration staff are ready to use the PKI. At this point, one can begin to employ the PKI itself.

This involves ensuring that any applications or servers that should employ the PKI cannot be reached without using the PKI. Achieving this goal often requires employing third-party network programs that interrupt normal network processing to require the PKI. Additionally, it may require making configuration changes to routers and operating systems to block back door entry into the applications and servers. Blocking these back-doors requires finding all connections to servers and applications; this is a non-trivial analysis effort that must be included in the project planning.

Finally, an organization may want to use the PKI to secure applications and other business processes. For example, organizations, as described above, may want to employ the PKI to provide single sign-on or legacy system authentication. This involves employing traditional systems integration methodologies — and leveraged software methodologies — to mate the PKI to these other applications using various application programming interfaces. Estimating this effort requires analysis and requirements assessment.

As outlined here, a work plan for creating a PKI would include five steps. The first step is to create a CPS. Then, the PKI is initialized. Third, user and administrator training must be performed. After training, the PKI connections must be enforced by cutting off extraneous connections. Finally, other system integration work, including custom development, is performed.

CONCLUSION

Security is an enabler for electronic business; without adequate security, senior management may not feel confident moving away from more expensive and slower traditional processes to more computer-intensive ones. Security designers must find usable solutions to organizational requirements for authentication, authorization, confidentiality, and integrity. Public key infrastructures offer a promising technology to serve as the foundation for E-business security designs. The technology itself has many components — certificate authorities, registration authorities, directory servers — but, even more importantly, requires careful policy and procedure implementation.

This chapter has described some of the basics of cryptography, both secret and public key cryptography, and has highlighted the technical and procedural requirements for a PKI. The authors have presented the five high-level steps that are required to implement a PKI, and have mentioned some vendors in each of the component areas. Obviously, in a chapter this brief, it is not possible to present an entire workplan for implementing a PKI — especially since the plans vary significantly from situation to situation. However, the authors have tried to give the reader a start toward such a plan by describing the critical factors that must be addressed, and showing how they all work together to provide an adequate return on investment.

116

Preserving Public Key Hierarchy

Geoffrey C. Grabow, CISSP

Public key infrastructures (PKIs) have always been designed with a top-level key called a root key. This single key is responsible for providing the starting point of trust for all entities below it in the hierarchy. If this root key is ever compromised, the entire trust hierarchy is immediately questionable.

The root key is primarily responsible for digitally signing subordinate Certificate Authorities (CAs). A compromise of the root means that an unauthorized CA will appear perfectly valid to users. Users will then engage in a transaction completely unaware that the security upon which they are relying is worse than worthless.

This single root key introduces a single point of failure.

It is a standard practice in security to design and build systems with a series of checks and balances to prevent any one part of the system from causing a catastrophic failure. However, this practice, for all practical purposes, has been ignored when it comes to a hierarchical PKI.

It is the intention of this chapter to propose a system in which this single point of failure is removed.

Cryptographically secure digital timestamps (CSDTs) have been used for a wide variety of purposes, including document archiving, digital notary services, etc. By adding a CSDT to every digital certificate issued within a PKI, one now has a method for ensuring not only that the certificate is valid, but also at what point in time that validity was declared.

When properly configured, certificates within a PKI, which are protected using CSDTs, can survive the compromise of the root key. If the root key is exposed, certificates still have their original value, and all that is lost is the ability to create new certificates. This allows transactions to continue, and the recovery process only requires the replacement of the root key.

A significant advantage of the system proposed herein is that it works within the parameters set forth in existing PKI standards.

Public Key Infrastructure (PKI)

Public key (or asymmetric) cryptography uses two different keys, usually referred to as a public key and a private key. Any information encrypted by K_{PUB}(Recipient) can only be decrypted by K_{PRI}(Recipient), and vice versa. The two keys are mathematically linked and it is computationally infeasible[1] to determine the private key from the public key. This allows the recipient to create a key pair and to publish K_{PUB}(Recipient) in a location that anyone can find it. Once the sender has a copy of K_{PUB}(Recipient), encrypted information can be sent to the recipient without the problem of transporting a secret key.

Sender:

$$\text{DATA} + K_{PUB}(\text{Recipient}) + \text{Encryption algorithm} = EK_{PUB}(\text{Recipient})[\text{Data}]$$

Recipient:

$$EK_{PUB}(Recipient)[DATA] + K_{PRI}(Recipient) + Decryption\ algorithm = Data$$

The reverse of this process is also true. If the recipient encrypts data with K_{PRI}(Recipient), it can be decrypted with K_{PUB}(Recipient). This means that anyone can decrypt the information and confidentiality has not been achieved; but if it can be decrypted using K_{PUB}(Recipient), then only K_{PRI}(Recipient) could have encrypted it, thereby identifying the individual[2] who sent the data. This is the principle behind a digital signature. However, in a true digital signature scheme, only a hash of the data is encrypted/decrypted to save processing time.

Standard PKI Hierarchical Construction

While asymmetric key systems have solved the key management problem in traditional symmetric key systems, they have introduced a new problem called "trust management." This problem raises the question of "How can I be sure the public key I am using really belongs to the intended recipient?" This problem, typically referred to as a man-in-the-middle attack, happens when a third party (attacker) introduces its public key to the sender, who is fooled into believing that it is the public key of recipient, and vice versa. Obviously, this would allow the attacker to read and potentially modify all communication between the sender and the recipient without either of them being aware of the attacker whatsoever.

This problem is solved through the use of a Certificate Authority. The CA digitally signs a certificate that belongs to the sender and another certificate that belongs to the recipient. The certificate includes the name and public key of its owners, the integrity of which can be checked through the use of the CA's public key. Unfortunately, that means that the sender and the recipient must belong to the same CA. If they are not members of the same CA, a hierarchy of CAs must be established (see Exhibit 116.1).

Each entity in Exhibit 116.1 has its own certificate that is signed by an entity higher up in the hierarchy. This is the method used to transfer trust from a known entity to one that is unknown. The exception to this is the Root, which creates a self-signed certificate. The Root must establish trust through direct contact and business relationships with the CAs.

In this environment, Alice can digitally sign a document and send it to Bob, along with a copy of her certificate as well as the certificate of CA#1. Because Bob already has a trust relationship with CA#2, and CA#2 has a trust relationship with the Root, Bob can validate the certificate of CA#2 and then validate Alice's certificate. Once Bob trusts Alice's certificate, he believes that anything that he can verify with Alice's public key must have been signed by Alice's private key, and therefore must have come from Alice.

The Impact of a Root Key Compromise

The problem with this hierarchical construction is the total reliance on the security of the Root private key. If the K_{PRI}(Root) is compromised by an attacker, that attacker can create a fraudulent CA#3, and then fraudulent

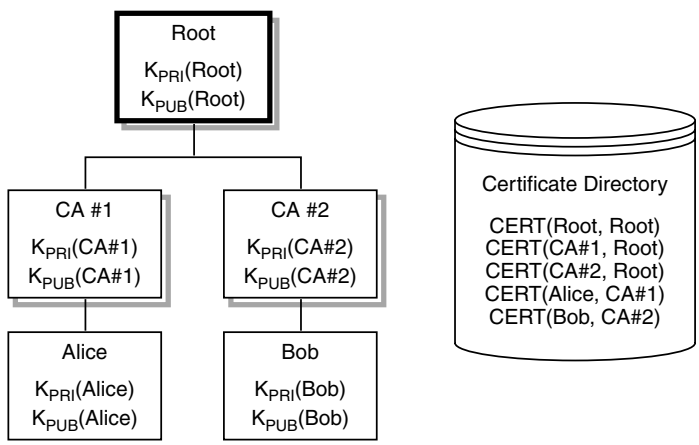


EXHIBIT 116.1 Basic PKI hierarchy

users under that CA. Because CA#3 can be positively validated using the public key of the Root, Alice, Bob, and everyone who trusts the Root will accept any users under CA#3. This puts Alice, Bob, and everyone else in this hierarchy in a situation in which they are trusting fraudulent users and are unaware that there is a problem.

If this occurs, the entire system falls apart. No transactions can take place because there is no basis for trust. An even more significant impact of this situation is that as soon as Alice and Bob are informed about the problem, they will not only stop trusting users under CA#3, but also not be able to trust anyone in the entire hierarchy. Because a CA#3 was created fraudulently, any number of fraudulent CAs can be created and there is no way to determine the CAs not to be trusted from those that should be.

If one cannot determine which CAs are to be trusted, then there is no way to determine which users' certificates are to be trusted. This causes the complete collapse of the entire hierarchy, from the top down.

Constructing Cryptographically Secure Digital Timestamps

Cryptographically secure digital timestamps (CSDTs) are nothing new. A wide variety of applications have been making use of secure timestamps for many years. It is not the intention of this chapter to delve into the details of the actual creation of a CSDT, but rather to indicate the minimum required data for inclusion within digital certificates.

Timestamp

Of course, because one of the primary components of a CSDT is the timestamp itself, a "trusted" time source is required. This can be achieved in several accepted methods and, for the purposes of this construct, it will be assumed that the actual timestamp within the CSDT is the correct one.

To allow for high-volume transaction environments, a 16-bit sequence number is appended to the timestamp to ensure that there can be no two CSDTs with the identical time. This tie-breaker value should be reset with each new timestamp. Therefore, if the time resolution is 0.0001 seconds, it is possible to issue 65,536 CSDTs that all happen within that same 0.0001 second, but the exact sequence of CSDT creation can be determined at any future time.

Hash of the Certificate

For a CSDT to be bound to a particular certificate, some data must be included to tie it to the certificate in question. A hash generated by a known and trusted algorithm, such as SHA-1 or MD5, is used to provide this connection. This is the same hash that is calculated and encrypted during the Certificate Authority signing process.

More importantly, it is critical to know that the time in the CSDT is the time when the CA signs the certificate. Therefore, not just the hash of the certificate should be included, but rather the entire digital signature added to the certificate by the CA. Using the CA's signature will also provide for future changes in CA signing standards.

However, because one of the goals of this chapter is to provide a new feature to existing certificate standards without changing the standards, one cannot append information to the certificate after the signature. Rather, the CSDT must be added to the certificate prior to it being signed by the CA and inserted into an x.509v3 extension field.

Certificate Authority Certificate Hash

As an additional measure, the hash of the CA's certificate is embedded in the CSDT to provide a record of which CA made the request to the Time Authority (TA).

Digital Signature of the Time Authority

To prevent tampering, the CSDT must be cryptographically sealed using a standard digital signature. Because the total amount of data in a CSDT is small, this can be accomplished by simply encrypting the data fields

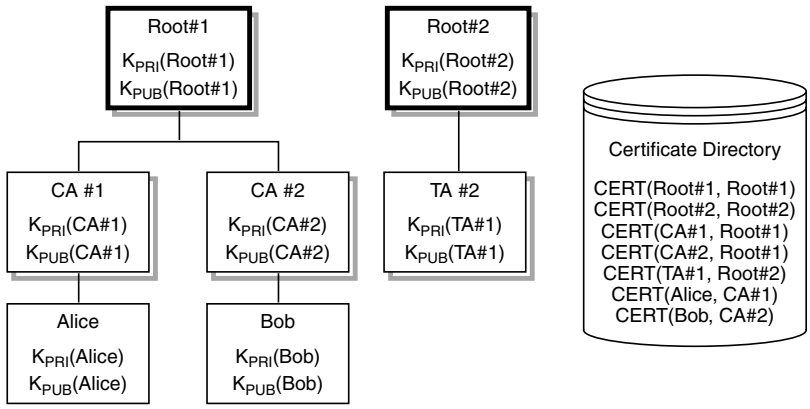


EXHIBIT 116.2 PKI with time authority

with the private key of the TA. However, to allow for growth and additional fields to be added in the future, it is better to encrypt a hash of all of the data to be secured.

Separation of Hierarchies

Of course, the x.509 standard already includes a timestamp so it can be determined at what date and time a certificate was signed by its CA. However, if the root private key was compromised and a fraudulent CA is created, that CA could simply set the time to any value desired prior to signing the certificate.

What is proposed is the inclusion of a timestamp signed by an authority that exists outside the hierarchy of which the CA is part (see Exhibit 116.2).

When a CA creates a certificate, it would follow its normal process for acquiring the public key and other data to be included in the certificate. However, prior to signing the certificate, it would request a CSDT from the Time Authority (TA). This CSDT would then be generated by the TA and returned to the CA. The CA would add the CSDT to the certificate, then sign it in the usual manner.

Should Root#1 be compromised at some point thereafter, all of the CAs created prior to the compromise can still be trusted because access to Root#1 does not give the ability to create the CSDTs. Users can then be informed that anything signed by the Root after a specific date is not to be trusted, but anything signed before that date is still trustworthy.

Walk-Through of Issuance of a Certificate Containing a CSDT

The sequence of events to add a CSDT to a public key certificate is as follows:

1. User generates the public/private key pair.
2. User sends public key and user-specific information to the Registration Authority (RA).
3. RA validates user's request and forwards the certificate request to the CA.
4. CA forms the certificate and calculates the User Certificate Hash (UCH).
5. CA sends a digitally signed request to the Time Authority (TA) containing the UCH.
6. TA receives the request and validates the CA's signature on the request using the CA's public key certificate.
7. TA gets the current time from its secure time source.
8. TA calculates the sequential tie-breaker counter value.
9. TA forms the contents of the CSDT:
   a. UCH (Step 4)
   b. Timestamp (Step 7)
   c. Tie-breaker counter (Step 8)
   d. Hash of CA's certificate (same value used in Step 6)
10. TA calculates the hash of the contents of the CSDT.

11. TA encrypts hash with its private key.
12. TA returns CSDT to the CA.
13. CA validates the TA's signature on the CSDT using the TA's public key certificate.
14. CA verifies UCH in the CSDT against the UCH sent to the TA.
15. CA adds CSDT to the user certificate.
16. CA performs a standard signing process on the completed certificate.
17. CA sends digital certificate to the user.

Recovery Walk-Through

With any system providing assurance, it is necessary to have a plan of action in the event of some problem. The following outlines the minimum necessary steps if a CA is compromised.

Given:

- A CA signed by a CA Root
- A TA signed by a TA Root
- 10,000 users, each of which has generated a public/private key pair
- Each user has gone through the process of getting a public key certificate
- The CA root key is compromised by some form of attack

In infrastructures where CSDTs are not used, all 10,000 user certificates are immediately questionable and cannot be trusted for further transactions. A typical scenario requires the CA to have already created a second replacement root, and to have distributed the second root's self-signed public key certificate when the first was distributed. Users then are told to stop trusting the first root or to delete it from their applications. All users must then generate new key pairs and go through the enrollment process under the new root before business can return to normal.

This is obviously a scenario that requires considerable time and effort, and causes considerable inconvenience for users attempting to execute E-business transactions. Additionally, as the number of users increases, the recovery time increases linearly.

When CSDTs are employed and CSDT-aware applications are used, much of that effort is not required. Immediately upon determining that a compromise has occurred, the CA must:

- Inform the TA not to accept any further requests under the compromised key
- Inform its users
- Generate a new set of keys
- Issue no further certificates under the compromised key

Users need take no action other than to inform their applications of the date/time of the compromise of the CA. All future certificate validation is tested with the CA's certificate as well as the CSDT. If the CA's signature on a certificate is valid, but the CSDT is not present or indicates a date after the compromise, the certificate is rejected and the users are informed that they were presented with an invalid certificate.

Known Issues

Because events such as generating a hash, encryption, and decryption are processes of nonzero duration, it must be acknowledged that the actual time of certificate issuance is not the time within the CSDT. This is not a problem because the time within the CSDT, and within the certificate itself, are not to be used as an absolute time, but rather as a starting point from which the certificate is to be considered valid.

As with any cryptographic system, timely knowledge of any compromise of the system is a critical factor in limiting any "window of opportunity" for an attacker. In this case, it is up to the CA to inform its users that it has had a compromise. Information regarding a compromise of the TA must also be disseminated to users, but users need not take any direct action as a result.

One of the primary responsibilities of a CA is to ensure that everyone who wished to rely on its signature has access to its public key certificate. This is also true for the TA, which must use similar methods to establish trust in its public keys. This may cause some extra effort on the part the CA and its users.

Summary

What has been proposed and discussed in this chapter is a method of providing redundancy in a PKI where none has previously existed. Previous methods of breaking the Root private key into multiple parts created dual control over a single point of failure, but did nothing to provide any systemic redundancy.

It is worthwhile noting that this system is being prototyped by beTRUSTed, the trusted third-party service established by PricewaterhouseCoopers. Their testing, in cooperation with several PKI software vendors, may prove the usefulness and security of this system in a real-world environment.

As with any cryptographic system or protocol, the system of using CSDTs described herein must be analyzed and checked by numerous third parties for possible weaknesses or areas where an attacker may compromise the system.

Notes

1. "Computationally infeasible" indicates that the time or resources required to determine the private key, given only the public key, are well beyond what is available.
2. This assumes that the private keys are generated, used, stored, and destroyed in a secure and proper manner.

Bibliography

1. Improving the Efficiency and Reliability of Digital Timestamping, http://www.surety.com/papers/BHS-paper.pdf.
2. How Do Digital Timestamps Support Digital Signatures?, http://x5.net/faqs/crypto/q108.html.
3. Digital Timestamping Overview, http://www.rsa.com/rsalabs/faq/html/7-11.html.
4. How to Digitally Timestamp a Document, http://www.surety.com/papers/1sttime-stampingpaper.pdf.
5. Answers to Frequently Asked Questions about Today's Cryptography, v3.0, Copyright 1996, RSA Data Security, Inc.

117

PKI Registration

Alex Golod, CISSP

PKI is comprised of many components: technical infrastructure, policies, procedures, and people. Initial registration of subscribers (users, organizations, hardware, or software) for a PKI service has many facets, pertaining to almost every one of the PKI components. There are many steps between the moment when subscribers apply for PKI certificates and the final state, when keys have been generated and certificates have been signed and placed in the appropriate locations in the system. These steps are described either explicitly or implicitly in the PKI Certificate Practices Statement (CPS).

Some of the companies in the PKI business provide all services: hosting Certificate and Registration Authorities (CAs and RAs); registering subscribers; issuing, publishing, and maintaining the current status of all types of certificates; and supporting a network of trust. Other companies sell their extraordinarily powerful software, which includes CAs, RAs, gateways, connectors, toolkits, etc. These components allow buyers (clients) to build their own PKIs to meet their business needs. In all the scenarios, the processes for registration of PKI subscribers may be very different.

This chapter does not claim to be a comprehensive survey of PKI registration. We will simply follow a logical flow. For example, when issuing a new document, we first define the type of document, the purpose it will serve, and by which policy the document will abide. Second, we define policies by which all participants will abide in the process of issuing that document. Third, we define procedures that the parties will follow and which standards, practices, and technologies will be employed. Having this plan in mind, we will try to cover most of the aspects and phases of PKI registration.

CP, CPS, and the Registration Process

The process of the registration of subjects, as well as a majority of the aspects of PKI, are regulated by its Certificate Policies (CP) and Certification Practices Statement (CPS). The definition of CP and CPS is given in RFC 2527, which provides a conduit for implementation of PKIs:

Certificate Policy: A named set of rules indicating the applicability of a certificate to a particular community or class of application with common security requirements. For example, a particular certificate policy might indicate applicability of a type of certificate to the authentication of electronic data interchange transactions for the trading of goods within a given price range.

Certification Practice Statement (CPS): A statement of the practices that a certification authority employs in issuing certificates.

In other words, CP says where and how a relying party will be able to use the certificates. CPS says which practice the PKI (and in many cases its supporting services) will follow to guarantee to all the parties, primarily relying parties and subscribers, that the issued certificates may be used as is declared in CP. The relying parties and subscribers are guided by the paradigm that a certificate "… binds a public key value to a set of information that identifies the entity (such as person, organization, account, or site) associated with use of the corresponding private key (this entity is known as the "subject" of the certificate)."[1] The entity or subject in this quote is also called an *end entity* (EE) or *subscriber.*

A CPS is expressed in a set of provisions. In this chapter we focus only on those provisions that pertain to the process of registration, which generally include:

- Identification and authentication
- Certificate issuance
- Procedural controls
- Key-pairs generation and installation
- Private key protection
- Network security in the process of registration
- Publishing

Reference to CP and CPS associated with a certificate may be presented in the X509.V3 certificates extension called "Certificate Policies." This extension may give to a relying party a great deal of information, identified by attributes *Policy Identifier* in the form of Abstract Syntax Notation One Object IDs (ASN.1 OID) and *Policy Qualifier*. One type of Policy Qualifier is a reference to CPS, which describes the practice employed by the issuer to register the subscriber (the subject of the certificate; see Exhibit 117.1).

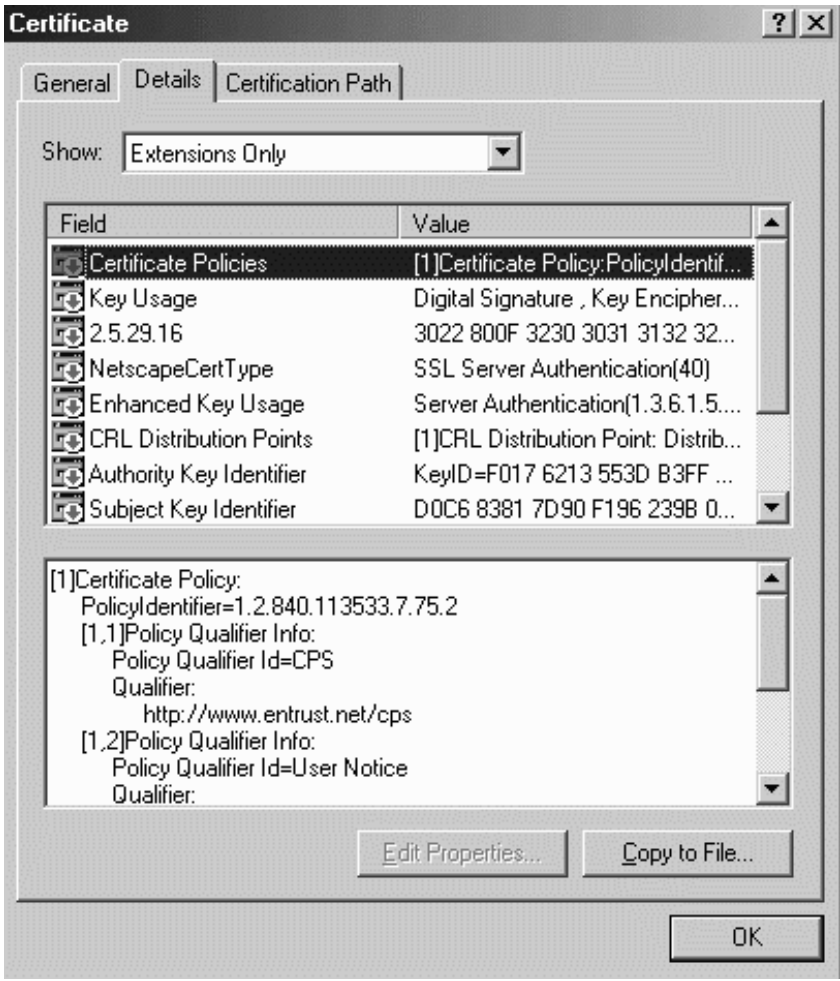


EXHIBIT 117.1 Certificate policies.

Registration, Identification, and Authentication

For initial registration with PKI, a subscriber usually has to go through the processes of identification and authentication. Among the rules and elements that may comprise these processes in a CPS are:

1. Types of names assigned to the subject
2. Whether names have to be meaningful
3. Rules for interpreting various name forms
4. Whether names have to be unique
5. How name claim disputes are resolved
6. Recognition, authentication, and role of trademarks
7. If and how the subject must prove possession of the companion private key for the public key being registered
8. Authentication requirements for organizational identity of subject (CA, RA, or EE)
9. Authentication requirements for a person acting on behalf of a subject (CA, RA, or EE), including:
   - Number of pieces of identification required
   - How a CA or RA validates the pieces of identification provided
   - If the individual must present personally to the authenticating CA or RA
   - How an individual as an organizational person is authenticated

The first six items of the list are more a concern of the legal and naming conventions. They are beyond the scope of this chapter.

Other items basically focus on three issues:

1. How the subject proves its organizational entity (above)
2. How the person, acting on behalf of the subject, authenticates himself in the process of requesting a certificate (above)
3. How the certificate issuer can be sure that the subject, whose name is in the certificate request, is really in the possession of the private key, and which public key is presented in the certificate request along with the subject name (above)

Another important component is the integrity of the process. Infrastructure components and subscribers should be able to authenticate themselves and support data integrity in all the transactions during the process of registration.

How the Subject Proves Its Organizational Entity

Authentication requirements in the process of registration with PKI depend on the nature of applying EE and CP, stating the purpose of the certificate. Among end entities, there can be individuals, organizations, applications, elements of infrastructure, etc.

Organizational certificates are usually issued to the subscribing organization's devices, services, or individuals representing the organization. These certificates support authentication, encryption, data integrity, and other PKI-enabled functionality when relying parties communicate to the organization. Among organizational devices and services may be:

- Web servers with enabled SSL, which support server authentication and encryption
- WAP gateways with WTLS enabled, which support gateway authentication
- Services and devices, signing a content (software codes, documents etc.) on behalf of the organization
- VPN gateways
- Devices, services, applications, supporting authentication, integrity, and encryption of electronic data interchange (EDI), B2B, or B2C transactions

Among procedures enforced within applying organizations (before a certificate request is issued) are:

- An authority inside the organization should approve the certificate request.
- After that, an authorized person within the organization will submit a certificate application on behalf of the organization.

- The organizational certificate application will be submitted for authentication of the organizational identity.

Depending on the purpose of the certificate, a certificate issuer will try to authenticate the applying organization, which may include some but not all of the following steps, as in the example below:[2]

- Verify that the organization exists.
- Verify that the certificate applicant is the owner of the domain name that is the subject of the certificate.
- Verify employment of the certificate applicant and if the organization authorized the applicant to represent the organization.

There is always a correlation between the level of assurance provided by the certificate and the strength of the process of validation and authentication of the EE registering with PKI and obtaining that certificate.

How the Person, Acting on Behalf of the Subject, Authenticates Himself in the Process of Requesting Certificate (Case Study)

Individual certificates may serve different purposes, for example, for e-mail signing and encryption, for user authentication when they are connecting to servers (Web, directory, etc.), to obtain information, or for establishing a VPN encryption channel. These kinds of certificates, according to their policy, may be issued to anybody who is listed as a member of a group (for example, an employee of an organization) in the group's directory and who can authenticate himself. An additional authorization for an organizational person may or may not be required for PKI registration.

An individual who does not belong to any organization can register with some commercial certificate authorities with or without direct authentication and with or without presenting personal information. As a result, an individual receives his general use certificate.

Different cases are briefly described below.

Online Certificate Request without Explicit Authentication

As in the example with VeriSign certificate of Class 1, a CA can issue an individual certificate (a.k.a. digital ID) to any EE with an unambiguous name and e-mail address. In the process of submitting the certificate request to the CA, the keys are generated on the user's computer; and initial data for certificate request, entered by the user (user name and e-mail address) is encrypted with a newly generated private key. It is sent to the CA. Soon the user receives by e-mail his PIN and the URL of a secure Web page to enter that PIN to complete the process of issuing the user's certificate. As a consequence, the person's e-mail address and ability to log into this e-mail account may serve as indirect minimal proof of authenticity. However, nothing prevents person A from registering in the public Internet e-mail as person B and requesting, receiving, and using person B's certificate (see Exhibit 117.2).

Authentication of an Organizational Person

The ability of the EE to authenticate in the organization's network, (e.g., e-mail, domain) or with the organization's authentication database may provide an acceptable level of authentication for PKI registration. Even the person's organizational e-mail authentication is much stronger from a PKI registration perspective than authentication with public e-mail. In this case, a user authentication for PKI registration is basically delegated to e-mail or domain user authentication. In addition to corporate e-mail and domain controllers, an organization's HR database, directory servers, or databases can be used for the user's authentication and authorization for PKI registration. In each case an integration of the PKI registration process and the process of user authentication with corporate resources needs to be done (see Exhibit 117.3).

A simplified case occurs when a certificate request is initiated by a Registration Authority upon management authorization. In this case, no initial user authentication is involved.

Individual Authentication

In the broader case, a PKI registration will require a person to authenticate potentially with any authentication bases defined in accordance with CPS. For example, to obtain a purchasing certificate from the CA, which is integrated into a B2C system, a person will have to authenticate with financial institutions — which will secure the person's Internet purchasing transactions. In many cases, an authentication gateway or server will do it, using a user's credentials (see Exhibit 117.4).

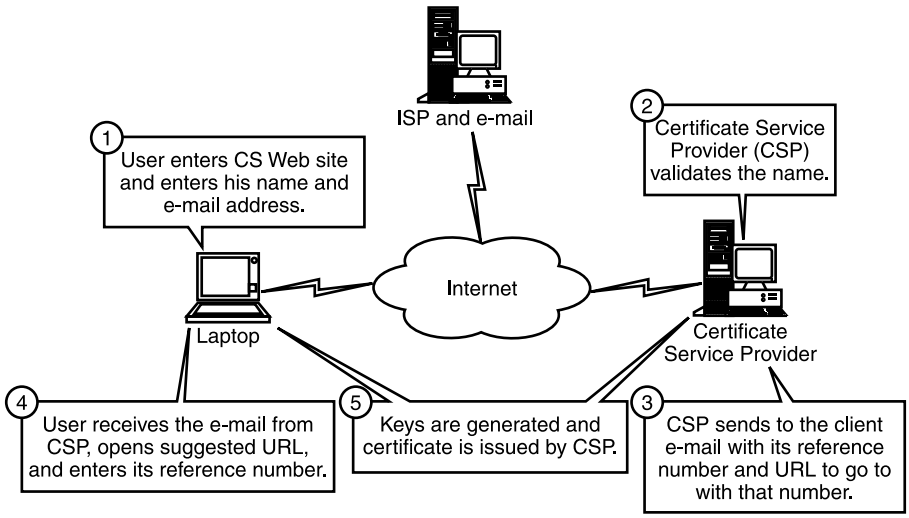


EXHIBIT 117.2 Certificate request via e-mail or Web with no authentication.

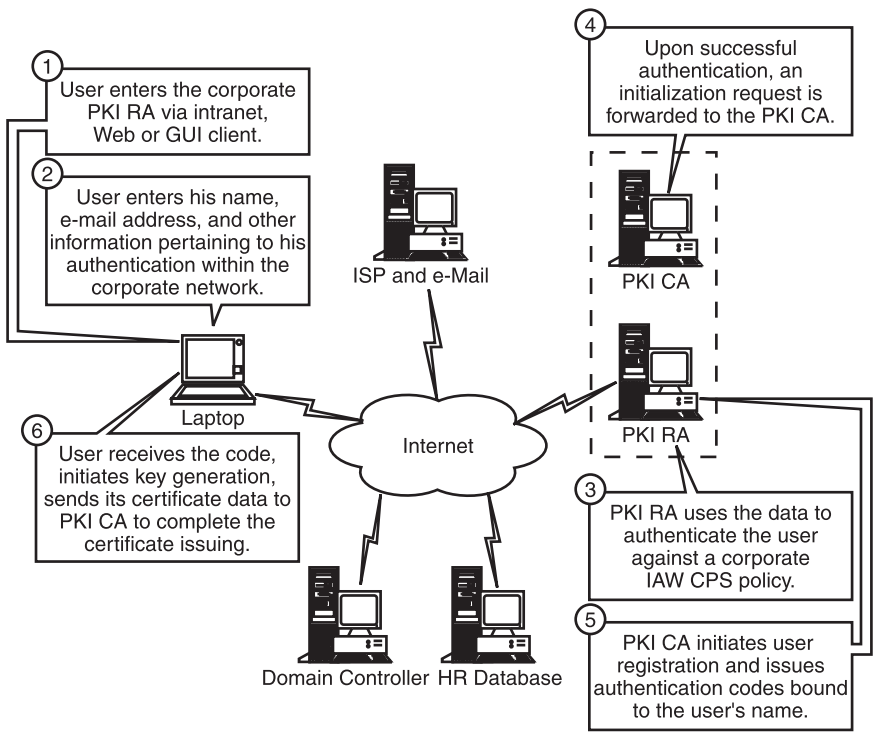


EXHIBIT 117.3 Certificate request via corporate e-mail or Web or GUI interface.

Dedicated Authentication Bases

In rare cases, when a PKI CPS requires a user authentication that cannot be satisfied by the existing authentication bases, a dedicated authentication base may be created to meet all CPS requirements. For example, for this purpose, a prepopulated PKI directory may be created, where each person eligible for PKI registration will be presented with a password and personal data attributes (favorite drink and color, car, etc.). Among

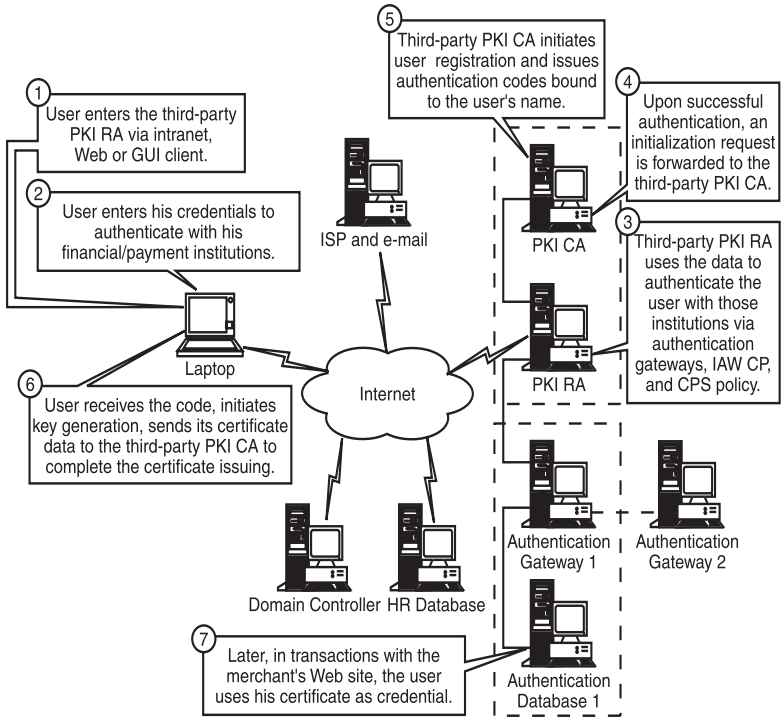


EXHIBIT 117.4 Certificate request via gateway interfaces.

possible authentication schemes with dedicated or existing authentication bases may be personal entropy, biometrics, and others.

Face-to-Face

The most reliable but most expensive method to authenticate an EE for PKI registration is face-to-face authentication. It is applied when the issued certificate will secure either high-risk and responsibility transactions (certificates for VPN gateways, CA and RA administrators) or transactions of high value, especially when the subscriber will authenticate and sign transactions on behalf of an organization. To obtain this type of certificate, the individual must be personally present and show a badge and other valid identification to the dedicated corporate registration security office and sign a document obliging use of the certificate only for assigned purposes. Another example is a healthcare application (e.g., Baltimore-based Healthcare eSignature Authority). All the procedures and sets of ID and documents that must be presented before an authentication authority are described in CPS.

Certificate Request Processing

So far we have looked at the process of EE authentication that may be required by CPS; but from the perspective of the PKI transactions, this process includes out-of-bound transactions. Whether the RA is contacting an authentication database online, or the EE is going through face-to-face authentication, there are still no PKI-specific messages. The RA only carries out the function of personal authentication of an EE before the true PKI registration of the EE can be initialized. This step can also be considered as the first part of the process of initial registration with PKI. Another part of initial registration includes the step of EE initialization, when the EE is requesting information about the PKI-supported functions and acquiring CA public key. The EE is also making itself known to the CA, generating the EE key-pairs and creating a personal secure environment (PSE).

The initial PKI registration process, among other functions, should provide an assurance that the certificate request is really coming from the subject whose name is in the request, and that the subject holds private keys that are the counterparts to the public keys in the certificate request.

These and other PKI functions in many cases rely on PKI Certificate Management Protocols[3] and Certificate Request Management Format.[4]

PKIX-CMP establishes a framework for most of the aspects of PKI management. It is implemented as a message-handling system with a general message format as presented below:[3]

```
PKIMessage :: = SEQUENCE {
header PKIHeader,
body PKIBody,
protection [0] PKIProtection OPTIONAL,
extraCerts [1] SEQUENCE SIZE (1..MAX) OF Certificate
OPTIONAL
}
```

The various messages used in implementing PKI management functions are presented in the PKI message body[3] (see Exhibit 117.5).

Initial Registration

In the PKIX-CMP framework, the first PKI message, related to the EE, may be considered as the start of the initial registration, provided that out-of-bound required EE authentication and CA public key installation have been successfully completed by this time. All the messages that are sent from PKI to the EE must be authenticated. The messages from the EE to PKI may or may not require authentication, depending on the implemented scheme, which includes the location of key generation and the requirements for confirmation messages.

- In the centralized scheme, initialization starts at the CA, and key-pair generation also occurs on the CA. Neither EE message authentication nor confirmation messages are required. Basically, the entire initial registration job is done on the CA, which may send to the EE a message containing the EE's PSE.
- In the basic scheme, initiation and key-pair generation start on the EE's site. As a consequence, its messages to RA and CA must be authenticated. This scheme also requires a confirmation message from the EE to RA/CA when the registration cycle is complete.

Issuing to the EE an authentication key or reference value facilitates authentication of any message from the EE to RA/CA. The EE will use the authentication key to encrypt its certificate request before sending it to the CA/RA.

Proof of Possession

A group of the key PKIX-CMP messages, sent by the EE in the process of initial registration, includes "ir," "cr," and "p10cr" messages (see the PKI message body above). The full structure of these messages is described in RFC 2511 and RSA Laboratories' Public-Key Cryptography Standards (PKCS). Certificate request messages, among other information, include "publicKey" and "subject" name attributes.

The EE has authenticated itself out-of-bound with RA on the initialization phase of initial registration (see above section on registration, identification, and authentication). Now an additional proof is required — that the EE, or the subject, is in possession of a private key, which is a counterpart of the public Key in the certificate request message. It is a proof of binding, or so-called proof of possession, or POP, which the EE submits to the RA.

Depending on the types of requested certificates and public/private key-pairs, different POP mechanisms may be implemented:

- For encryption certificates, the EE can simply provide a private key to the RA/CA, or the EE can be required to decrypt with its private key a value of the following data, which is sent back by RA/CA:
  - In the direct method it will be a challenge value, generated and encrypted and sent to the EE by the RA. The EE is expected to decrypt and send the value back.

EXHIBIT 117.5 Messages Used in Implementing PKI Management Functions

```
PKIBody :: = CHOICE {— message-specific body elements
      ir    [0] CertReqMessages,— Initialization Request
      ip    [1] CertRepMessage,— Initialization Response
      cr    [2] CertReqMessages,— Certification Request
      cp    [3] CertRepMessage,— Certification Response
      p10cr [4] CertificationRequest,— PKCS #10 Cert. Req.
              — the PKCS #10
                                                certification request*
      popdecc[5] POPODecKeyChallContent,— pop Challenge
      popdecr[6] POPODecKeyRespContent,— pop Response
      kur   [7] CertReqMessages,— Key Update Request
      kup   [8] CertRepMessage,— Key Update Response
      krr   [9] CertReqMessages,— Key Recovery Request
      krp   [10] KeyRecRepContent,— Key Recovery Response
      rr    [11] RevReqContent,— Revocation Request
      rp    [12] RevRepContent,— Revocation Response
      ccr   [13] CertReqMessages,— Cross-Cert. Request
      ccp   [14] CertRepMessage,— Cross-Cert. Response
      ckuann[15] CAKeyUpdAnnContent,— CA Key Update Ann.
      cann  [16] CertAnnContent,— Certificate Ann.
      rann  [17] RevAnnContent,— Revocation Ann.
      crlann[18] CRLAnnContent,— CRL Announcement
      conf  [19] PKIConfirmContent,— Confirmation
      nested[20] NestedMessageContent,— Nested Message
      genm  [21] GenMsgContent,— General Message
      genp  [22] GenRepContent,— General Response
      error [23] ErrorMsgContent— Error Message
}
```

* RSA Laboratories, Public-Key Cryptography Standards (PKCS), RSA Data Security Inc., Redwood City, CA, November 1993 release.

Source: RFC 2510.

- In the indirect method, the CA will issue the certificate, encrypt it with the given public encryption key, and send it to the EE. The subsequent use of the certificate by the EE will demonstrate its ability to decrypt it, hence the possession of a private key.
- For signing certificates, the EE merely signs a value with its private key and sends it to the RA/CA.

Depending on implementation and policy, PKI parties may employ different schemes of PKIX-CMP message exchange in the process of initial registration (see Exhibit 117.6).

An initialization request ("ir") contains, as the PKIBody, a CertReqMessages data structure that specifies the requested certificate. This structure is represented in RFC 2511 (see Exhibit 117.7).

A registration/certification request ("cr") may also use as PKIBody a CertReqMessages data structure, or alternatively ("p10cr"), a CertificationRequest.[5]

Administrative and Auto-Registration

As we saw above, the rich PKIX-CMP messaging framework supports the inbound initial certificate request and reply, message authentication, and POP. However, it does not support some important out-of-bound steps of PKI initial registration, such as:

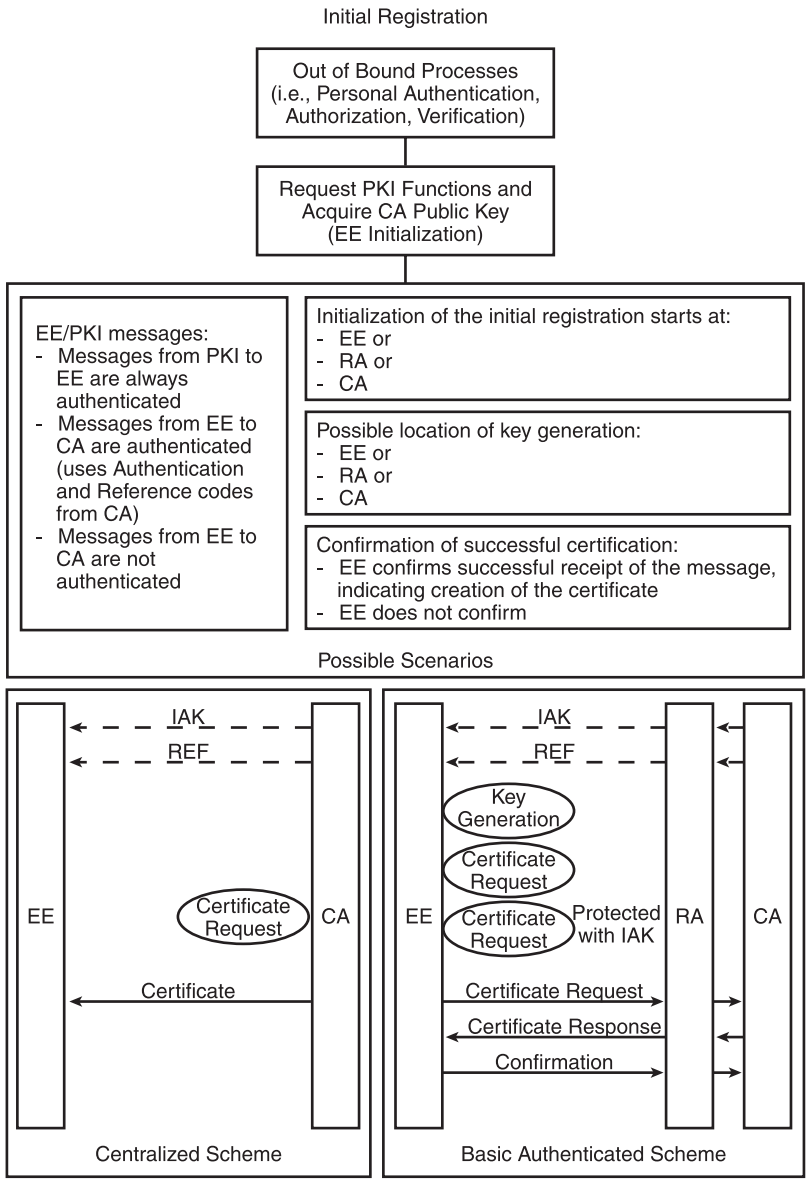


EXHIBIT 117.6 Different schemes of PKIX-CMP message exchange.

- Authentication of an EE and binding its personal identification attributes with the name, which is a part of the registration request
- Administrative processes, such as managers' approval for PKI registration

To keep the PKIX-CMP framework functioning, the EE can generally communicate either directly with the CA or via the RA, depending on specific implementation. However, the CA cannot support the out-of-bound steps of initial registration. That is where the role of the RA is important. In addition to the two functions above, the RA also assumes some CA or EE functionality, such as initializing the whole process of initial registration and completing it by publishing a new certificate in the directory.

In the previous section on "Certificate Request Processing," we briefly mentioned several scenarios of user authentication. In the following analysis we will not consider the first scenario (online certificate request without explicit authentication) because certificates issued in this way have a very limited value.

```
CertReqMessages :: = SEQUENCE SIZE (1..MAX) OF CertReqMsg
CertReqMsg :: = SEQUENCE {
      certReqCertRequest,
      pop   ProofOfPossession OPTIONAL,
 —    content depends upon key type
      regInfoSEQUENCE SIZE(1..MAX) OF AttributeTypeAndValue
                  OPTIONAL}
CertRequest :: = SEQUENCE {
      certReqIdINTEGER,— ID for matching request and reply
      certTemplateCertTemplate,— Selected fields of cert to be issued
      controlsControls OPTIONAL}— Attributes affecting issuance
CertTemplate :: = SEQUENCE {
      version[0] VersionOPTIONAL,
      serialNumber[1] INTEGEROPTIONAL,
      signingAlg[2] AlgorithmIdentifierOPTIONAL,
      issuer[3] NameOPTIONAL,
      validity[4] OptionalValidityOPTIONAL,
      subject[5] NameOPTIONAL,
      publicKey[6] SubjectPublicKeyInfoOPTIONAL,
      issuerUID[7] UniqueIdentifierOPTIONAL,
      subjectUID[8] UniqueIdentifierOPTIONAL,
      extensions[9] ExtensionsOPTIONAL}
OptionalValidity :: = SEQUENCE {
      notBefore[0] Time OPTIONAL,
      notAfter[1] Time OPTIONAL} — at least one must be present
Time :: = CHOICE {
      utcTimeUTCTime,
      generalTimeGeneralizedTime}
```

EXHIBIT 117.7 Data structure specifying the requested certificate.

Case Study

The following are examples of the initial registration, which requires explicit EE authentication.

Administrative Registration

1. An EE issues an out-of-bound request to become a PKI subscriber (either organizational or commercial third party).
2. An authorized administrator or commercial PKI clerk will authenticate EE and verify its request. Upon successful authentication and verification, an authorized administrator submits the request to the RA administrator.
3. The RA administrator enters the EE subject name and, optionally, additional attributes into the RA to pass it to the CA. The CA will verify if the subject name is not ambiguous and will issue a reference number (RN) to associate the forthcoming certificate request with the subject and an authentication code (AC) to encrypt forthcoming communications with EE.
4. The RA administrator sends the AC and RN in a secure out-of-bound way to the EE.
5. The EE generates a signing key-pair, and using AC and RN, establishes inbound "ir" PKIX-CMP exchange.
6. As a result, the EE's verification and encryption certificates, along with signing and decryption keys, are placed in the EE PSE. The EE's encryption certificate is also placed in the public directory.
7. If the keys are compromised or destroyed, the PKI administrator should start a recovery process, which quite closely repeats the steps of initial registration described here.

As we see, most of the out-of-bound steps in each individual case of administrative PKI registration are handled by administrators and clerks. Moreover, the out-of-bound distribution of AC/RN requires high confidentiality.

Auto-Registration

1. Optionally (depending on the policy), an EE may have to issue an out-of-bound application to become a PKI subscriber (either organizational or commercial third party). An authorized administrator or commercial PKI clerk will evaluate the request. Upon evaluation, the EE will be defined in the organizational or commercial database as a user, authorized to become a PKI subscriber.
2. The EE enters his authentication attributes online in the predefined GUI form.
3. The form processor (background process of the GUI form) checks if the EE is authorized to become a PKI subscriber and then tries to authenticate the EE based on the entered credentials.
4. Upon successful authentication of the EE, the subsequent registration steps are performed automatically, as well as the previous step.
5. As a result, the EE's verification and encryption certificates, along with signing and decryption keys, are placed in the EE PSE. The EE's encryption certificate is also placed in the public directory.
6. If the keys are compromised or destroyed, the EE can invoke via a GUI form a recovery process without any administrator's participation.

Comparing the two scenarios, we can see an obvious advantage to auto-registration. It is substantially a self-registration process. From an administration perspective, it requires simply to authorize the EE to become a PKI subscriber. After that, only exceptional situations may require a PKI administrator's intervention.

Authentication Is a Key Factor

We may assume that in both scenarios described above, all the inbound communications follow the same steps of the same protocol (PKIX-CMP). The difference is in the out-of-bound steps, and more specifically, in the user (EE) authentication. Generally, possible authentication scenarios are described in the section on "Registration, Identification, and Authentication." Most of those scenarios (except face-to-face scenarios) may be implemented either in the administrative or auto-registration stage. The form, sources, and quality of authentication data should be described in the CPS. The stronger the authentication criteria for PKI registration, the more trust the relying parties or applications can use. There may be explicit and implicit authentication factors.

In the administrative registration case above, authentication of the organizational user may be totally implicit, because his PKI subscription may have been authorized by his manager, and AC/RN data may have been delivered via organizational channels with good authentication mechanisms and access control. On the other hand, registration with a commercial PKI may require an EE to supply personal information (SSN, DOB, address, bank account, etc.), which may be verified by a clerk or administrator.

Auto-registration generally accommodates verification of all the pieces of the personal information. If it is implemented correctly, it may help to protect subscribers' privacy, because no personal information will be passed via clerks and administrators. In both the organizational and commercial PKI registration cases, it may even add additional authentication factors — the ability of the EE/user to authenticate himself online with his existing accounts using one or many authentication bases within one or many organizations.

Conclusion

For most common-use certificates, which do not assume a top fiscal or a highest legal responsibility, an automated process of PKI registration may be the best option, especially for large-scale PKI applications and for the geographically dispersed subscribers' base. Improvement of this technology in mitigating possible security risk, enlarging online authentication bases, methods of online authentication, and making the entire automated process more reliable, will allow the organization to rely on it when registering subscribers for more expensive certificates, which assume more responsibility.

For user registration for certificates carrying a very high responsibility and liability, the process will probably remain manual, with face-to face appearance of the applicant in front of the RA, with more than one proof of his identity. It will be complemented by application forms (from the applicant and his superior) and

verification (both online and offline) with appropriate authorities. The number of certificates of this type is not high, and thus does not create a burden for the RA or another agency performing its role.

References

1. S. Chokhani and W. Ford, Internet X.509 Public Key Infrastructure, Certificate Policy and Certification Practices Framework, RFC 2527, March 1999.
2. VeriSign Certification Practices Statement, Version 2.0., August 31, 2001.
3. C. Adams and S. Farrell, Internet X.509 Public Key Infrastructure, Certificate Management Protocols, RFC 2510, March 1999.
4. M. Myers, C. Adams, D. Solo, and D. Kemp, Certificate Request Message Format, RFC 2511, March 1999.
5. RSA Laboratories, *Public-Key Cryptography Standards* (PKCS), RSA Data Security Inc., Redwood City, CA, November 1993 Release.

118

Implementing Kerberos in Distributed Systems

Joe Kovara, CTP and Ray Kaplan, CISSP, CISA, CISM

Kerberos is a distributed security system that provides a wide range of security services for distributed environments. Those services include authentication and message protection, as well as providing the ability to securely carry authorization information needed by applications, operating systems, and networks. Kerberos also provides the facilities necessary for delegation, where limited-trust intermediaries perform operations on behalf of a client. Entering its second decade of use, Kerberos is arguably the best tested and most scrutinized distributed security system in widespread use today.

Kerberos differs from many other distributed security systems in its ability to incorporate a very wide range of security technologies and mechanisms. That flexibility allows a mixture of security technologies and mechanisms to be used, as narrowly or broadly as required, while still providing the economies of scale that come from a common, reusable, and technology-neutral Kerberos security infrastructure. Technologies and mechanisms that have been incorporated into Kerberos and that are in use today include certificate-based public key systems, smart cards, token cards, asymmetric-key cryptography, as well as the venerable user ID and password.

Kerberos' longevity and acceptance in the commercial market are testaments to its reliability, efficiency, cost of ownership, and its adaptability to security technologies past, present, and — we believe — future. Those factors have made Kerberos the *de facto* standard for distributed security in large, heterogeneous network environments. Kerberos has been in production on a large scale for years at a variety of commercial, government, and educational organizations, and for over a decade in one of the world's most challenging open systems environments: Project Athena[1] at MIT, where it protects campus users and services from what is possibly the security practitioner's worst nightmare.

History of Development

Many of the ideas for Kerberos originated in a discussion of how to use encryption for authentication in large networks that was published in 1978 by Roger Needham and Michael Schroeder.[2] Other early ideas can be attributed to continuing work by the security community, such as Dorothy Denning's and Giovanni Sacco's work on the use of time stamps in key distribution protocols.[3] Kerberos was designed and implemented in the mid-1980s as part of MIT's Project Athena. The original design and implementation of the first four versions of Kerberos were done by MIT Project Athena members Steve Miller (Digital Equipment Corp.) and Clifford Neuman, along with Jerome Salzer (Project Athena technical director) and Jeff Schiller (MIT campus network manager).

Kerberos versions 1 through 3 were internal development versions and, since its public release in 1989, version 4 of Kerberos has seen wide use in the Internet community. In 1990, John Kohl (Digital Equipment Corp.) and Clifford Neuman (University of Washington at that time and now with the Information Sciences Institute at the University of Southern California) presented a design for version 5 of the protocol based on input from many of those familiar with the limitations of version 4. Currently, Kerberos versions 4 and 5 are

available from several sources, including freely distributed versions (subject to export restrictions) and fully supported commercial versions. Kerberos 4 is in rapid decline, and support for it is very limited. This discussion is limited to Kerberos 5.

Current Development

Although there have been no fundamental changes to the Kerberos 5 protocol in recent years,[4] development and enhancement of Kerberos 5 continues today.[5] That development continues a history of incremental improvements to the protocol and implementations. Implementation improvements tend to be driven by commercial demands, lessons learned from large deployments, and the normal improvements in supporting technology and methodologies.

Standards efforts within the Internet Engineering Task Force (IETF) continue to play a predominant role in the Kerberos 5 protocol development, reflecting both the maturity of the protocol as well as the volatility of security technology. Protocol development is primarily driven by the emergence of new technologies, and standards efforts continue to provide an assurance of compatibility and interoperability between implementations as new capabilities and technologies are incorporated. Those efforts also ensure that new developments are vetted by the Internet community. Many additions to Kerberos take the form of separate standards, or IETF Request for Comments (RFCs).[6] Those standards make use of elements in the Kerberos protocol specifically intended to allow for extension and the addition and integration of new technologies. Some of those technologies and their integration into Kerberos are discussed in subsequent sections.

As of this writing, both Microsoft[7] and Sun[8] have committed to delivery of Kerberos 5 as a standard feature of their operating systems. Kerberos 5 has also been at the core of security for the Open Software Foundation's Distributed Computing Environment (OSF DCE) for many years.[9] Many application vendors have also implemented the ability to utilize Kerberos 5 in their products, either directly, or through the Generic Security Service Applications Programming Interface (GSS-API).

Standards and Implementations

When discussing any standard, care must be exercised in delineating the difference between what the standard defines, what is required for a solution, and what different vendors provide. As does any good protocol standard, the Kerberos 5 standard leaves as much freedom as possible to each implementation, and as little freedom as necessary to ensure interoperability. The basic Kerberos 5 protocol defines the syntax and semantics for authentication, secure messaging, limited syntax and semantics for authorization, and the application of various cryptographic algorithms within those elements.

The Kerberos 5 protocol implies, but does not define, the supporting infrastructure needed to build a solution that incorporates and makes useful all of the standard's elements. For example, the services that make up the logical grouping of the Kerberos security server are defined by the Kerberos 5 standard. The manifestation of those services — the underlying database that those services require, the supporting management tools, and the efficiency of the implementation — are not defined by the standard. Those elements make the difference between what is theoretically possible and what is real. That difference is a reflection of the state of technology, market demands, and vendor implementation abilities and priorities. In this discussion we have attempted to distinguish between the elements that make up the Kerberos 5 protocol, the elements that are needed to build and deploy a solution, and the variations that can be expected in different implementations.

Perceptions and Technology

A review of perceptions about Kerberos will find many anecdotal and casual assertions about its poor usability, inferior performance, or lack of scalability. This appears to be inconsistent with the acceptance of Kerberos by major vendors and can be confusing to those tasked with evaluating security technologies. Much of that confusion is the result of the unqualified use of the term "Kerberos." Kerberos 4 and Kerberos 5 are very different, and any historical references must be qualified as to which version of Kerberos is the subject. As an early effort in distributed security, considerable study was devoted to the weaknesses, vulnerabilities, and limitations of Kerberos 4 and early drafts of the Kerberos 5 standard.[10] Modern implementations of Kerberos 5 address most, if not all, of those issues.

As a pioneering effort in distributed security, Kerberos exposed many new, and sometimes surprising, security issues. Many of those issues are endemic to distributed environments and are a reflection of organization and culture, and the changing face of security as organizations moved from a centralized to a distributed model. As a product of organization and culture, there is little if anything that technology alone can do to address most of those issues. Many of the resulting problems have been attributed to Kerberos, the vast majority of which are common to all distributed security systems, regardless of the technology used.

Various implementations of Kerberos have dealt with the broader organizational security issues in different ways, and with different degrees of success. The variability in the success of those implementations has also been a source of confusion. Enterprises that have a business need for distributed security and that understand the organizational, cultural, and security implications of distributed environments — or more accurately distributed business — tend to be most successful in deploying and applying Kerberos. Until very recently, organizations that fit that description have been in a small minority. Successes have also been achieved at other organizations, but those implementations tend to be narrowly focused on an application or a group within the organization. It should be no surprise that organizations that are in need of what Kerberos has to offer have been in the minority. Kerberos is a distributed security system. Distributed computing is still relatively young, and the technology and business paradigms are still far from convergence.

Outside of the minority of organizations with a business need for distributed security, attempts to implement broad-based distributed security systems such as Kerberos have generally failed. Horror stories of failed implementations tend to receive the most emphasis and are typically what an observer first encounters. Stories of successful implementations are more difficult to uncover. Those stories are rarely discussed outside of a small community of security practitioners or those directly involved, as there is generally little of interest to the broader community; "we're more secure than we were before" does not make for good press.

Whether drivers or indicators of change, the advent of the Internet and intranets bespeak a shift, as a greater number of enterprises move to more distributed organizational structures and business processes and discover a business need for solutions to distributed security problems. Those enterprises typically look first to the major vendors for solutions. Driven by customer business needs, those vendors have turned to Kerberos 5 as a key element in their security solutions.

Trust, Identity, and Cost

The vast majority of identity information used in organizations by computer systems and applications today is based on IDs and passwords, identity information that is bound to individuals. That is the result of years of evolution of our computer systems and applications. Any security based on that existing identity information is fundamentally limited by the trust placed in that information. In other words, security is limited by the level of trust we place in our current IDs and passwords as a means of identifying individuals.

Fundamentally increasing the level of trust placed in our identity information and the security of any system that uses those identities requires rebinding, or reverifying, individual identities. That is a very, very expensive proposition for all but the smallest organizations. In simple and extreme terms: any authentication technology purporting to improve the authenticity of individuals that is based on existing identity information is a waste of money; any authentication technology that is not based on existing identity information is too expensive to deploy on any but a small scale. This very simple but very fundamental equation limits all security technologies and the level of security that is practical and achievable.

We must use most of our existing identity information; the alternatives are not affordable. Although the situation appears bleak, it is far from hopeless; we must simply be realistic about what can be achieved, and at what cost. There is no "silver bullet." The best that any cost-effective solution can hope to do is establish the current level of trust in individual identities as a baseline and not allow further erosion of that trust. Once that baseline is established, measures can be taken to incrementally improve the situation as needed and as budgets allow. The cheaper those goals can be accomplished, the sooner we will start solving the problem and improving the level of trust we can place in our systems.

Kerberos provides the ability to stop further erosion of our trust in existing identities. Kerberos also allows that level of trust to be improved incrementally, by using technologies that are more secure than IDs and passwords. Kerberos allows both of those to be achieved at the lowest possible cost. The ability for Kerberos to effectively utilize what we have today, stop the erosion, and allow incremental improvement is one of the key factors in the success of Kerberos in real-world environments.

Technology Influences

Although technology continues to advance and provide us with the raw materials for improving Kerberos, many of the assumptions and influences that originally shaped Kerberos are still valid today. Although new security technologies may captivate audiences, the fundamentals have not changed. One fundamental of security that should never be forgotten is that a security system must be affordable and reliable if it is to achieve the goal of improving an organization's security.

An affordable and reliable security system makes the most of what exists, and does not require the use of new, expensive or unproven technologies as a prerequisite to improving security. A good security system such as Kerberos allows those newer technologies to be used but does not mandate them. With rapid advances in technology, single-technology solutions are also doomed to rapid obsolescence. Solutions that are predicated on new technologies will, by definition, see limited deployment until the cost and reliability of those solutions are acceptable to a broad range of organizations. The longer that evolution takes, the higher the probability that even newer technologies will render them, and any investment made in them, obsolete.

Moreover, history teaches us that time provides the only real validation of security. That is a difficult proposition for security practitioners when the norm in the information industry is a constant race of the latest and greatest. However, the historical landscape is littered with security technologies, most created by very smart people, that could not stand the test of time and the scrutiny of the security community. The technology influences that have shaped Kerberos have been based on simple and proven fundamentals that provide both a high degree of assurance and a continuing return on investment.

Protocol Placement

Kerberos is often described as an "application-layer protocol." Although that description is nominally correct, and most descriptions of Kerberos are from the perspective of the application, the unfortunate result is a perception that Kerberos requires modification of applications to be useful. Kerberos is not limited to use at the application layer, nor does Kerberos require modification of applications. Kerberos can be, and is, used very effectively at all layers of the network, as well as in middleware. Placing Kerberos authentication, integrity, confidentiality, and access control services below the application layer can provide significant improvements in security without the need to modify applications. The most obvious example of security "behind the scenes" is the use of Kerberos for authentication and key management in a virtual private network (VPN).

However, there are limits to what can be achieved without the cooperation and knowledge of an application. Those limits are a function of the application and apply to all security systems. Providing an authenticated and encrypted channel (e.g., using a VPN) may improve the security of access to the application and the security of information flowing between a client and the application. However, that alone does nothing to improve the usability of the application and does not take advantage of Kerberos' ability to provide secure single sign-on. For example, an application that insists on a local user ID for the users of that application will require mapping between the Kerberos identity and the application-specific user ID. An application that insists on a password will typically require some form of "password stuffing" to placate the application — even if the password is null. Some applications make life easier by providing hooks, call-outs, or exits that allow augmenting the application with alternative security mechanisms. Other applications that do not provide this flexibility require additional and complex infrastructure in order to provide the appearance of seamless operation. Note that these issues are a function of the applications, and not the security system. All security systems must deal with identical issues, and they will generally be forced to deal with those issues in similar ways.

Although we can formulate solutions to authentication, confidentiality, integrity, and access control that are useful and that are independent of a broad range of applications, the same cannot be said of delegation and authorization. In this context, the assertion that Kerberos requires modification of the application is correct. However, that requirement has little if any affect on the practical employment of Kerberos, because very few applications in use today need, or could make use of, those capabilities. Applications that can understand and make use of those capabilities are just starting to appear.

Passwords

One of the primary objectives of Kerberos has always been to provide security end-to-end. That is, all the way from an individual to a service, without the requirement to trust intermediaries. Kerberos can be, and is, also used to provide security for intermediate components such as computer systems, routers, and virtual private

networks. However, humans present the most significant challenge for any security system, and Kerberos does an exemplary job of meeting that challenge.

The simple user ID and password are far and away the most common basis for identification and authentication used by humans and applications today. Whatever their faults, simple IDs and passwords predominate the security landscape and will likely do so for the foreseeable future. They are cheap, portable, and provide adequate security for many applications — virtually all applications in use today. Kerberos is exceptional in its ability to provide a high level of security with nothing more than those IDs and passwords. Kerberos allows more sophisticated identification and authentication mechanisms to be used, but does not mandate their use.

Kerberos is specifically designed to eliminate the transmission of passwords over the network. Passwords are not transmitted in any form as a part of the Kerberos authentication process. The only case in which a password or a derivation of the password (i.e., a key derived from the password) is transmitted is during a password-change operation — assuming, of course, that passwords are being used for authentication, and not an alternative technology such as smart cards. During a password-change operation, the password or its derivation is always protected using Kerberos confidentiality services.

Cryptography

The need to provide effective security using nothing more than very low-cost methods such as an ID and password has had a significant influence on the Kerberos protocol and its use of cryptography. In particular, using a password as the sole means for identification and authentication requires that the password is the basis of a shared secret between the user and the Kerberos security server. That also requires the use of symmetric-key cryptography. Although shared secrets and symmetric-key cryptography have been derided as "legacy" authentication technology, there are few if any alternatives to passwords if we want to provide an affordable and deployable solution sooner rather than later.

The efficiency of cryptographic methods has also had a significant influence on the protocol and its use of cryptography. Although Kerberos can incorporate asymmetric-key cryptography, such as elliptic curve cryptography (ECC) and RSA, Kerberos can provide all of the basic security services using shared secrets and symmetric-key cryptography. Because of the CPU-intensive nature of asymmetric-key cryptography, the ability to use symmetric-key cryptography is extremely important for environments or applications that are performance-sensitive, such as high-volume transaction-processing systems, where each transaction is individually authenticated.

Online Operation

In a distributed environment, individuals and services are scattered across many computer systems and are geographically dispersed. Whatever their physical distribution, those individuals and services operate within a collective enterprise. Typically, the association between an individual and his access to enterprise services is reestablished at the beginning of each workday, such as through a log-in. Day-to-day work in the distributed enterprise requires an individual to make use of many different services, and an individual typically establishes an association with a service, performs work, and then terminates the association. All of these functions occur online.

The association between individual and service may be very short-lived, such as for the duration of a single transaction. In other cases that association is long-lived and spans the workday. Whatever the duration of the association, the vast majority of work is performed online. That is, the individual and the service interact in real-time. Offline operation, which is sometimes necessary, is fast becoming a rarity. Notable exceptions are "road warriors," who must be capable of operating offline. However, that is a function of the limitations of connectivity, not of any desire to operate offline — as any road warrior will tell you.

The combined ability to provide both efficient and secure access to services, and the ability to serve as the basis for a collective security mechanism is one of Kerberos's major strengths. To deliver those capabilities, and deliver them efficiently, the Kerberos security server operates online. Extending that concept to an aggregate "enterprise security service" that incorporates Kerberos allows economies and efficiencies to be achieved across multiple security functions, including authentication, authorization, access control, and key management — all of which can be provided by, or built from, Kerberos. Although the concept of an aggregate enterprise security service is not native to Kerberos, the union of the two is very natural. Moreover, given the direction of technology and the composition and conduct of modern distributed enterprises, online security services

are both required and desirable. These attributes have much to do with the adoption of Kerberos as the basis for providing enterprise security, as opposed to Internet security.

Organizational Model

There are many different approaches to distributed security, and each involves tradeoffs between scalability and resources. The only objective measure of a distributed security system is cost, as measured by the resources required to achieve a given level of security over a given scale. Resources include computational overhead, network bandwidth, and people. The resulting cost bounds the achievable security and the scalability of the system. The tradeoffs that must be made involve both the technology and the security model appropriate to an organization. The extremes of those organizational models are autocracy and anarchy.

Autocracy

All control flows from a central authority. That authority defines the association between itself and the individual and the level of trust it places in an individual. This model requires a level of control that is cost-prohibitive in today's distributed environments. The classic military or business models tend toward this end of the spectrum.

Anarchy

All authority flows from individuals. Each individual defines the association between himself and an enterprise and the level of trust they place in an enterprise. This model achieves no economies of scale or commonality. The Internet tends toward this end of the spectrum.

Where in that spectrum an enterprise lives depends on business practices and culture, and every enterprise is different. Within a single enterprise it is not unusual to find organizational units that span the entire spectrum. That variability places significant demands on a distributed security system, and in some cases those demands may conflict. Conflicting demands occur when multiple enterprises — or even different business units within the same enterprise — with very different business practices or cultures engage in a common activity, such as is typical in supplier and partner relationships. The extreme case of conflicting demands is most often seen when the enterprise meets the Internet. As enterprise boundaries continue to dissolve, the probability of conflicting demands increases, as does the need for security systems to cope with those conflicting demands.

Kerberos most naturally falls in the middle of the spectrum between the extremes of autocracy and anarchy. Depending on implementation and the technology that is incorporated, Kerberos can be applied to many points along that spectrum and can be used to bridge points along the spectrum. Kerberos' effectiveness drops as you approach the extreme ends of the spectrum. As a security system, Kerberos provides a means to express and enforce a common set of rules across a collective; by definition, that collective is not anarchy. As a distributed security system, Kerberos is designed to solve problems that result from autonomous (and hence untrusted) elements within the environment; by definition, that cannot be an autocracy. Note that "distributed" does not necessarily imply physically distributed. For example, if the LAN to which your computer is connected cannot ensure the confidentiality and integrity of data you send across it, then you are in a distributed security environment.

Trust Models

The level of trust that is required between entities in a distributed system is a distinguishing characteristic of all distributed security systems, and affects all other services that are built on the system, as well as the scalability of the system. A prerequisite to trust is authentication: knowing the identity of the person (or machine) you are dealing with. In Kerberos, the entities that authenticate with one another are referred to as "principals," as in "principals to a transaction."

Direct Trust

Historically, users and applications have established direct trust relationships with one another. For example, each user of each application requires a user ID and password to access that application; the user ID and password represents a direct trust relationship between the user and the application. As the number of users

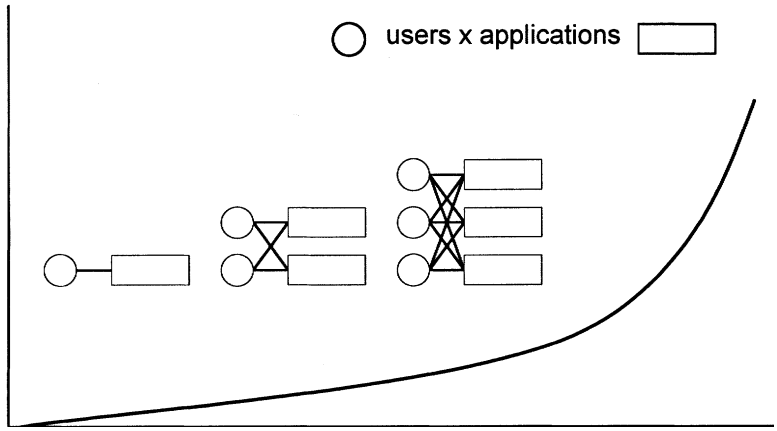


EXHIBIT 118.1 Direct trust relationships.

and applications grows, the number of direct relationships, and the cost of establishing and managing those relationships, increases geometrically (Exhibit 118.1). A geometric increase in complexity and cost is obviously not sustainable and limits the scalability of such solutions to a small number of applications or users.

A secure authentication system does not, in and of itself, reduce the complexity of this problem. The increase in complexity is a function of the number of direct trust relationships and has nothing to do with the security of the user-to-application authentication mechanism. An example of this is seen in Web-based applications that use IDs and passwords for authentication through the SSL (Secure Sockets Layer) protocol. The SSL protocol can provide secure transmission of the ID and password from the client to the server. However, that alone does not reduce the number of IDs and passwords that users and servers must manage.

Mitigating the increasing cost and complexity of direct trust relationships in the form of many IDs and passwords is the same problem that single sign-on systems attempt to solve. One solution is to use the same user ID and password for all applications. However, this assumes that all applications a user has access to are secured to the level of the most demanding application or user. That is required because an application has the information required to assume the identity of any of its users, and a compromise of any application compromises all users of that application. In a distributed environment, ensuring that all applications, their host computer systems, and network connections are secured to the required level is cost-prohibitive. The extreme case occurs with applications that are outside the enterprise boundaries. This is a nonscalable trust model.

Indirect Trust

Achieving scalable and cost-effective trust requires an indirect trust model. Indirect trust uses a third party, or parties, to assist in the authentication process. In this model, users and applications have a very strong trust relationship with a common third party, either directly or indirectly. The users and applications, or principals, trust that third party for verification of another principal's identity. The introduction of a third party reduces the geometric increase in complexity (shown in the previous section) to a linear increase in complexity (Exhibit 118.2).

All scalable distributed security systems use a trusted third party. In the Kerberos system, the trusted third party is known as the Key Distribution Center (KDC). In public key systems, the trusted third party is referred to as a Certificate Authority (CA). In token card systems, the token card vendor's server acts as a trusted third party. Many other applications of third-party trust exist in the world, one of the most obvious being credit cards, where the bank acts as the trusted third party between consumer and merchant. Neither consumer nor merchant shares a high degree of trust with each other, but both trust the credit card issuer. Note that without a credit card, each consumer would have to establish a direct trust relationship with each merchant (i.e., to obtain credit). Credit cards have made it much easier for consumers and merchants to do business, especially over long distances.

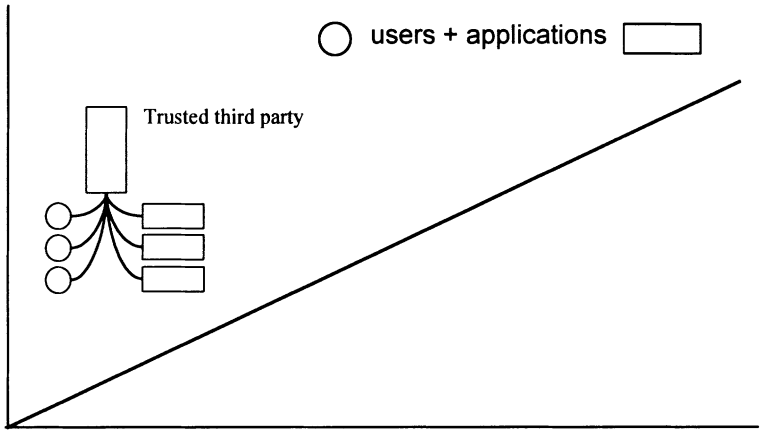


EXHIBIT 118.2 Indirect trust relationships.

Much like credit cards, a trusted third-party authentication system makes it easier for principals to do business — the first step of which is to verify each other's identity. In practical terms, that makes applications, information, and services more accessible in a secure manner. That benefits both consumers and providers of applications, information, and services, and reduces the cost to the enterprise.

Security Model

The manner in which a trusted third party provides proof of a principal's identity is a distinguishing characteristic of trusted third party security systems. This has a significant effect on all other services provided by the security system, as well as the scalability of the system. Kerberos uses a credential-based mechanism as the basis for identification and authentication. Those same credentials may also be used to carry authorization information. Kerberos credentials are referred to as "tickets."

Credentials

Requiring interaction with the trusted third party every time verification of identity needs to be done would put an onerous burden on users, applications, the trusted third party, and network resources. In order to minimize that interaction, principals must carry proof of their identity. That proof takes the form of a credential that is issued by the trusted third party to a principal. The principal presents that credential as proof of identity when requested.

All scalable distributed security systems use credentials. The Kerberos credential, or ticket, is analogous to an X.509 certificate in a public key system. These electronic credentials are little different conceptually than physical credentials, such as a passport or driver's license, except that cryptography is used to make the electronic credentials resistant to forgery and tampering. As with physical credentials, an electronic credential is something you can "carry around with you," without the need for you to constantly go back to an authority to reassert and verify your identity, and without the need for services to go back to that authority to verify your identity or the authenticity of the credential. Note that the use of a trusted third party for authentication does not imply the use of credentials. Token card systems are an example of trusted third-party authentication without credentials. The result of the authentication using such a card is a simple yes–no answer, not a reusable credential, and every demand for authentication results in an interaction with both the user and the token card server.

The stronger a credential, the stronger the assurance that the principal's claimed identity is genuine. The strength of a credential is dependent on both technology and environmental factors. Because a credential is carried by each principal, the credential must be tamper-proof and not forgeable. A credential's resistance to tampering and forgery is contingent on the strength of the cryptography used. Assurance of identity is contingent on the diligence of the trusted third party in verifying the identity of the principal's identity prior to issuing the credential. Assurance of identity is also contingent on the secure management of the credential

by the principal. As with physical credentials, electronic Kerberos credentials, and the information used to derive them must be protected, just as an individual's private key in a public key system must be protected.

As in the real world, all electronic credentials are not created equal. Simply possessing a credential does not imply universal acceptance or trust. As in the real world, the use and acceptance of a credential depends on the trust placed in the issuing authority, the integrity of the credential (resistance to forgery or tampering), and the purpose for which it is intended. For verification of identity, both passports and driver's licenses are widely accepted. A passport is typically trusted more than a driver's license, because the criteria for obtaining a passport are more stringent and a passport is more difficult to forge or alter. However, a passport says nothing about the holder's authorization or ability to operate a motor vehicle. A credential may also be single-purpose, such as a credit card. The issuing bank, as the trusted third party, provides protection to both the consumer and the merchant for a limited purpose: purchasing goods and services.

Credential Lifetime

As with physical credentials, the application and integrity of electronic credentials should limit the lifetime for which those credentials may be used. That lifetime may be measured in seconds or years, depending on the use of the credential. The strength of the cryptography that protects the integrity of the credential also effectively limits the lifetime of a credential. Credentials with longer lifetimes require stronger cryptography, because the credential is potentially exposed to attack for a longer period of time. However, cryptography is rarely the limiting factor in credential lifetime. Other issues, such as issuing cost and revocation cost, tend to be the determining factors for credential lifetime.

The distinguishing characteristic of credential-based systems is the lifetime of the credentials that they can feasibly accommodate. The longer the lifetime of a credential, the less often a new credential must be issued. However, the longer the life of a credential, the higher the probability that information embedded in the credential will change, or that the credential will be lost or stolen. The old "telephone book" revocation lists published by credit card companies is an example of the cost and complexity of revocation on a very large scale. Credit card companies have since moved to online authorization in order to lower costs and respond more rapidly.

Long-lived credentials reduce the credential-issuing cost but increase the credential-revocation cost. The shorter the lifetime of a credential, the more often a new credential must be issued. That increases the cost of the issuing process but reduces the cost of the revocation process. Credentials that are used only for authentication can have a relatively long lifetime. An individual's identity is not likely to change, and revocation would be necessary only if the credential was lost or stolen, or if the association between the individual and the issuing authority has been severed (e.g., such as when an employee leaves a company). Credentials that explicitly or implicitly carry authorization information generally require a shorter lifetime, because that information is more likely to change than identity information.

Different systems accommodate different lifetimes depending on the cost of issuing and revoking a credential and the intended use of the credential. While Kerberos credentials can have lifetimes of minutes or decades, they typically have lifetimes of hours or days. The process of constructing and issuing credentials is extremely efficient in Kerberos. That efficiency is key to Kerberos's ability to support authorization, capabilities, and delegation where new credentials may need to be issued frequently.

Capabilities

Credentials that carry authorization information are referred to as "capabilities," as they imply certain capabilities, or rights, upon the carrier of the credential. Kerberos supports capabilities by allowing authorization information to be carried within a Kerberos credential. As with other credentials, it is imperative that capabilities be resistant to tampering and forgery. We most often think of authorization information as coming from a central authorization service that provides commonly used information to various services (e.g., group membership information) where that information defines the limit of an individual's authorization. Kerberos supports this model by allowing authorization information from an authorization service to be embedded in a Kerberos credential when it is issued by the KDC; that authorization information is then available to services as a normal part of the Kerberos authentication process. Kerberos also supports a capability model based on "restricted proxies," in which the authorization granted to intermediate services may be restricted by the client.[11]

Delegation

There are also situations in which an individual authorizes another person to act on his behalf, thereby delegating some authority to that person. This is analogous to a power of attorney. Consider the simple example of a client who wants to print a file on a file server using a print server. The client wants to ensure that the print server can *print* (read) only the requested file, and not *write* on the file, or read any other files. The file server wants to ensure that the client really requested that the file be printed (and thus that the print server needs read-access to the file) and that the print server did not forge the request. The client should also limit the time for which the print server has access to the file, otherwise the print server would have access to the file for an indefinite period of time.

The extreme case is when an individual delegates unrestricted use of his identity to another person. As with an unrestricted power of attorney, allowing unrestricted use of another's identity can be extremely dangerous. (Obviously the authority that one individual can delegate to another must be limited by the authority of the delegating individual — we cannot allow an individual to grant authority they do not have, or the security of the entire system would crumble.) Unrestricted use of another's identity can also make end-to-end auditing much more difficult in many applications. Kerberos allows delegation of a subset of an individual's authority by allowing them to place authorization restrictions in a capability. The restricted proxy in Kerberos serves this function and is analogous to a restricted power of attorney. In the example above, the client would typically restrict the print server's right to read only the file that is to be printed using a restricted proxy. When the print server presents the resulting capability to the file server, the file server has all the information needed to ensure that neither the print server nor the client can exceed its authority, either individually or in combination.

In modern networks and business processes, it is common to find situations such as the above. Three-tier applications are another example. Here, the middle tier acts on the client's behalf for accessing back-end services. Delegation ensures the integrity and validity of the exchange and minimizes the amount of trust that must be placed in any intermediary. The need for delegation grows in significance as applications and services become more interconnected and as those connections become more dynamic. Without delegation, the identity and the rights of the originator, and the validity of a request, become difficult or impossible to determine with any degree of assurance. The alternative is to secure all intermediaries to the level required by the most sensitive application or user that makes use of the intermediary. This is cost-prohibitive on any but a very small scale.

Security Services

Many component security services are required to provide a complete distributed security service. The effectiveness of a distributed security system can be gauged by the component services it provides,[12] the degree to which those components operate together to provide a complete distributed security service, and the efficiency with which it provides those services.

Authentication

An authentication service permits one principal to determine the identity of another principal. The strength of an authentication service is the level of assurance that a principal's claimed identity is genuine. Put another way, the strength depends on the ease with which an attacker may assume the identity of another principal. For example, sending a person's ID and password across a network in the clear provides a very weak authentication, because the information needed to assume the identity of that person is readily available to any eavesdropper. Kerberos provides strong authentication by providing a high level of assurance that a principal's claimed identity is genuine. Kerberos also provides mutual authentication so that the identity of both client and service can be assured.

The reason for authentication is to ensure the identity of each principal prior to their conversing. However, without continuing assurance that their conversation has not been subverted, the utility of authentication alone is questionable. The Kerberos authentication protocol implicitly provides the cryptographic material, or "session keys," needed for establishing a secure channel that continues to protect the principal's conversation after authentication has occurred.

Secure Channels

A secure channel provides integrity and confidentiality services to communicating principals. Kerberos provides these services either directly through the use of Kerberos protocol messages, or indirectly by providing the cryptographic material needed by other protocols or applications to implement their own form of a secure channel.

Integrity

An integrity service protects information against unauthorized modification and provides assurance to the receiver that the information was sent by the proper party. Kerberos provides message integrity through the use of signed message checksums or one-way hashes using a choice of algorithms. Each principal in a Kerberos message exchange separately derives a checksum or hash for the message. That checksum or hash is then protected using a choice of cryptographic algorithms. The session keys needed for integrity protection are a product of the Kerberos authentication process.

Integrity applies not only to a single message, but to a stream of messages. As applied to a stream of messages, integrity also requires the ability to detect replays of messages. Simple confidentiality protection does not necessarily accomplish this. For example, recording and then replaying an encrypted message such as "Credit $100 to account X" several hundred times may achieve an attacker's goal without the need to decrypt or tamper with the message contents. The Kerberos protocol provides the mechanisms necessary to thwart replay attacks for both authentication and data.

Confidentiality

A confidentiality service protects information against unauthorized disclosure. Kerberos provides message confidentiality by encrypting messages using a choice of encryption algorithms. The session keys needed for confidentiality protection are a product of the Kerberos authentication process. Analysis based on message network addresses and traffic volume may also be used to infer information. An increase in the traffic between two business partners may predict a merger. Kerberos does not provide a defense against traffic analysis. Indeed, most don't since it is a very difficult problem.

Access Control

An access control service protects information from disclosure or modification in an unauthorized manner. Note that access control requires integrity and confidentiality services. Kerberos does not directly provide access control for persistent data, such as disk files. However, the Kerberos protocol provides for the inclusion and protection of authorization information needed by applications and operating systems in making access control decisions.

Authorization

An authorization service provides information that is used to make access control decisions. The secure transport of that authorization information is required in order to ensure that access control decisions are not subverted. Common mechanisms used to represent authorization information include access control lists (ACLs) and capabilities.

An ACL-based system uses access control lists to make access control decisions. An ACL-based system is built on top of other security services, including authentication, and integrity and confidentiality for distribution and management of ACLs. Kerberos does not provide an ACL-based authorization system but does provide all of the underlying services an ACL-based system requires.

Capability-based systems require the encapsulation of authorization information in a tamper-proof package that is bound to an identity. Capability-based authorization is a prerequisite to delegation in a distributed environment. Kerberos provides the facilities necessary for both capability-based authorization and delegation.

Non-Repudiation

Non-repudiation services provide assurance to senders and receivers that an exchange between the two cannot subsequently be repudiated by either. That assurance requires an arbitration authority that both parties agree to; presentation of sufficient and credible proof by the parties to the arbitrator; and evaluation of that proof by the arbitrator in order to settle the dispute. For example, in the case of an electronic funds transfer between two business entities, a court of law would be the arbitrator that adjudicates repudiation-based disputes that arise between the two businesses.

The technological strength of a non-repudiation service depends on the resistance to tampering or falsification of the information offered as proof and the arbitrator's ability to verify the validity of that information. Resistance to tampering or falsification must be sufficient to prevent modification of the proof for as long as a dispute might arise. Although Kerberos offers the basic authentication and integrity services from which a non-repudiation service could be built, the effectiveness of that service will depend on the required strength of the service, and it is dependent on what technologies are incorporated into a Kerberos implementation and the management of the implementation.

The symmetric-key cryptography as used by basic Kerberos implementations is generally not sufficient for non-repudiation, because two parties share a key. Since that key is the basis of any technical proof, either party in possession of that key can forge or alter the proof. If augmented with strict process controls and protection for the KDC, symmetric-key cryptography may be acceptable. However, that process control and protection can be quite expensive. (Note that banks face this issue with the use of PINs, which use symmetric-key cryptography; and the fact that two parties share that key — the consumer and the bank — is rarely an issue, because the bank provides sufficient process controls and protection for management of the PIN.) Kerberos does not offer the arbitration services that are required for the complete implementation of such a service.

Availability

Availability services provide an expected level of performance and availability such as error-free bandwidth. Perhaps the best example of an availability problem is a denial-of-service attack. Consider someone simply disconnecting the cable that connects a network segment to its router. Kerberos does not offer any services to deal with this set of problems. Distributed security systems generally do not offer availability services.

Functional Overview

The ultimate objective of any Kerberos user is to gain access to application services. The process by which that occurs involves several steps, the last step being the actual authentication between the user and the application service. A key part of that process involves the trusted third party in the Kerberos system, the Kerberos security server (KDC). Although descriptions of that process correctly focus on the interaction between users and the KDC, one of the key design elements of Kerberos is the ability for clients and services to securely interact, with little or no involvement of the KDC.

Kerberos is a trusted third-party, credentials-based authentication system. The KDC acts as the trusted third party for humans and services, or principals that operate on client or server computer systems. Kerberos principals authenticate with one another using Kerberos credentials, or tickets. These tickets are issued to principals by the KDC. A client principal authenticates to a service principal using a ticket. The Kerberos security server is not directly involved in that client–service authentication exchange. The result of an authentication exchange between a client and service is a shared session key that can be used to protect subsequent messages between the client and the service.

Components

The primary components of a Kerberos system are the client and server computer systems on which applications operate, and the Kerberos security server (KDC.) In addition to those physical components, there are a number of additional logical components and services that make up the Kerberos system, such as the authentication service and the principals that make use of Kerberos services.

KDC

The keystone of the Kerberos system is the Kerberos security server, generally referred to as the "KDC," or Key Distribution Center. Although the term KDC is not an accurate description of all the services provided, it has stuck. The KDC is the trusted third party in the Kerberos distributed security system. The KDC provides authentication services, as well as key distribution and management functions. There may be multiple KDCs, depending on the level of service and performance that is required. The KDC consists of a set of services and a database that contains information about principals.

Principal

The entities to which the KDC provides services are referred to as "principals." Principals share a very high degree of trust with the KDC. They may be human or may represent a service or a machine. Every principal has an identifier that is used by the KDC to uniquely identify a human or service and allow one principal to determine the identity of another during the Kerberos authentication process. Depending on the cryptographic mechanisms used, a principal may also share a secret key with the KDC, thus the high level of trust required between principals and the KDC.

The primary difference between human and service principals results from the available means for storing the password, or key, and the persistence of that key. A person can securely carry a password in his head, whereas services cannot. Services that use shared secrets for authentication require access to a key. Unlike keys that are used by humans — which are typically derived from a password — service keys are typically random bit strings. If unattended operation for services is required, that key must be kept in persistent storage that is accessible to the service. That key storage is referred to as a "key table" and is generally kept in a file on the host computer system on which the service operates. Key tables may contain keys for multiple services, or may be unique to a service. The security of key tables is dependent on the host computer system's security. This is identical to the problem of protecting private keys in public-key or asymmetric-key systems. More secure solutions for protection of key tables require tamper-proof hardware such as a smart card.

The most significant functional difference between a client and a service results from the difference in key persistence. Kerberos clients do not maintain the user's key in any form beyond a very short period of time during the initial authentication process. However, services always have ready access to their key in the key table. The result is that clients generally can only initiate communications, whereas services may either initiate or accept communications (i.e., a service may also act as a client).

Ticket

A ticket is part of a cryptographically sealed credential issued by the KDC to a client. A ticket, along with other confidential information, allows a client to prove their identity to a service, without the client and service having any preestablished relationship. A ticket is specific to a client–service pair. That is, a ticket specifies both a client principal and the service principal: the client principal to whom the ticket was issued, and the service principal for which it is intended. A client may reuse tickets. Once a client obtains a ticket for a service, subsequent authentication of the client to the service does not require involvement of the KDC.

Realm

The KDC logically consists of a set of services and a database that contains information about principals. In Kerberos that collective is referred to as a "realm," and the authentication service within the KDC is the trusted third party for all principals in the realm. Realms may be defined based on either security requirements in order to separate domains of trust, or as an administrative convenience for grouping principals. Some implementations allow a single KDC to serve multiple realms to reduce the number of physical systems needed. Principals in different realms can interact using "cross-realm" (sometimes referred to as "inter-realm") authentication. Cross-realm authentication generally requires prior agreement between the administrators of the different realms.

Principal Identifier

Kerberos defines several principal identifier forms, including a native Kerberos form, as well as an X.500 distinguished-name form. We describe only the native Kerberos name form here. Simple principal identifiers take the form name@REALM. Principal identifiers are case sensitive. By convention, the realm name is the DNS domain name in upper case. For example, hanley@Z.COM refers to the principal named hanley in domain

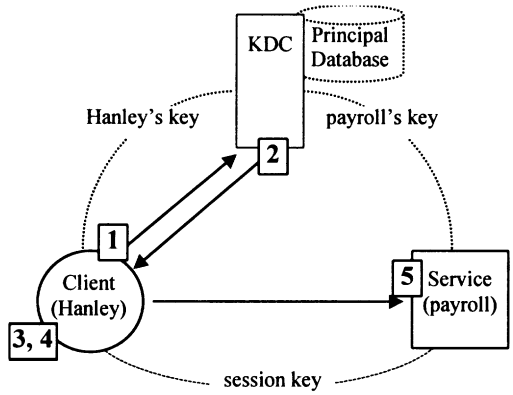


EXHIBIT 118.3 Basic Kerberos authentication.

z.com. Principal identifiers may also contain an instance. Instances are typically used only for service principals (discussed later in this chapter).

Authentication

The simplest and most basic form of the Kerberos protocol performs authentication using a shared secret and symmetric-key cryptography: the user and KDC share a secret key, and the service and KDC share a secret key. However, the user and service do not share a secret key. Providing the ability for a user and service to authenticate, and establish a shared secret, where none previously existed, is the fundamental purpose of the Kerberos protocol.

For this basic form of Kerberos authentication to work, users and services must first share a secret key with the KDC. Methods for first establishing that shared secret vary. The steps of the basic authentication process are discussed below and shown in Exhibit 118.3.

1. A user, or more precisely, Kerberos client software on the user's work station acting on behalf of the user, prompts the user for his ID. The client then sends that ID to the KDC as an assertion of the user's identity, along with the name of a service that the client wishes to access (for example, "I'm Hanley and I want access to the payroll service").
2. The authentication service (AS) of the KDC receives that request, constructs a reply, and sends that reply to the client.
   2.1. The AS checks to ensure that the requesting client (Hanley) and service (payroll) principals exist in the principal database maintained by the KDC. Assuming they exist, the AS constructs a "service ticket" for the requested service (payroll) and places the user's principal name (Hanley) into that service ticket.
   2.2. The AS then generates a random key, referred to as the "session key."
   2.3. The AS then places the session key into the service ticket. The service ticket is then encrypted, or "sealed," using the service's key, obtained from the principal database. That service key is a secret key the (payroll) service shares with the KDC. That key is held in the principal database, as well as by the service.
   2.4. The AS constructs the client part of the reply and places the same session key (from step 2.2) into the client part of the reply. The client part of the reply is then encrypted using the user's key, obtained from the principal database. That is, the secret key (i.e., password) the user (Hanley) shares with the KDC. That key is held in the principal database, as well as by the user.
3. The client receives the reply from the AS, and prompts the user for his password. That password is then converted to a key, and that key is then used to decrypt, or "unseal," the client part of the reply from the AS (from step 2.4).

   If that decryption succeeds, then the password/key entered by the user is the same as the user's key held by the KDC (i.e., the key used to encrypt the client part of the reply). The decryption process also exposes the session key placed into the reply by the AS (from step 2.4). Note that the client cannot

tamper with the service ticket in the reply, because it is encrypted, or "sealed," using the service's key, not the client's key.

If the decryption does not succeed, then the password the user entered is incorrect, or the real AS did not issue the reply, or the user is not who he claims to be. In any case, the information in the AS' reply is useless because it cannot be decrypted without the proper password/key, and the process ends.

The following steps assume that the decryption process succeeded. Note that the AS has no knowledge of whether or not the decryption process on the client succeeded.

4. When the client (Hanley) wishes to authenticate to the service (payroll), the client constructs a request to the service. That request contains the service ticket for the payroll service issued by the AS (from step 2.3).
5. The service receives the request from the client, and uses its service key to decrypt the ticket in the request, i.e., the key that is the shared secret between the (payroll) service and the KDC, and that was used to encrypt the service ticket by the AS (from step 2.3).

   If the decryption succeeds, the service's key and the key that the ticket is encrypted in are the same. Because the KDC is the only other entity that knows the service's key, the service knows that the ticket was issued by the KDC, and the information in the ticket can be trusted. Specifically, the client principal name placed into the ticket by the AS (from step 2.1) allows the service to authenticate the client's identity. The decryption process also exposes the session key placed into the service ticket by the AS (from step 2.3).

   If the decryption fails, then the ticket is not valid. It was either not issued by the real AS, or the user has tampered with the ticket. In any case, the ticket is useless because it cannot be decrypted, and the process ends.

At this point, the service (payroll) has proof of the client's identity (Hanley), and both the client and the service share a common key: the session key generated by the AS (from step 2.2), and successfully decrypted by the client (from step 3) and by the service (from step 5). That common session key can then be used for protecting subsequent messages between the client and the service. Note that once the ticket is issued to the client, there is no KDC involvement in the authentication exchange between the client and the service. Also note that the user's password/key is held on the work station, and thus exposed on the work station, only for the period of time required to decrypt the reply from the KDC.

A thief could eavesdrop on the transmission of the reply from the KDC to the client. However, without the user's key, that reply cannot be decrypted. A thief could also eavesdrop on the transmission of the service's ticket. However, without the service's key, that ticket cannot be decrypted. Without knowledge of the user's or service's keys, the attacker is left with encrypted blobs that are of no use. There are other more sophisticated attacks that can be mounted, such as a replay attack, and there are other countermeasures in Kerberos to help thwart those attacks; those attacks and countermeasures are discussed in subsequent sections.

Credentials Caching

The authentication exchange described above allows a client and service to securely authenticate and securely establish a shared secret — the session key — without requiring a preestablished secret between the client and service. While those are useful and necessary functions of any distributed authentication service, it requires that the user obtain a service ticket each time access is required to a service. It also requires that the user enter a password each time a service ticket is obtained in order to decrypt the ticket. This behavior would obviously not be a very efficient use of people's time or network bandwidth.

A simple additional step to cache credentials — that is, the service ticket and session key — would allow the reuse of credentials without having to constantly go back to the AS or requiring user involvement. A "credentials cache" on the client serves this purpose, and all Kerberos implementations provide a credentials cache. Thus, as the user collects service tickets during the day, they can be placed into the credentials cache and reused. This eliminates involvement between the user and the AS when the same service is accessed multiple times. Note that a client requires both a ticket and the ticket's associated session key (a credential) to make use of a ticket. Thus the term "credentials cache," and not "ticket cache."

Kerberos can also limit the usable life of credentials by placing an expiration time into the ticket when the AS constructs the ticket. The ticket expires after that time, and the user must go back to the AS to obtain another ticket. While Kerberos tickets can have virtually any lifetime, the typical lifetime of a Kerberos ticket is the average workday.

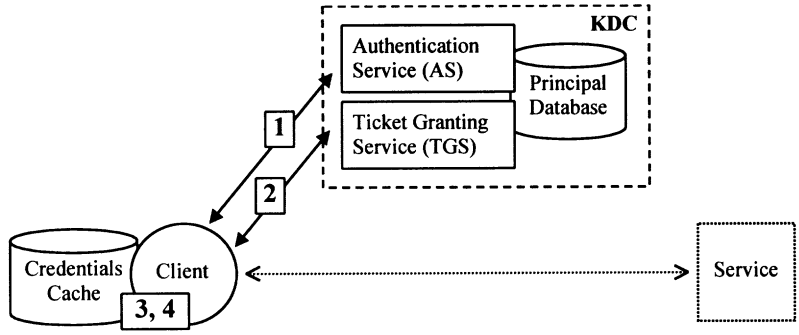


EXHIBIT 118.4 Authentication and ticket-granting services.

Ticket-Granting

Even with credentials caching, interaction between the user and the authentication service (AS) would still be required every time the user wants another ticket. For environments in which a user may access dozens of services during the day, this is unacceptable. One possible solution would be to cache the user's password in order to obtain service tickets without user interaction. However, that exposes the user's password to theft by rogue client software. Note that rogue software could also steal credentials from the credentials cache. However, those credentials will typically expire after a day or less. So, while a thief may have a day's fun with stolen credentials, at least the thief does not get indefinite use of the user's identity. Thus, we can limit the duration of such a compromise to the lifetime of the credentials. The ability to limit a compromise in both space and time is an extremely important attribute of a distributed security system. However, if the user's password is stolen, it is much more difficult to limit such a compromise.

The solution to this problem builds on the three parts that we already have: the authentication service (AS), which can issue tickets for services to clients; the credentials cache on the client that allows reuse of a ticket; and the ability to authenticate a user to a service using an existing credential. Using those components, we can then build a service that issues tickets for other services, much like the AS. However, our new service accepts a ticket issued by the AS, instead of requiring interaction with the user.

Our new service is known as the "ticket-granting service," or TGS. The TGS operates as part of the KDC along with the authentication service (AS) and has access to the same principal database as the AS. We have not dispensed with the AS, but the primary purpose of the AS is now to issue tickets for the TGS. A ticket issued by the AS for the TGS is known as a "ticket-granting ticket," or TGT. Using that ticket-granting ticket (TGT), a client can use the ticket-granting service (TGS) to obtain tickets for other services, or "service tickets." Thus, for example, instead of asking the authentication service (AS) for a ticket for the payroll service, the client first asks the AS for a ticket-granting ticket (TGT) for the ticket-granting service (TGS); then, using that TGT, asks the TGS for a service ticket for the payroll service. Although that introduces an additional exchange between the client and the KDC, it typically need be done only once at the beginning of the workday (see Exhibit 118.4).

By using the AS only once at the beginning of the day to obtain a TGT, and then using that TGT to obtain other service tickets from the TGS, we can make the entire operation invisible to the user and significantly improve the efficiency and security of the process. Thus, the behavior becomes:

1. The first action of the day is to obtain a TGT from the AS as previously described (e.g., providing an ID and password). Only, instead of the user specifying the name of a service, the client automatically requests a ticket for the TGS on behalf of the user.
2. The TGT and session key returned by the AS from the prior step is placed into the credentials cache, along with the TGT's session key.
3. When a service ticket is needed, the client sends a request to the TGS (instead of to the AS). That request includes the TGT and the name of the service for which a ticket is needed. The TGS authenticates the client using the TGT just like any other service and, just like the AS, constructs a service ticket for the requested service and returns that ticket and session key to the client.

4. The service ticket and session key returned from the TGS is placed into the credentials cache for reuse. The client may then contact the service and authenticate to the service using that service ticket.

A TGT is identical to any other service ticket and is simply shorthand for "a ticket for the TGS." The AS and TGS are virtually identical, and both can issue tickets for any other service. The primary difference between the AS and TGS is that the TGS uses a TGT as proof of identity, whereas the AS can be used to issue the first, or "initial" ticket. The proof the AS requires before that initial ticket is issued to a user can involves forms that are not a Kerberos ticket, such as a token card, smart card, public key X.509 certificate, etc. Those various forms of proof are referred to as "preauthentication." Subsequent sections describe the AS and TGS exchanges, the client–service exchanges, and preauthentication in greater detail.

Functional Description

This section builds on the previous discussions and provides a description of both the Kerberos protocol and the interaction of various components in a Kerberos system. Application of the protocol to solve various distributed security problems is also used to illustrate concepts and applications of the protocol. This description is not definitive or complete, and there are many details that have been omitted for clarity and brevity. For a complete description of the protocol, the official standard, Internet RFC 1510, should be consulted.

Initial Authentication

The Kerberos initial authentication process is the point in time when an individual proves his identity to Kerberos and obtains a ticket-granting ticket (TGT). Typical implementations integrate the initial authentication process with the host OS log-in, providing a single point of authentication for the user each morning. A variety of technologies can be brought to bear at this point, depending on the level of assurance that is needed for an individual's identity. Once initial authentication is completed, the TGT obtained as a result of that initial authentication can be used to obtain service tickets from the ticket-granting service (TGS) for other services. Those service tickets are the basis for client–service authentication, as well as the establishment of the keys needed to subsequently protect client–service interactions.

The simplest form of initial authentication uses an ID and password, as previously described:

1. The client asserts its identity by sending a Kerberos principal name to the KDC. The client sends no proof of its identity at this time. To put it another way, the proof offered by the client at this time is null.
2. The KDC then constructs a TGT and a reply that is encrypted in the user's key. That key is derived from the user's password and is a shared secret between the user and the KDC.
3. The KDC then sends the (encrypted) reply with the TGT back to the client.
4. The client receives the reply from the KDC, then prompts the user for his password and converts the password to a key. That key is then used to decrypt the reply from the KDC.
5. If the reply from the KDC decrypts properly, the user has authenticated. If the reply does not decrypt properly, the password provided by the user is incorrect.

Note that authentication actually occurs on the client, and the KDC has no knowledge of whether or not the authentication was successful. The KDC can infer that the authentication was successful only if the client subsequently uses the TGT that is part of the reply to obtain a service ticket. The drawback of this approach is that anyone can make a request to the KDC asserting any identity, which allows an attacker to collect replies from the KDC, and subsequently mount an offline attack on those replies. The Kerberos preauthentication facility can be used to help thwart those attacks.

Preauthentication

The term "preauthentication" is used to describe an exchange in which the user sends some proof of his identity to the KDC as part of the initial authentication process. If that proof is unacceptable to the KDC, the KDC may demand more, or alternate, preauthentication information from the client, or may summarily reject or ignore the client. In essence, the client must authenticate prior to the KDC issuing a credential to the client; thus the term "preauthentication." The proof of identity used in preauthentication can take many forms and

is how most technologies such as smart cards and tokens are integrated into the Kerberos initial authentication process.

What technologies are used depends on the level of assurance required for a user's identity and is typically associated with a user (or a role performed by a user). For example, Kerberos administrators might be required to use two-factor authentication, whereas a simple ID and password would suffice for other users. Implementations vary in the types of preauthentication they support. Preauthentication data may include a digital signature and an X.509 public key certificate; token card data; challenge–response; biometrics information; location information; or a combination of different types of those preauthentication data.

Preauthentication may require several messages between the client and KDC to complete the initial authentication process. For example, the challenge–response exchange used for some token cards may require additional messages for the challenge from the KDC and the response from the client. Only the simplest form of preauthentication is described here. The simplest form of preauthentication uses an ID and password, and an encrypted timestamp:

1. The client prompts the user for his principal ID and password, and converts the password to a key.
2. The client then obtains the current time and encrypts that (along with a random confounder), attaches its principal ID, and sends the request to the KDC.
3. If the KDC can decrypt the timestamp in the request from the client, it has some proof that the user is who he says he is. The KDC may also require that the timestamp be within certain limits.

After this point the process is the same as the simple (nonpreauthentication) exchange. Note that this approach affords greater protection by making it more difficult for an attacker to obtain a TGT for other users or otherwise attack a captured TGT.[13] However, an offline attack may still be mounted against replies sent from the KDC to other users that are sniffed off of the network. Thus, good passwords are still as important as ever, and most Kerberos implementations provide facilities for password policy enforcement to minimize the risk of weak passwords.

KDC–Client Exchanges

The exchanges used for initial authentication with the AS and the subsequent exchanges used to obtain service tickets with the TGS, are both built from the same basic mechanism. In this section we also identify the message names that Kerberos uses for the various requests and replies.

1. The client sends an authentication request (AS-REQ) message to the authentication service. In that request, the client specifies that it wants a ticket for the TGS.
2. The AS sends a ticket-granting ticket (TGT) back to the client in an AS reply (AS-REP) message. That TGT is simply a service ticket for the TGS. The AS-REP contains both the TGT and the session key required in order for the client to use that TGT.
3. When the client wants a service ticket for another service, it requests a ticket from the TGS by placing the TGT into a TGS request (TGS-REQ) message. The TGS sends a service ticket for the requested service back to the client in a TGS reply (TGS-REP) message. The TGS-REP contains both the service ticket and the session key required for the client to use that service ticket.

Again, a TGT is functionally no different than any other ticket. Nor is the TGS conceptually any different than any other service. The only reason for using a special TGS-REQ message to talk to the TGS is to codify the conventions used by the ticket-granting service and optimize the protocol. However, if you look closely at the AS-REQ and TGS-REQ messages, they are very similar and are sometimes referred to collectively as a KDC request (KDC-REQ) message. The same is true of the AS-REP and TGS-REP messages, which are collectively referred to as a KDC reply (KDC-REP) message.

Initial Tickets

Although the primary purpose of the AS is to issue TGTs, the AS may issue tickets for any service, not just TGTs for the TGS. The only real difference between tickets issued by the AS and tickets issued by the TGS are that tickets obtained from the AS are marked as "initial" tickets; tickets obtained from the TGS (using a TGT) are not marked "initial." Initial tickets can be useful if an application wants to ensure that the user obtained the ticket from the AS (i.e., the client went through initial authentication to obtain the service ticket) and did not obtain the service ticket using a TGT. For example, the change-password service requires that the user

obtain an initial ticket for the change-password service. This requires that the user enter his password to obtain a ticket that is marked initial (i.e., a ticket that the change-password service will accept). A ticket for the change-password service obtained from the TGS using a TGT will not be marked initial and will be rejected by the change-password service. This precludes the use of a stolen TGT to change a user's password, or someone using an unlocked work station to change the work station user's password using a cached TGT.

Ticket Construction

Every ticket adheres to the same basic format and contains the same basic information. That information includes the name of the client principal, the name of the service principal, the ticket expiration time, and a variety of other attributes and fields. When a client requests a ticket for a service, the reply from the KDC contains the service ticket, encrypted in the key of that service. Most of the information in the service ticket is also exposed to the client as part of the reply. That information is provided to the client so that the client can ensure that what it received is what the client requested.

The KDC may also provide defaults for various fields in the ticket, which the client did not specify, but which the client may need to know. For example, each ticket has a lifetime; the client may or may not specify the ticket lifetime in a request. If the client does not specify a lifetime, the KDC will provide a default value. The KDC may also enforce maximum values for various fields. For example, if the sitewide maximum ticket lifetime is eight hours, the KDC will not issue a ticket with a lifetime longer than eight hours, regardless of what the client requests. Knowing the lifetime of a ticket is important for a client so that if the ticket is expired, a new ticket can be requested automatically from the TGS without user involvement. For instance, long-running batch jobs.

Most implementations also allow each service to specify a maximum ticket lifetime, and the KDC will limit the lifetime of a ticket issued for a service to the service-defined maximum. Some services, such as the change-password service, typically have maximum ticket lifetimes that are very short (e.g., ten minutes), with the objective being to make those tickets "single use." Most password-change clients also do not cache such tickets, because holding on to them would be of no value.

Client–Service Exchanges

The authentication exchange that occurs between a client and a service is conceptually similar to the client–KDC exchanges. However, the messages used are different to accommodate specific needs of client–service authentication and to eliminate information that is required only for client–KDC exchanges. The messages used for client–service application authentication are collectively referred to as the application (AP) or client–server (CS), messages.

In the following example, we assume that the client already has a service ticket in its credentials cache and, if not, the client will obtain the required service ticket prior to beginning this exchange.

1. The client constructs an application request (AP-REQ) message and sends it to the service. The AP-REQ contains the service ticket as (previously issued by the KDC and stored in the credentials cache as part of a client-TGS exchange). The AP-REQ also contains an authenticator. The authenticator contains various information, including a time-stamp, and may be used by the service to ensure that the AP-REQ is not a replay. The client encrypts the authenticator, and some other information in the AP-REQ, with the session key that is associated with the service ticket (obtained originally from the KDC as part of the TGS-REP).
2. The service receives the AP-REQ and decrypts the ticket in the AP-REQ using its own service key. This exposes the information in the service ticket, including the client's identity, various flags, and the random session key generated by the KDC when the KDC issued the service ticket to the client. After this decryption process is completed, both the client and service are in possession of a common key: the random session key generated by the KDC when the service ticket was originally constructed and issued to the client by the KDC.
3. The session key obtained in the previous step is used to decrypt the authenticator. The authenticator contains information that allows the service to ensure that the AP-REQ message is not a replay. The authenticator may also contain a "subsession" key (see below).

4. If the client requests mutual authentication, the service is obliged to reply to the client with an application reply (AP-REP) message that is encrypted in either the session key from the ticket or a subsession key. The AP-REP allows the client to validate the identity of the service.

Other provisions of the AP-REQ and the AP-REP allow for the establishment of initial sequence numbers for data message sequencing, and the establishment of a new subsession key that is independent of the session key in the service ticket (which was generated by the KDC). Either the client or the service can generate a new subsession key. This allows a fresh session key, unknown to the KDC, to be used for every session between the client and the service.

Confidentiality and Integrity

Once the appropriate session keys are established, the Kerberos "safe" (SAFE) messages can be used for integrity protection, and "private" (PRIV) messages can be used for confidentiality protection. Those messages also provide for additional protection using sequence numbers, timestamps, and address restrictions (discussed later in this chapter). Alternatively, the application may choose to use its own form of integrity and confidentiality protection for data. For example, an IPSec (Internet Protocol Security) implementation could use the basic AP-REQ and AP-REP exchange to establish the keys for two end points, where the end points are network stacks or systems, instead of a human and a service.

TGS AP-REQ

Examination of the protocol will show that an AP-REQ is also used in the TGS request (TGS-REQ). The AP-REQ is the client's way of authenticating and securely communicating with a service, and the TGS is simply another service, albeit with special capabilities. The AP-REQ used to authenticate to the TGS contains the TGT (the service ticket for the TGS), just as any AP-REQ for any service. Because the TGS-REQ requires more than just an AP-REQ, the AP-REQ in the TGS-REQ is carried in a preauthentication element of the TGS-REQ.

Replay Protection

Replay protection ensures that an attacker cannot subvert the system by recording and replaying a previous message. As mentioned previously, confidentiality and integrity protection alone do not protect against replay attacks. Kerberos can use timestamps or a form of challenge response, to protect against replay attacks. The type of replay detection that is appropriate depends on whether a datagram-oriented protocol, such as UDP/IP, or a session-oriented protocol, such as TCP/IP, is used. Note that all protocols that provide replay protection will have mechanisms and requirements similar to those described here, regardless of the type of cryptography that is used.

Timestamps

Replay protection using timestamps is most suited to datagram- or transaction-oriented protocols and requires loosely synchronized clocks based on a secure time service and the use of a "replay cache" by the receiver. A replay cache is simply a cache of messages previously seen by the receiver, or more likely, a hash of each of those messages. The receiver must check each received message against the replay cache to determine if the message is a replay. Note that the replay cache must be maintained in persistent storage if replay detection is to survive a restart of the service.

Obviously, the replay cache could grow forever unless it is bounded in some manner. Timestamps help to limit the size of the replay cache. By defining a bounded window of time for the acceptance of messages, the replay cache can be limited to messages that are received within that window. A service will summarily reject any message with a timestamp outside of that window, and messages outside that window can be discarded from the cache. Thus, the replay cache must be checked only for messages that fall within that window, and the size of the replay cache can be limited to messages received within that window.

That window of time over which the replay cache must operate is referred to as the acceptable "clock skew." Clock skew represents the maximum difference that is allowable between the clocks of two different systems. If the systems' clocks differ by more than the clock skew, all messages will be rejected. A typical value for clock skew is five minutes. Smaller clock skew values require closer synchronization of system clocks but reduce the overhead of maintaining and checking the replay cache. Larger clock skew values allow looser synchronization of system clocks, but increase the overhead of maintaining and checking the replay cache.

Datagram- or transaction-based applications must deal with duplicate, dropped, and out-of-sequence messages as a normal network occurrence. Thus, well-behaved datagram- or transaction-based applications should already have mechanisms for replay detection within the application, regardless of security considerations. If those applications protect their messages using Kerberos confidentiality or integrity services, there is usually no need to use Kerberos replay protection for the application data. Although Kerberos can provide the necessary replay protection "out of the box" for those applications, the applications should be examined to ensure that the protection provided by Kerberos is not redundant and does not add unnecessary overhead.

Challenge–Response

Replay protection using a challenge–response exchange is most suited to session-oriented protocols, such as TCP/IP. The subsession key facility within the Kerberos AP-REQ and AP-REP messages provides a means to effect the challenge–response exchange. Challenge–response eliminates the requirement for clock synchronization between the client and the service, and the need for the service to maintain and check a replay cache. However, challenge–response adds an additional message from the service back to the client. Thus, challenge–response is typically suitable only for session-oriented communications where the cost of the messages can be amortized over an entire session, or where those messages can be piggybacked on the application's normal session-initiation messages. Individual messages within the session must then be protected using sequencing and confidentiality or integrity to ensure that the messages within the session are not subject to replay attacks. Mechanisms similar to what are described here can also be used to minimize the need for clock synchronization between clients and the KDC.

Making use of the subsession key facility within the AP-REQ and AP-REP messages requires mutual authentication. Challenge–response also requires that the service respond with a new random subsession key in the AP-REP for each AP-REQ. In effect, the new random subsession key in the AP-REP generated by the service is the challenge. The client's ability to subsequently decrypt the AP-REP, extract the new subsession key, and protect subsequent messages to the service using that subsession key provide proof that the AP-REQ was not a replay and serves as the client's response to the service's challenge.

Note that the service cannot verify that the client has passed the challenge until the service receives the first data message from the client to the service protected by the subsession key. Thus, the client is technically not authenticated to the service until the first data message from the client is successfully received and decrypted by the service. By the same token, the service is technically not authenticated to the client until the first data message from the service in reply to the client is received and decrypted by the client (the AP-REP from the service could be a replay to the client). Whether that technical issue is a security issue depends on the behavior of the client and server. If the client or service engage in a significant and irreversible act prior to the completion of authentication on both sides, damage could result. Generally however, the worst that can happen is a denial-of-service attack that is difficult to diagnose.

Session Keys

Tickets may be sniffed off the network by an attacker during client–KDC or client–service exchanges. Thus, a ticket alone is insufficient to prove the identity of the client principal name embedded in a ticket or the right of the holder to use that ticket. The session key associated with a ticket provides the additional information necessary for that proof. Every ticket issued by the KDC has a unique session key (unless a client specifically requests otherwise). A Kerberos credential is a ticket and the associated session key. The following sections review the role session keys play in the various exchanges.

Authentication Service

During the initial authentication exchange, the client uses the key derived from the user's password to decrypt the reply (the AS-REP message issued by the AS). That reply, as do all KDC replies, contains a ticket (in this case, the TGT returned by the AS). When the client decrypts that reply, the decryption exposes a session key. All requests and replies between the client and the TGS from that point onward are protected using that session key from the AS-REP. Using the session key that results from the initial AS exchange eliminates the need to store the user's key in any form on the work station. That is, once the initial authentication exchange between the client and the AS is completed, subsequent exchanges use the session key returned by that exchange and not the key derived from the user's password. The TGT, as with any ticket, is sealed with the service key of the service for which the ticket is intended, which in this case is the TGS. The client typically places the TGT and the TGT's session key into a credentials cache for future use.

Ticket-Granting Service

When the KDC builds a TGS reply (TGS-REP), it first constructs a ticket for the requested service. As part of that construction process, the KDC generates a random session key that is placed into the ticket. The KDC then encrypts that ticket in the service's key (the key it shares with the service.) That ticket is then placed into the reply (TGS-REP) to the client, with the ticket ultimately destined for the service. That same random session key is also placed into the reply destined for the client. The reply is then encrypted with the session key associated with the TGT in the client's request to the TGS (TGS-REQ). When the construction of the reply (TGS-REP) is completed by the KDC, we have: (1) a service ticket containing the session key; (2) that service ticket encrypted in the service's key; (3) a reply containing the same session key; and (4) that reply encrypted in the session key associated with the TGT.

When the reply is received and decrypted by the client — using the TGT's session key — one copy of the ticket's session key, along with other relevant information about the ticket, is exposed to the client. The other copy of the session key, along with most of the same information exposed to the client, is still sealed in the service ticket. The content of that service ticket is not accessible to the client, because it is encrypted in the service's key (the key the service shares with the KDC), which is not known to the client. That prevents the client from tampering with the information in the ticket. The client typically places the ticket, along with the other ticket information, including the session key for that ticket, into a credentials cache for future use.

Client–Service Exchanges

Session keys play the same role in the client–service exchange as they do in the client–KDC exchanges. The authenticator constructed by the client as part of the application request (AP-REQ) message is encrypted using the session key associated with the service ticket. That same session key is accessible to the service when the service decrypts the service ticket using its own service key. That session key from the service ticket is then used to decrypt (and thus validate) the authenticator.

Cross-Real Authentication

A realm typically defines a collective trust, or common security domain. Obviously there are limits to the size of such a domain both in manageability and in the collective and common trust that domain represents. For example, collective or common trust usually drops precipitously at enterprise boundaries, and sometimes at organizational boundaries within an enterprise. However, it is often the case that those various domains, or realms, must still communicate securely.

Between realms, Kerberos provides cross-realm authentication services. Cross-realm authentication allows principals in one realm (e.g., clients) to authenticate with principals in another realm (e.g., services). Conceptually, cross-realm authentication treats each realm in the path between a client and a service as simply another service. The client's realm effectively issues a ticket for the ticket-granting service (TGS) in the service's realm; that ticket is referred to as a cross-realm or inter-realm TGT. For example, a client in realm X accessing a service in realm Y first goes to a KDC in realm X to obtain a cross-realm TGT for realm Y; that TGT is then presented to a KDC in realm Y in order to obtain a service ticket for the end, or "target" service.

Cross-realm authentication requires prior agreement between the administrators of the two realms in order to establish the keys on the respective KDCs. Those keys effectively allow one realm to issue cross-realm TGTs that will be honored by the other realm. As with other services, possession of a ticket does not ensure right of access; access is ultimately determined by the service and not the issuing realm or KDC. The trust established between realms for cross-realm authentication lies in the promise that the realms will not lie about the identity of their respective clients. The ability to issue a cross-realm TGT is not necessarily bilateral; this allows one-way cross-realm authentication, although this feature is rarely used.

The client may collect cross-realm TGTs obtained during cross-realm authentication, just as any other tickets, and hold them in its credential cache for reuse. Once the client obtains the cross-realm TGT for the target realm, the client can request tickets from the target realm's TGS directly, just as the client would request tickets directly from the TGS in its own realm. Once the client obtains the ticket for the target realm's TGS, the client–service authentication process is identical to the client–service authentication process within a single realm. Thus, cross-realm authentication between a client and any service in the other realm requires that the additional cross-realm authentication steps be performed only once. For example, given realms X and Y, where the realm administrators have previously established a cross-realm relationship, a client in realm X that wants to get to a service in realm Y must first obtain a cross-realm TGT from a KDC in realm X for realm Y. That

cross-realm TGT may then be used to get a ticket from a KDC in realm Y for a service in realm Y and the KDC in realm X does not participate in the latter step.

Any number of realms can have a direct, or pair-wise, cross-realm relationship, in which case a client goes directly between those realms as described above. Where many realms are involved, direct relationships between every pair of realms can be a significant management overhead for establishing all of the necessary cross-realm keys. For example, with ten realms, a direct relationship between every pair of realms requires that each realm maintain nine pairs of cross-realm keys (a key pair assumes a bilateral relationship), for a total of 90 cross-realm key pairs. Although this is manageable for a relatively small number of realms, such as one might find within an enterprise, it becomes unmanageable for a large number of realms. Note that this is the geometric trust complexity problem discussed earlier.

To reduce the complexity of cross-realm key management, realms may also be arranged in transitive relationships. This reduces the number of direct relationships that must be managed but may require a client to traverse, or transit, intermediate realms in order to get to the realm of the end service. For example, given realms X, Y, and Z, where X–Y has a direct relationship, Y–Z has a direct relationship, but X-Z does not have a direct relationship. In this case, X–Z has a transitive relationship through Y. In order for a client in X to get to a service in Z, the client must transit Y, because X and Z do not have a direct relationship. The client first obtains a cross-realm TGT from realm X to realm Y. That cross-realm TGT is then used to obtain a cross-realm TGT from realm Y to realm Z. The cross-realm process may be extended to as many steps as are necessary for a client to reach the target realm of a service. Each step in that process is identical and results in a cross-realm TGT for a realm that is "closer" to the realm of the service.

Within a collective, realms are typically organized as a tree, or "realm hierarchy," where each realm has a direct relationship with one parent and potentially several children. To get from one realm to another, the client may have to climb up the tree toward the root, and then down the tree to get to the desired service's realm, collecting inter-realm TGTs along the way. The tradeoff between direct and transitive realm structures is the key management overhead required for direct relationships vs. the network overhead required to transit intermediate realms. Both direct and transitive relationships can be used in combination. For example, the majority of realms may be arranged using transitive cross-realm relationships, as in a realm hierarchy. Where performance or trust is an issue for specific realms, those realms can also have direct cross-realm relationships, allowing clients to go directly to the target realm, thereby "short circuiting" the need to transit intermediate realms in the realm hierarchy.

Tickets issued as a result of cross-realm authentication have within them the names of the realms transited by the client within them. The list of transited realms is referred to as the "transited realms list." This allows a service (or any intermediate realm) to ensure that all the realms in the path that participated in cross-realm authentication can be trusted not to lie about the client's identity. However, in general, a realm will either be trusted or not. A trusted realm will be part of a cross-realm collective. Untrusted realms will be excluded from that collective or will not be placed in the path between critical clients and services. If principals or services must avoid the use of a less trusted realm due to the sensitivity of their work, direct relationships can be established between those realms, bypassing those less trusted realms.

Ticket Restrictions

If the client sends a credential — that is, a ticket and the associated session key — to another principal, the recipient's use of the client's identity is limited solely by the ticket's implicit restrictions. The lifetime of a ticket is one obvious implicit restriction that defines the time during which a ticket may be used. Another implicit restriction is the service name in the ticket; that service name is an implicit restriction on the use of the ticket. If the service name in that ticket is the ticket-granting service (TGS), and hence the ticket is a TGT, then the holder may obtain any other tickets. Obviously, handing over your TGT (along with the TGT's session key) to another principal requires a very high level of trust in that principal.

In some cases, the implicit restrictions in a ticket may be sufficient. For example, consider a client that wishes to print a file on a file server using a print server. If the client sufficiently trusts the print server, the client can simply send a credential (ticket and session key) for the file server to the print server. The print server can then use that credential to access the file server in the client's name. The service ticket (for the file server) in that credential only allows the print server to access the file server using the client's identity; it does not allow the print server to access any other services using the client's identity. However, the client must trust the print server sufficiently to allow the print server unrestricted use of the client's identity when accessing the

file server. If that trust is not warranted, authorization data can be used to further restrict the print server's use of the client's identity.

In many cases we would like to restrict certain common uses of a credential by another principal without having to first agree on the syntax or semantics of authorization data. There are several common forms of restrictions provided by Kerberos to deal with these cases. (Most if not all of these cases could use authorization data to restrict the ticket's use.) The codification of these restrictions by Kerberos is in large part recognition of common use. These restrictions also allow common constraints on ticket usage that are based on site policies that are enforced by the KDC.

Address Restrictions

A ticket's use may be limited to specific network addresses, such as the originating client work station. Those address restrictions may be used to help restrict the use of credentials sent to another principal and can also help to foil the use of stolen credentials. Multihomed systems (systems with more than one network address or interface) require special care to ensure that address restrictions include the appropriate addresses for the system. In some cases it may be appropriate to restrict use to a subset of the addresses or interfaces on the system (e.g., inbound or outbound interfaces on a firewall). In other cases there may be no control over, or any desire to control, which addresses or interfaces are used, such as on a high-performance server with many network interfaces. Address restrictions placed on a TGT are propagated to service tickets obtained with that TGT unless otherwise specified. Address restrictions may also be empty, in which case there are no restrictions on where a ticket may be used from. There are obvious security concerns with empty address restrictions. However, outside of a few uses, the use of address restrictions has fallen out of favor. This is due to the difficulty for clients and intermediaries to determine the addresses that a recipient may need.

Address restrictions provide the ability to restrict the use of credentials to a specific machine when those credentials are sent to an intermediary. It may also be desirable to restrict the intermediary's ability to propagate those credentials to other systems and services. (The term "propagation" used here means propagating the use of a credential; there is nothing that can be done to prohibit physical propagation of the ticket.) Ticket attributes known as "forwardable" and "proxiable" allow restricting the subsequent propagation of credentials by a recipient. Those restrictions are binary; they restrict further propagation of the credential by the recipient, or they do *not* restrict further propagation of the credential by the recipient. Finer-grained control must use restrictions in the authorization data. Sites may choose to limit the KDC's willingness to forward or proxy tickets. Similar indicators known as "forwarded" and "proxy" allow a service to determine if a ticket has been obtained in this manner. Services may modify their behavior based on the setting of those indicators. For example, a file server might choose to allow only read-access to certain files when presented with a ticket that has the proxy indicator set.

Proxiable

The proxiable attribute allows the holder of the ticket to ask the ticket-granting service (TGS) to modify the address or lifetime restrictions in the ticket. That results in another ticket with different address or lifetime restrictions. That resulting ticket always has the proxy attribute set. That proxy attribute may be checked by services to determine whether the ticket is from the original client or an intermediary. Proxiable tickets are used to restrict the use of a client's identity to a specific service; a proxiable ticket allows no changes to the ticket other than to the address restrictions. Sending a proxiable ticket to an intermediary allows that intermediary to propagate the ticket to other intermediaries.

For example, a client may provide an intermediary a service ticket for a file server where that ticket has the proxiable attribute set. This allows the intermediary to obtain another proxy or proxiable tickets for the file server and send that ticket to another intermediary, thus allowing other intermediaries access to the file server using the client's identity. Alternatively, the client may obtain a proxy ticket without the proxiable attribute set in the ticket. Lacking the proxiable attribute, that ticket can be used only by intermediaries that satisfy the address restrictions in the ticket. If there are no address restrictions in that ticket, there are effectively no restrictions on which intermediaries may use the ticket. However, what the ticket may be used for is still restricted implicitly by the ticket itself (e.g., the service name in the ticket). Client-specified authorization restrictions may further restrict the use of a credential (see below).

Forwardable

The forwardable attribute is similar to the proxiable attribute. The most significant difference is that the TGS will not issue another TGT based on a TGT with only the proxiable attribute set. A forwardable TGT effectively

allows the holder (assuming they also have the TGT's session key) unrestricted use of the identity in the TGT: forwardable and forwarded tickets — including other TGTs — can be obtained by anyone holding such a TGT. A TGT that is only proxiable does not allow the holder to obtain another TGT.

A forwardable TGT is typically sent if unrestricted use of the client's identity is desirable. One of the few cases where this is desirable is when a user logs into another computer system using, e.g., telnet. In that case the use is effectively establishing the same identity on another remote system. Although we could require the user to go through an initial authentication process again on that remote system (to obtain a TGT), that would provide little additional security and simply irritate the user. The difference in application between forwardable and proxiable tickets can be subtle, but important. In essence, there are three attributes that determine what requests the TGS will honor based on the ticket presented to it: forwardable, proxiable, and whether or not the ticket is a TGT.

Lifetime

A ticket's lifetime is an implied restriction. A proxiable or forwardable ticket's lifetime may be decreased but never increased.

Proxy Services

A proxy service is a service that performs a function on behalf of the client and that uses another end service in order to perform that function on behalf of the client (for example, a client wishing to print files using a print server where the files reside on a file server). The print server acts as a proxy for the client in order to access the files on the file server. The basic form of a proxy provides only implicit restrictions on the use of the client's identity by the intermediate service. This may be sufficient for some clients and services. In the previous example, the client must first obtain a proxy ticket for the print server. That ticket will show the requesting client as the client principal name, and the file server as the service principal name. That proxy ticket may be based on an existing service ticket the client holds for the file service, or it may be obtained directly using a TGT.

1. The client obtains a proxy service ticket for the file server. If the client possesses a ticket for the file server with the proxiable attribute set, that ticket may be used to request a proxy ticket from the TGS. The client sends the file server service ticket in its possession to the TGS, requesting a proxy ticket along with new address restrictions, if any. The TGS returns a service ticket for the file server with new address restrictions. That service ticket will, by default, have the proxiable attribute cleared and will always have the proxy indicator set.

   If the client does not possess a proxiable ticket for the file server, the client must obtain a proxy ticket for the file server using a TGT. That TGT must have the proxiable attribute set. This process is similar to the one described above, only it follows more typical TGS semantics.
2. The client authenticates to the print server using a conventional client–service authentication exchange. The client then sends the proxy credential (ticket and session key) obtained in the previous step to the print server. A variety of means may be used to send those credentials; the Kerberos "credentials" (CREDS) message is intended specifically for this purpose and ensures that the session key associated with the ticket is protected during the transfer of those credentials.
3. The print server uses the file server credential obtained in the previous step to authenticate to the file server, and obtain access to the file server, using the client's identity.

Note that when presented with such a ticket, the file server has no way of knowing that it is not really the client, but the print server, that is requesting access — the client name shown in the ticket is the originating client, not the print server. The file server may infer some information from the fact that the proxy indicator is set in the credential, for example. While useful, this does not provide very granular control and requires that the client must have an fairly high level of trust in the print server. Unless the file server places additional restrictions on access to files based on the setting of the proxy indicator, the print server has full access to any of the client's files. More granular restrictions require the use of client-provided authorization restrictions.

Authorization

Kerberos defines the rules for packaging authorization data elements in tickets and the semantics for placing those elements into tickets. Kerberos does not define the interpretation of those authorization data elements.

There are several points in time where authorization information may be provided or embedded into a ticket, ranging from the initial authentication exchange, to the client–service authentication exchange, and several points in between. There are also several possible sources of authorization information, including the client, as well as authorization services that may be a part of, or accessible to, the KDC. Authorization data provided by clients is referred to as restrictions, because the data restricts the authorized use of a client's identity. (Client-provided authorization data obviously should not be used to amplify the client's authorization, or clients could grant themselves any authority.)

Each authorization data element has a type associated with it. Kerberos defines the syntax of the type information, but does not generally define the interpretation of those types. Authorization data element types are application- or service-specific. Kerberos does not otherwise define the contents of the underlying authorization data elements, and KDCs generally do not interpret those elements, but treat them as opaque objects. Interpretation of authorization data elements is generally a function of each service. By convention or agreement, some elements may have meaning to a large number of services, and thus have a common syntax and interpretation for those services. In other cases, authorization data elements will be meaningful only to a single service, and thus the interpretation of those elements can be performed only by that service. Thus, the use of authorization data requires that the client and the end service (i.e., the applications) agree on the syntax and semantics of the authorization data.

In essence, Kerberos simply provides the ability to securely pass authorization data through intermediate services: the data is sealed (encrypted) in the ticket for the end service by the KDC using the end service's key; the data is unsealed (decrypted), by the end service using its service key. Because authorization data is sealed in a ticket, an intermediate service cannot tamper with that information. However, an intermediate service may be able to modify certain implicit restrictions or may add authorization information to the ticket, depending on ticket attributes.

During the initial authentication process between the client and the authentication service (AS), both the KDC and another authorization source may provide authorization data that is to be placed into the TGT. That data is generally propagated to all other tickets obtained using that TGT. That is, when the TGT is used to subsequently obtain a service ticket from the TGS, the authorization data in the TGT is copied to the service ticket as part of the service ticket construction by the TGS. KDC-supplied authorization data typically bounds the client's authorization. The authorization data placed into the TGT typically represents information that is widely applicable, and that would be of interest to most or all services. For example, KDC-supplied authorization data may include all of a client's group memberships.

The ticket-granting service (TGS) provides the same facilities as the AS for placing authorization data into a ticket. The KDC, or another authorization source, may provide authorization data that is to be placed into the service ticket. In addition, the client may also provide additional authorization data (i.e., restrictions) to be placed into the resulting ticket. That authorization data is in addition to the authorization data that is copied from the TGT used to obtain the service ticket. The authorization data placed into a service ticket as part of the TGS exchange typically represents information that is specific to a service; it may also represent information that is specific to a client–service pair.

Finally, the client–service authentication process provides an additional point at which the client can provide authorization data to the service. The client places additional authorization data into the authenticator that is part of the application request (AP-REQ) message. That authorization data represents restrictions that the client wishes to communicate to the service and that is specific to the session. Thus, at the point when a client authenticates to a service, the service has the sum of the authorization data and that is provided as part of the authenticator in the AP-REQ, the service ticket, and the TGT. That authorization data includes all client-specified restrictions.

Note that the AS does not define the ability for clients to specify authorization data (i.e., restrictions) in the authentication service request (AS-REQ) message, and thus place restrictions into the TGT. (The syntax of the AS-REQ allows this, but the semantics of the protocol preclude it, although it could be provided as preauthentication data if needed.) However, there is nothing that prevents a client from subsequently requesting a TGT from the TGS and placing restrictions into the resulting TGT at that time — for example, in the case of obtaining a proxy or forwarded TGT using an existing proxiable or forwardable TGT. The TGT is simply a ticket for the TGS, and there is nothing that precludes the TGS — or any service for that matter — from issuing a ticket for itself.

Capabilities and Delegation

A capability refers to a credential that has certain rights associated with its possession. Those rights may be both implicit in the fields of the associated ticket and explicit, using authorization data encapsulated in the ticket. A capability that has no address restrictions is sometimes referred to as a "bearer proxy," because it may be used by anyone (client or service) who possesses the credential.[14]

Anyone who possesses a credential with a ticket that is forwardable or proxiable can change or remove address restrictions from the ticket. Anyone who possesses a credential with a ticket that is forwardable or proxiable can also add to the authorization data. That authorization information should never be additive and thus allow the holder to amplify his privileges, thus the use of the term "restrictions" to refer to client-provided authorization information in such tickets. That is, it is acceptable for any holder to further restrict authorization by adding to the authorization data to the ticket; it is not acceptable for any holder to further amplify authorization by adding authorization data to the ticket.

To illustrate the use of capabilities, we again use the example of the client, print server, and file server. The approach illustrated in this example must be used carefully to guard against unwarranted amplification of privileges by intermediate services. For this example, we define authorization data with semantics that are similar to what one might find in an ACL with the triplet:

<id=principal><object=name><permissions=list>

In this triplet, "user" specifies who (a principal identifier); "object" specifies the name of the object to be acted on; and "permissions" specifies the allowable actions by the user on the object. If "id" is empty, then the implied ID is the client name listed in the associated ticket. An authorization data element is thus a triplet as defined above.

Once again, the client wishes to print a file using a print server (the intermediate, or proxy, service), where the file is on a file server (the end service). However, the client does not place a tremendous amount of trust in this print server, and therefore wants to restrict the print server's access. Specifically, the client wants to restrict the print server to read-access for a single file that is to be printed, and wants to restrict that access to a relatively short period of time. We assume that the client already has a service ticket for the print server and a proxiable service ticket for the file server.

1. The client requests a proxy ticket from the ticket-granting service (TGS) for the file server. In the TGS request, the client provides the proxiable service ticket for the file server that is already in the client's possession; requests a lifetime of 30 minutes; specifies the proxy attribute; and has cleared the proxiable and forwardable attributes. If the client wishes to restrict the ticket to the use of a specific print server with a known network address, then the address restrictions in the TGS request specify only the print server's network address. The client could leave the address restrictions empty if the network address of the print server was unknown, or enumerate a list of addresses if the print server is multihomed, or if any one of a pool of networked printers might be used to satisfy the request.

   The following element is specified in the authorization data field of the TGS request (or more accurately, the authorization data field of the AP-REQ that is part of the TGS request):

   <id=><object=/home/Hanley/thesis.ps><permissions=read>

   The interpretation of that triple is: id is null, and therefore interpreted as the client name in the ticket; object specifies the file "/home/Hanley/thesis.doc"; permissions specify read-access. The interpretation of that authorization is: "The client principal name specified in the ticket cannot perform any operation except to read the file '/home/Hanley/thesis.doc'."
2. The TGS constructs a new ticket and sends the new ticket back to the client. That new ticket is identical to the original proxiable service ticket for the file server (provided in the TGS request), except that the new ticket has the client-specified authorization data sealed within it; the proxy indicator set; the proxiable and forwardable attributes clear; and a lifetime of 30 minutes (the new ticket may also have different address restrictions). The new ticket also has a new session key.
3. The client authenticates to the print server using a client–service authentication exchange.
4. The client sends the proxy credential (ticket and session key) obtained in step 2 to the print server using a credentials (CREDS) message.

5. The print server authenticates to the file server using the proxy credential, obtained from the client in the previous step, using a conventional client–service authentication exchange. The print server is now communicating with the file server under the client's identity.
6. When the file server unseals the ticket received in the previous step, the authorization data in the ticket, placed there by the TGS in step 2, is exposed to the file server.

At this point, the print server and file server have authenticated, with the print server using the identity of the client. The file server has no knowledge of the fact that it is the print server actually acting on the client's behalf. However, the print server — through the authorization data in the ticket — knows that restrictions have been placed on the client's access and, we must assume, will enforce those restrictions. (If we cannot trust the file server to properly enforce access controls on its own files, then it is of questionable use for storing controlled information. We cannot solve that problem with Kerberos.) Also, because the ticket expires after 30 minutes, the print server will no longer be able to access the client's file on the print server after that time.

The conventions that control how authorization data is interpreted, the potential sources of that authorization data, and the ticket attributes used, are extremely important to ensure the integrity of this example. By convention, we have agreed that the presence of any authorization elements (i.e., authorization triples) in the authorization data implicitly restricts actions to those that are explicitly enumerated. While those enumerated elements are necessary, they are not sufficient for a complete and secure solution. If the ticket given to the print service had the proxiable or forwardable attribute set, the print service could go back to the TGS and obtain a new service ticket with different authorization. That would allow the print service to obtain access to any of the client's files. Note that this also implies that care should be exercised to ensure that no unwarranted authorization data is in the proxy ticket, as might be the case if the original (proxiable) ticket from which the proxy ticket was obtained had unwanted authorization information in it. Moreover, we cannot allow those tickets to be proxiable or forwardable, to eliminate the possibility of the print server amplifying its privileges by adding authorization data to a ticket.

Because the authorization data is created by the client, that authorization, while sufficient for the needs of the client, is not sufficient for the needs of the file server. The file server did not participate in the creation of the authorization data, and therefore should treat it as suspect. If the file server based all access control decisions only on the authorization data in the ticket, any client could grant itself any rights to any file. For example, there is nothing to stop the client from requesting a proxy with authorization data that specifies access to another user's files and using the resulting proxy ticket itself. This is one reason why proxiable and forwardable tickets should never be given out freely to untrusted intermediaries if authorization data could be used to amplify privileges.

If the file server blindly believed and obeyed the authorization data in the ticket, a client could use a proxy to gain access to any files. That would obviously not be very secure. Thus, this example is secure only if the file server has additional rules it applies to make authorization decisions, such as ACLs, to limit the authorization of the client. In other words, the file server must first check the authorization specified by its ACLs against the client's identity; with that as the authorized limits for the client, the file server can then determine if the authorization specified in the ticket is within those limits.

Note the temporal difference between capabilities and ACLs. To provide temporary, delegated access to a print server in an ACL-based system, the ACL on the file server would have to be modified temporarily to allow access by the file server. Constantly modifying ACLs could seriously degrade performance. However, there are practical limits to how much authorization data can be placed into a capability. This points to a need for both mechanisms: ACLs for long-lived and relatively static authorization information, and capabilities for more dynamic and context-specific information, as is found in delegation.

In the example above, the capability constructed by the client may be used by anyone who possesses the capability (subject to, for example, address restrictions). The client could also restrict the use of the capability to a specific principal using the "id" field in the authorization triplet. For example, by placing the print server's principal identifier into the ID field. This would require that the print server use two credentials to access the file server: the proxy credential provided by the client (showing the client identity in the ticket, and showing the print server's identity in the authorization data); and a credential for the print server itself (showing the print server's identity), to prove to the file server that the print server is the principal listed in the "id" field of the authorization triplet of the client proxy credential.

Identity-based restrictions, in conjunction with the other usage guidelines discussed above, would eliminate the possibility of the print server giving the client's proxy credential to another service, and of the other service subsequently using the credential to obtain unauthorized access to the client's files. This type of restriction

would be preferable to address restrictions and also provides the ability for the file server to audit and control access based on the identity of both the client and the intermediate service. This would allow the file server to, for example, enforce additional restrictions based on the identity of the intermediate server. For example, the file server may choose to prohibit write-access to files by print servers, regardless of what permissions are specified in the authorization data. Another example is to restrict access to certain files by "public" printers, regardless of the file specified in the authorization data.

Management

Management, performance, and operation are all reflections of one another. A system that makes many demands on the environment will require more resources to meet and maintain those demands, whether those demands be disk storage, CPU, network bandwidth, users, or support personnel. A system that makes many assumptions about the environment will require more resources to meet and maintain those assumptions. Those assumptions are simply implied demands the system places on its environment. Those demands have a direct influence on the cost of achieving an acceptable level of performance and the ability of the implementation to perform its intended function. The greater the demands, the higher the cost of operating and managing the system, or the supporting elements that the system depends on. If those demands are not satisfied, a system's performance and usability will suffer. In the extreme case, performance becomes so poor that the system cannot carry out its intended function.

The cost of satisfying demands and assumptions can rise very rapidly in a distributed environment. The more distributed an environment, the less likely that demands will be satisfied over a given number of systems, and the higher the cost of satisfying those demands. Of special concern is the ability of a system to function effectively in the face of changes in the environment. The more distributed an environment, the higher the probability that changes to the environment will occur over a given unit of time and that intervention will be required to compensate for those changes. Thus, the cost of maintaining assumptions increases.

Those problems are magnified in distributed security. The greater the demands placed on the environment by the security system, the more likely it is that performance problems will result and that the security system will fail to carry out its assigned function. The more assumptions that are made about the environment, the more likely it is that intervention will be required to compensate for those changes. Intervention increases the probability of errors, which can lead to security problems.

It is important to distinguish the demands made by Kerberos as a technology and the demands made by Kerberos as a security system. Kerberos technology makes modest demands on the environment, and satisfying those demands should be well within the means of most organizations. Kerberos as a security system can make very insignificant or very oppressive demands on the environment, depending on the level of security an organization needs or chooses to enforce. We use the term "appropriate" to describe that level of security and to qualify those elements that are outside the scope of Kerberos — or any security technology. If an organization decides that "appropriate security" means "very high security," then demands, assumptions, cost, and effort will all increase.

Users

One of the first concerns usually raised by network and system administrators is "What is this going to do to my users?" That is a justifiable concern, because any change that is visible to users will tend to produce a heavy influx of support calls. Kerberos can be virtually invisible and undemanding of users, or extremely visible and oppressive in its demands. That choice is a function of the level of security the site chooses to enforce using Kerberos. For the security needs of the vast majority of sites, Kerberos need not be visible to the user community.

Users are generally unaware of Kerberos, except during the initial authentication process (i.e., sign-on), when they must provide their Kerberos principal identifier and a password, or some other proof of identity. If the Kerberos sign-on is integrated into the host sign-on, Kerberos can be made invisible to the user. If the Kerberos sign-on is not integrated into the host sign-on, or the host has no concept of a sign-on, a separate Kerberos utility to allow the user to sign on and complete the initial authentication process is required.

The result of the Kerberos initial authentication is a ticket-granting ticket (TGT), which is placed into a credentials cache, and which applications may subsequently use for obtaining service tickets in order to authenticate to services. The process of obtaining service tickets using the TGT, and the subsequent authentication exchange between the client and the service, is invisible to the user. Kerberos utilities are typically

provided to view the tickets contained in the credentials cache. However, with the exception of diagnostics and troubleshooting, those utilities are typically not used and are unnecessary.

One of the few times a user might encounter different behavior due to Kerberos is if their TGT expires. All tickets, including the TGT, have a lifetime. Applications will automatically request a new ticket if the old one has expired. However, an application cannot request a new TGT without user involvement. That is, the user must go through the initial authentication process to obtain a TGT. Whether the user community ever encounters that behavior will depend on the lifetime chosen for TGTs. If that lifetime is longer than the average workday, most users will never see this behavior.

Assumptions

Kerberos makes certain assumptions about the environment and the security of the various systems and individuals that make up the Kerberos environment. When discussing these assumptions it is important to distinguish what is required for any distributed or network environment, what is required for any distributed security system, what requirements are specific to Kerberos, and what requirements are specific to a Kerberos implementation.

Minimal assumptions and requirements necessary for any distributed environment include:

- A functional network for clients and services to interact.
- A functional network directory service for clients and services to locate each other.
- A functional software distribution system to distribute software to computer systems that host clients and services.

Assumptions and requirements that are common to virtually all distributed security systems are negotiable and depend on acceptable cost and risk. These include:

- Appropriately secure systems for hosting clients and services
- Appropriately secure software distribution service
- Appropriate protection of identity information by individuals (passwords, smart cards, tokens, etc.)

Assumptions and requirements that are Kerberos-specific are negotiable and depend on acceptable cost and risk. These include:

- Appropriately secure systems for hosting KDCs
- Appropriately secure time service, with loosely synchronized clocks on all systems on which Kerberos operates

The following discussion provides security recommendations for the assumptions and requirements enumerated above. These recommendations are common to virtually all implementations. However, they do not account for budget or other organizational constraints, and actual requirements will depend on cost–risk tradeoffs, which will be different for each deployment.

Directory Service

Kerberos typically requires the Internet domain name service (DNS) to construct the names of service-based principals and locate those principals on the network. An ineffective DNS or an inconsistent naming structure can make this job more cumbersome. Although many network services depend on a network naming system to function, a compromised name service does not present a security threat to Kerberos, other than possibly a denial-of-service attack. Note that such a denial-of-service attack would likely affect many network services, and not just Kerberos.

Software Distribution Service

Any large distributed environment requires a software distribution service for cost-effectively distributing and installing software on physically remote systems. That distribution system should be secure to ensure that the integrity of the security software itself is not compromised.

Secure Time Service

Loosely synchronized clocks are typically required between the KDCs, and between KDCs and application servers (e.g., within five minutes). Implementations vary in their requirements for clock synchronization. Unsynchronized clocks primarily represent a security threat due to replay attacks. Depending on the Kerberos

implementation and the protocols used, clock synchronization may or may not be required. However, synchronized clocks are generally desirable in any large network, especially for auditing and network and system management to correlate activities and events across the network. If timestamps are used as the basis for replay protection, the time service used to synchronize clocks should be secure.

KDCs

Because the KDC is the trusted third party for all principals in the realms it serves, the KDC should be both logically and physically secure. Failure to secure the KDC can result in the compromise of an entire realm. The KDC should support no applications, users, or protocols other than Kerberos. (That is, everything except Kerberos has been removed from the machine.) Ideally, the system will not support remote network access except by means of the Kerberos services it offers. Remote administration of KDCs and principals is a fact of life in today's environment. Most modern Kerberos implementations provide a secure remote administration facility.

Services

Systems that host services, or "application servers," should be secured to the level required by the most sensitive application or data on that server. Failure to adequately secure the application servers may result in the compromise of services that operate on that application server, and their data. Note that a compromise of an application server compromises only those applications on the server and does not compromise any other principals.

Clients

Client systems should be secured to the level required by the most sensitive user of the client or the most sensitive application that is accessed from that client. Failure to adequately secure client systems may result in the compromise of any users of the client system or compromise of data accessed from the system. A compromised client puts all users of the client at risk. For example, a password grabber on a client compromises anyone who uses the client; a virus potentially compromises the data of any application accessed from that client. A compromised client does not compromise principals that do not use that client. However a client compromise could spread if one of the users of that client has elevated privileges, e.g., a Kerberos administrator. Kerberos administrators (or anyone with elevated privileges) should not use a client system unless they have an appropriate level of trust in that system.

Identity Information

Identity information, no matter what the form, requires appropriate protection of that information by individuals. If passwords are used, those passwords should be sufficiently strong. Most modern Kerberos implementations provide password policy enforcement to minimize the use of weak passwords. If public key credentials are used, protection of those credentials is as important as password protection. If additional security is required, technologies that provide two-factor authentication, such as token cards or smart cards, may be used; appropriate care in protecting those devices must still be exercised by the individual. Note that a compromise of an individual does not implicitly compromise any other Kerberos component or principal. However, as with any system, administrative personnel who have elevated privileges should be of special concern. For those individuals, two-factor authentication may be appropriate.

Operation

In terms of operational management, clients are by far the most important, with services a distant second, followed by KDCs. Implicit in that ranking are the associated infrastructure elements that are required for each Kerberos component to perform its function. That ranking obtains from the relative numbers of the components. Clients are typically the most numerous by orders of magnitude, and their sheer numbers magnify even the smallest manageability problem. That is not to say that management of KDCs is unimportant, but if given the choice between a few skilled people trained and dedicated to managing a few KDCs vs. 100,000 users and clients, the choice should be obvious.

Clients

Other than installation, the primary manageability concern with clients is locating KDCs and services (discussed later in this chapter).

Servers

The primary management overhead associated with service principals is the maintenance of the key table. As previously discussed, the key table holds a service principal's key. Communication of the key should be done securely, which means either manually communicating the key out-of-band or pulling the key from the KDC using a key management utility on the system on which the service operates. The latter method of pulling the key from the KDC is preferable.

For example, once Kerberos client software is installed on the application server, a key management utility can be used by an administrator to access the KDC, establish a secure session, generate the service key, and place the service key into the service's key table. The administrator effectively provides the secure channel for securely communicating the initial service key. Once the initial keys are established, secure key update, or "key rollover," can be automated. That key rollover can be initiated on the server to pull a new key from the KDC to the server, or a KDC can push a new key to the server. Implementations vary in the sophistication of the key management utilities available and the facilities for automating the key rollover process.

KDCs

A fully equipped KDC generally includes a variety of services for administration and management, database propagation, password change, etc. Some of those services can be quite complex. However, the main services provided by a KDC are for authentication and are quite simple. Those services do not, as a rule, maintain state or require write-access to the principal database.

Most implementations differentiate between "primary" and "secondary" (or "master" and "slave") KDCs depending on the services they provide. A primary KDC typically provides a reference copy of the principal database, as well as hosting services that require write-access to the database. Secondary KDCs typically maintain read-only copies of the database. Implementations vary tremendously in the mechanisms used to propagate information from primary to secondary KDCs. In the most primitive mechanisms, a bulk propagation of the entire database is performed at fixed intervals. More sophisticated mechanisms incrementally propagate only those database records that change in real time. The issues associated with periodic bulk propagation are numerous and significant. Incremental propagation is a prerequisite for any large-scale production implementation.

Services that require write access to the principal database include those required for day-to-day administration of the principal database, such as adding, deleting, and changing principals. Administrative functions are generally performed using a special administrative tool, either locally on the KDC, or remotely. Password-change operations also require write access to the principal database. Password-change is typically the only operation in which the general client population requires access to a service on the primary KDC — that is, a service that has write-access to the principal database. Although implementations vary, the inability of clients to access the primary KDC will typically preclude password-change operations. That argues for a primary KDC configuration that provides system and network redundancy and automatic failover. Beyond the administrative functions associated with principals, there is little additional work involved in managing a KDC.

The primary services used by clients — the authentication service (AS) and ticket-granting service (TGS) — do not generally require write-access to the database. Thus, secondary KDCs should, as a rule, be the client's first selection when locating a KDC to provide those services. It is not unusual for all AS and TGS requests to be serviced by secondary KDCs, and to dedicate the primary KDC to administrative services. This allows the resources of the primary KDC to be dedicated to services that only the primary KDC can provide, which allows it to serve a much larger client community.

Each entry in the principal database is typically encrypted in a "master key" that is defined when the database is created. That master key prevents compromise of the realm should a backup of the principal database be inadvertently released, for example. However, for unattended restart of the KDC and unattended operation of services that must manipulate the database, the master key must be kept in persistent storage. If unattended KDC restart is not required, the master key can be typed in on the console when the KDC starts. However, that typically does not make the master key available to other services that may require access to the database, such as administrative services. Because of those issues, virtually all implementations use a master key that is kept in persistent storage, such as a disk file. Obviously, keeping the master key secure is of paramount importance, and any backups should exclude storage containing a copy of the master key.

Realms

Most of the issues involved in the use of multiple realms revolve around the client's ability to locate KDCs and services in a realm. The ease or difficulty with which clients can perform those functions, and the associated management overhead, are usually the determining factors in whether or not an organization uses multiple realms.

If multiple realms are used, cross-realm keys must be established between realms, and appropriate entries placed into the principal database. Key generation and creation of the principal database entries require very little effort. However, those cross-realm keys must be communicated between realms in a secure fashion. Unless a secure channel already exists between realms, those keys should be communicated using a secure, out-of-band mechanism, such as physical mail. Once those initial keys are established, a secure channel can be formed to change the keys periodically.

Note that a user can have identities in multiple realms. For example, the same physical individual may have a principal identity in multiple realms. Although those two identities may represent the same individual, Kerberos does not make that association. By the same token, there is nothing that prevents a client computer system from being used for authenticating an individual to any realm or accessing a service in any realm. That situation would not be unusual in an environment with multiple realms and a roving user community. Although it is typical for client systems to define a default realm as a convenience for users, that default realm is only a convenience and, unless otherwise constrained, does not limit the use of the client by individuals in a single realm.

A service, or more precisely, the instantiation of an application on a host computer system, may also operate in multiple realms. While it is unusual, and there are security implications that must be considered, there is nothing that prevents one system from hosting applications that have identities in multiple realms. Nor is there anything that prevents the same application on the same system from having an identity in multiple realms. Having a common system or application that has an identity in multiple realms may be an alternative to cross-realm authentication. For example, consider a database that is shared between two groups in different realms. The database service can be placed into one realm, with the other group using cross-realm authentication to access it. Alternatively, the database can have an identity in both realms, with each group accessing the database as a service in their own realm, thus eliminating the need for cross-realm authentication. Again, there are security implications in such an approach that must be taken into account. Specifically, management of the service keys must be carefully considered.

Principals

Management of principals is similar to that of any system that maintains identity information. Principals must be added, removed, and modified. A principal identifier should not be reused until all services that may have local copies of the principal identifier have been notified. For example, if a service uses a principal identifier in a local access control list (ACL), the ACL must be updated before the principal identifier is reused to ensure that the new entity does not have unwarranted access to that service.

All implementations provide tools to perform administrative functions. For large-scale deployments, it may also be desirable to couple Kerberos administration to an enterprise administrative system. As with any system that uses passwords, resetting passwords is probably the most common administrative function performed in Kerberos. Some implementations allow administrative functions to be tightly constrained (for example, limiting help desk personnel to performing password resets and not allowing them to perform other administrative functions, such as adding, removing, or otherwise examining or modifying principal entries).

Key Strength and Rollover

As mentioned above, there are a number of keys that should be rolled over periodically. Those keys are generally randomly generated bit strings and are very resistant to any attack short of an exhaustive key search. Thus, the strength of the keys and the required rollover frequency depend almost entirely on the key length used. This suggests that the strongest possible key strength, such as triple-DES, should be used for critical keys. An exhaustive search of the triple-DES key space is well beyond the means of any organization today or for the foreseeable future, with the possible exception of a few government intelligence agencies.

As for all services, the key strength and rollover frequency for a service should be appropriate for the sensitivity of the service. One service stands out as demanding the highest possible level of protection: the

ticket-granting service (TGS). All ticket-granting tickets (TGTs) received by clients are sealed in the key of the TGS, and all authentication with services is ultimately rooted in that TGT. If the TGS' key is compromised, the TGS can be impersonated, and with it the entire realm. Obviously, protecting the TGS's key is of paramount importance. Close behind the TGS in importance are the keys used for administrative services and cross-realm authentication.

Automation of the key-rollover process should eliminate virtually all management overhead associated with key rollover. For remote systems, rollover can be initiated from the KDC and pushed to the service, or it may be initiated by the service and pulled from the KDC. However it is done, automation of the rollover process for services on remote systems implies that an existing key is used to establish the secure channel for key rollover. If shared secrets and symmetric key cryptography are used as the basis for establishing that secure channel, the rollover process should strive to camouflage the key rollover sequence. That minimizes the probability of an attacker recording the sequence containing the new key and the subsequent compromise of the new key based on an old key.

Names and Locations

The majority of the management and operational issues with Kerberos revolve around names, the association of those names with physical or logical entities, and the location of those entities in the network. The naming and location issues faced by Kerberos are not unique to Kerberos and are faced by virtually all distributed environments.

Historically, services have been tied to machines, and those machines have a name that people know and understand, and the network software can be used to connect a client to that machine and implicitly to a service. In many environments, a single system or service might be known by many names, and as long as the client is able to connect to the service, no one much cares. When a system such as Kerberos is introduced that relies on names to identify and authenticate unique entities, names start to matter much more. All of a sudden, the name may be used not only for location, but authentication, and the client, the service, and Kerberos must all agree on what those names are attached to, and the network naming or directory service must also agree with where they are located.

Name services such as DNS provide solutions to the simple client–server connection problem. However, as the coupling between physical systems and services becomes more tenuous, we are left with the problem of finding an instance of the service (i.e., a system on which the service is operating) somewhere in the network. That service name may or may not have any relationship to a computer system's network name. Although there are many solutions to this problem, as of this writing there are no solutions that an implementation can rely on in most environments.

Name Spaces

Kerberos defines a name space consisting of realms and principals. Other than their own principal name, most users will have little or no knowledge of other Kerberos principal names, especially those associated with services. Thus it is left up to the Kerberos software and the environment to somehow map the names that people are familiar with to the corresponding Kerberos principal identities and locate those entities in the network. If Kerberos names are associated with an existing name space, such as DNS, and a name in one name space can be mapped trivially to another, most of the issues become relatively innocuous. If the names in the Kerberos name space are not associated with an existing name space, management effort and the probability of errors goes up significantly, as should be obvious from the discussion below.

Services

Services typically use an "instance" in the principal name to help distinguish different instances of the same service, e.g., name/instance@REALM. For example, the instance may distinguish the same service operating on different computer systems. Although it is generally the case that the same principal name would imply similar functions across different instances, that is by convention only. Different principal identifiers — the concatenation of the name, instance, and realm — are treated as completely different entities by Kerberos.

The instance is used by virtually all Kerberos implementations to locate the service on the network. For service principals, Kerberos clients by convention use the fully qualified DNS domain name of the host computer system on which a service operates as the instance. For example, wadmin/www.z.com@Z.COM might be a Web administrative service application on the system www.z.com. Other services may also be present

on the same system, and each of those services could have its own name with the same instance. For example, ccare/www.z.com@Z.COM might be a customer care service application running on the same system.

By convention, there is a generic host principal used for authentication to generic host services, such as telnet. By convention, those generic services share the principal name "host." For example, telnet clients would use the service principal name host/y.z.com@Z.COM to access to a telnet server running on system y.z.com. The principal identifier host/x.z.com@Z.COM represents the same principal name (host) with a different instance (x.z.com). Although host/y.z.com@Z.COM and host/x.z.com@Z.COM may imply a common service (i.e., a common function) on different systems, Kerberos makes no such implication. From the perspective of Kerberos, those principal identifiers are different, and therefore represent different entities; any implied similarity is by convention only.

Note that there is an implied relationship between the instance and the location of the service, and a client must know both in order to use a service. To establish a connection with the service (regardless of whether Kerberos is used), the location must be known, and the principal name must be known for the client to form the correct service name for that service and obtain the correct service ticket. This implied relationship can be either a great convenience or a great pain, depending on whether the relationship holds true.

Within a single realm, the principal names used for services and the manner in which a client forms the identifier of a service principal have a significant effect on the usability of the implementation. Services that use the common and generic "host" principal name are well defined and not a problem. For other services, those services' principal identifiers must be defined and known to the client. The instance name used for service principals can also present a problem for the client. Although the Kerberos convention is to use the fully qualified DNS domain name, or "long form," for the instance in the principal identifier, some DNS implementations return the "short form." This can present problems if one system uses the short form and another system uses the long form. From the perspective of Kerberos, those two identifiers are different, and hence different principals. Both of those identifiers must have a principal entry and an entry in the key table for the service — which increases management overhead — or an error will result when a client uses the wrong principal identifier to attempt to access the service.

KDCs

Before a client can do anything with Kerberos, it must locate a KDC in order to authenticate and obtain tickets for the individual using the client. Note that unlike service principals, which generally use the instance portion of the principal name to also locate the machine on which the service is operating, there is no implied KDC location based in the realm name. The only inference one can make from a realm name is that a KDC is operating on a system somewhere in the corresponding domain. For example, we can infer that a KDC for the realm Z.COM is probably located on a system somewhere in domain z.com.

If multiple KDCs are used for availability or performance, there must also be some means of directing the client to the appropriate KDC, or for the client to automatically locate a KDC should the first choices be unavailable. For systems that use primary and secondary KDCs, the client will also need to know how to locate the primary KDC for a realm for password-change operations.

Different individuals in different realms may use the same client. It is unrealistic to expect those individuals to know the names or addresses of KDCs in their realm, and therefore the job of locating a KDC falls to the Kerberos client software. Applications on the client may also access different services in different realms. As with individual principals, it is unrealistic for those applications to have embedded within them knowledge as to the location of KDCs in different realms, and again that job falls to the Kerberos client software.

Traversing multiple realms can also present problems for the client. Kerberos defines a standard mechanism for traversing realms that are arranged in a hierarchy. For other realm structures, there is no defined mechanism. Moreover, the client must know the realm in which a service resides. If a service is in a different realm, the client must perform cross-realm authentication to get to that service. In order to perform that cross-realm authentication, the client again must locate a KDC in each of the realms it must traverse.

The basic KDC-realm location problem has a variety of solutions, and implementations vary in how they solve the problem. The simplest and most primitive solution is to use a configuration file on the client. Typically, that configuration file defines a default realm and KDC, which the client uses unless told otherwise. That solution is sufficient for basic implementations. That configuration file may also enumerate a list of alternate KDCs and realms, and the primary KDC for each realm. Thus, changes to the environment may require that configuration file to be updated on many clients. For a relatively static environment, that may be acceptable. For even a moderately dynamic environment, that is unacceptable.

To solve the KDC realm location problem in an effective manner, as much static configuration information as possible must be removed from the client. Solutions that address the problem may make use of naming conventions for KDCs and may include the use of DNS aliases, rotaries, and informational records. Other solutions may use "referrals" or "redirection" to direct the client to the appropriate source. This solution requires only that the client be able to contact at least one KDC; that KDC is assumed to have the knowledge of how to get to other KDCs and realms, and can refer or redirect the client as needed.

Interoperability

The Kerberos 5 protocol defines what is necessary for implementations to be "wire-level" interoperable, and different implementations tend to be quite good about wire-level interoperability. However, the Kerberos standard does not address many of the host-specific or environmental issues that every functional Kerberos implementation must deal with, and there is no guarantee that two implementations will deal with the same issue the same way. *De facto* standards have typically developed on different platforms to address these issues. If a platform vendor provides a Kerberos implementation, that vendor will generally set the standard on their platform. Thus, while these issues are generally not significant, they are worth noting.

- Locating a KDC within a realm may be done in different ways. This can result in duplicate management effort in order to maintain consistency between two different representations of that information.
- Credentials cache locations and formats may vary. The primary concern is the ability for applications to access the TGT for obtaining service tickets. Unless applications use a common credentials cache to hold the TGT, the user may be forced to go through an additional sign-on.

The most significant interoperability issues between KDCs and clients are not a function of the Kerberos protocol, but specific features that KDCs or clients may require or support. This usually manifests itself in the types of preauthentication mechanisms supported, such as token cards, public key X.509 certificates, etc.

Although the standard defines client–KDC interactions, no standards, neither formal nor *de facto*, define KDC propagation mechanisms and administrative interfaces. Thus, those propagation mechanisms and administrative interfaces tend to be vendor-specific. The result is that, although it is quite feasible to use a mixture of clients and KDCs from different vendors, all KDCs within a realm must typically come from the same vendor. Between realms, cross-realm authentication couples the KDCs in those realms (not database propagation). Because cross-realm authentication is defined by the Kerberos standard, KDCs from different vendors in different realms should have no trouble interoperating.

Performance

Performance is the degree to which Kerberos can perform its intended function with a given level of resources. Kerberos will consume some resources, and the efficiency of Kerberos can be gauged by how effectively it uses those resources. Resources take the form of network bandwidth, and disk and CPU on clients, servers, KDCs, and personnel.

For performance, the KDC is typically the most important component, with services a distant second and clients third. That order obtains from the relative concentration of work performed by each of those components and the effects of inefficiencies or failure on other components. An inefficient KDC can affect a large number of clients and services, whereas an inefficient client generally affects only that client. Implicit in that ranking are the infrastructure elements needed to support each component. The efficiency of a KDC, by any measure, makes little difference if the network or directory service needed for clients to communicate with the KDC is inefficient or inoperable.

Encryption

One of the first concerns that usually comes to mind with any security system that uses encryption is the additional CPU and network overhead. In Kerberos, the use of encryption for authentication in the authentication service (AS), ticket-granting service (TGS), and application (AP) messages is intentionally limited, and the resulting cryptographic overhead is minor.

For applications that encrypt and decrypt data, the overhead may be very noticeable (whether or not those applications use Kerberos). That overhead depends on the amount of data that is encrypted, the encryption algorithms used, the efficiency of the implementation's algorithms, and the availability and use of hardware

cryptographic acceleration by the implementation. Data encryption and decryption overhead is generally not an issue on clients, as even moderately efficient software cryptographic implementations on today's client platforms are normally faster than the network. However, for servers the situation may be reversed, as those servers are typically the focal points for many clients. That is, the cost of encryption and decryption is spread over many clients, and a much smaller number of servers. Those servers may justify the investment in hardware cryptographic accelerators if performance is an issue.

Encryption of application data adds no measurable overhead to the network. The sole exception to this are protocols that exchange a very small amount of information in each message and that use a block cipher such as DES. This causes messages that are shorter than the block size of the cipher to be padded out to the block size of the cipher. For example, DES is a block cipher with a block size of eight bytes; encrypting a single byte results in an output that is eight bytes. However, the additional overhead added by Kerberos in this case will likely be unnoticeable, as it will be dwarfed by the overhead of the message envelope. Simply put, any protocol that transmits a few bytes of data in each message is, by definition, horribly inefficient at moving data — encrypted or not — and encryption will cause a very minor increase in that inefficiency.

Network

The demands Kerberos places on a network are modest and rarely an issue. Network demands will depend on several factors, including the behavioral pattern of clients, network topology, and the location of KDCs within the network. The KDC can communicate with clients using either UDP or TCP. Because of its greater efficiency, UDP is the preferred method. However, if firewalls are placed between clients and KDCs, UDP may not be feasible; for those clients, TCP may be used.

The additional network traffic produced by the Kerberos authentication process is simple to determine:

- *Initial authentication*. A single exchange between the client and a KDC at the beginning of the workday (AS-REQ and AS-REP). This exchange may involve more than one message in each direction, depending on the technology used for initial authentication. For example, a challenge–response token card typically requires an additional exchange between the client and a KDC.
- *Obtaining a service ticket*. A single exchange between the client and a KDC the first time an application service is accessed during the workday (TGS-REQ and TGS-REP). Different services require different service tickets, and thus each time a service is accessed the first time during the workday, this exchange will occur.
- *Client-to-service authentication*. A single message from the client to the service (AP-REQ). If the client requests mutual authentication, there is one additional message from the service to the client (AP-REP). The Kerberos authentication exchange between the client and service may be embedded in the application's session establishment messages and will not show up as an additional message, but rather as a nominal increase in size of the standard session establishment messages.

The size of the messages varies depending on various options and the amount of authorization information embedded in tickets. Assuming no authorization information, message sizes range from approximately 100 to 500 bytes.

KDCs

KDC performance is rarely an issue. The primary services provided by a KDC — those that are most used and have the greatest effect on performance — are the authentication service (AS) and ticket-granting service (TGS). The AS and TGS typically do not require local state, and typically require only read-access to the principal database. This allows liberal placement of KDCs within the network and eliminates the need to bind clients to specific KDCs. Moreover, because of the very simple and symmetric message exchanges and the reuse of common syntax and semantics in the protocol, KDC implementations tend to be quite compact and very efficient in their use of memory and CPU. Rates in excess of 20 AS and TGS exchanges per second for a KDC on a small system are not unusual.

The limiting factor on KDC performance is usually the I/O associated with the principal database. CPU overhead for encryption and decryption is usually a distant second (assuming that symmetric-key cryptography is being used), owing to the relatively small size of the messages processed by the KDC and the limited use of encryption for those messages. Disk resource requirements depend on the database used and the number of principals in the database; although requirements vary, a rule of thumb is 1 Kb of disk for each principal in the database.

Clients and Services

Implementations vary in what they require of systems that host clients and services. Generally, the additional overhead imposed on clients, services, and the additional network overhead for an application is unobtrusive. Disk and memory usage on those systems is typically quite small; the primary variation and resource consumption is typically not in the implementation of the Kerberos protocol, but in ancillary facilities such as graphical user interfaces. Again, although the basic Kerberos authentication process is typically unobtrusive, applications that encrypt large amounts of data may see very visible effects on performance.

Provisioning

As discussed previously, the inherent demands Kerberos places on the network are quite modest. Most modern networks should have little or no trouble with the additional network traffic. However, the network topology, KDC placement, and the location of clients and servers relative to each other and KDCs can have either an insignificant or a very significant effect on the network. Most network operations groups have the knowledge and experience to properly provision and locate KDCs in the network, and those groups should be consulted when determining provisioning requirements.

Key Services

Many modern networks have the concept of "key services," which are required for the proper functioning of a modern enterprise network. Key services typically include naming services, such as DNS, and may include time services, such as NTP. The systems that host those services are typically located in facilities at key points in the network, and those facilities are intended to ensure the availability of key services to all users in the face of network outages and other failures.

Those key service facilities will typically have a higher level of physical security than many other facilities. Key services facilities will usually define the location of KDCs in the network, as well as secure time services, if used. Those key service facilities also provide a baseline for the physical security of the KDCs. That security may or may not be sufficient.

Primary KDC

The primary KDC should be dedicated to administrative functions and data distribution. The primary KDC should use a high-availability platform with no single point of failure. The number of secondary KDCs and their propagation requirements obviously contributes to sizing of the primary KDC. The most significant effect on sizing the primary KDC is client password-change frequency. For example, for a user population of 100,000, with a password expiration of three months (approximately 60 working days), the system will be required to handle an average of approximately 1700 password-change operations per day. Virtually all of those password changes will occur at sign-on (when the expiration is detected and the user is forced to change his password), and most will center on a narrow band at 8 AM in any time zone. That can present a potentially significant load on the primary KDC. Network connectivity should be appropriate for that load. This also points out the need to distribute password expiration as evenly as possible when loading the principal database.

Secondary KDCs

Secondary KDCs should perform the vast majority of the day-to-day work: providing the authentication and ticket-granting services most used by clients. There is a great deal of freedom in the sizing and location of secondary KDCs. User communities of 5,000 to 20,000 are within the performance range of a small to moderate-sized secondary KDC. Availability, not performance requirements, will be the major factor in determining secondary KDC provisioning. Clients should, as a rule, always be directed to a nearby secondary KDC as their first choice. This argues for a greater number of smaller secondary KDCs placed closer to clients.

If availability is a concern, large subnets, campuses, or other major user communities that may be separated by a network failure should have two secondary KDCs, in order to eliminate a single point of failure. Exact physical placement of that secondary pair will be determined by network topology. For example, the pair may be physically distant from each other and still provide a high level of redundancy and availability, depending on the network topology. On the other hand, placing both secondary KDCs on a single network segment that may fail increases cost and does little for redundancy.

If Kerberos is used for local work station access control, availability to the client is critical. If clients and application servers are separated, and if access to those application servers is the predominant factor, then

secondary KDCs should be close to the application servers, and not to the clients. Simply put, if the network between the client and the application server is inoperable, a secondary KDC local to the client will not do much good if the objective is to allow the client to securely communicate with the application server.

Clients and Servers

Client and server platforms will not, as a rule, require any additional resources for Kerberos. However, if large amounts of application data are encrypted, servers may require additional CPU capability or hardware cryptographic accelerators. Encryption of application data does not add any measurable overhead to the network. Additional CPU requirements should scale linearly with the amount of data and will depend on the strength of the cryptographic algorithm, and the key size used. Thus, the additional CPU required to meet the demands of the application can be determined with simple timing tests. If hardware cryptographic accelerators are used, scheduling overhead and key setup time for the accelerator may put an upper bound on performance for small messages. Simple metrics such as the number of bytes per second that can be encrypted or decrypted are not sufficient to determine the real-world performance of hardware accelerators.

Deployment

The appropriate deployment strategy for Kerberos depends both on the intended application and the infrastructure that is in place. Typically, the application will define what demands are placed on Kerberos, and that will, in turn, define the demands on the organization and infrastructure. Other than client software distribution and configuration, those organizational and infrastructure demands are typically the gating factor in any Kerberos deployment. For narrowly focused applications, deployment is generally not an issue and is driven exclusively by the application requirements, with Kerberos simply a component embedded in, and deployed with, that application. For broad-based applications, such as secure single sign-on or enterprise access control, the deployment strategy is typically much more complex. That complexity arises not so much from the technology, but from the more complex and varied organizational and environmental requirements of those deployments.

Deployment stakeholders typically include the user community, security groups, network operations groups, and user administration groups, among others. All will be affected by any large-scale deployment, and all will have a say, directly or indirectly, in a deployment. The introduction of a broad-based security system will, by definition, cross organizational and functional boundaries, and friction is usually the result. If pushed too far and too fast, that deployment friction can generate heat sufficient to incinerate even a well-oiled machine. Unless the organization has a demonstrated need and desire to take big steps, small steps should be the rule. That applies to all security systems.

Successful large-scale deployments tend to be done in two phases: partial infrastructure deployment, followed by incremental client deployment, along with any incremental requirements in the supporting infrastructure. Supporting infrastructure, including any KDCs required for availability and performance, can occur in tandem with deployment of pockets of clients. Alternatively, a KDC "backbone" can be deployed prior to any client deployments.

DNS

The identifier space for DNS should be a concern. Although rationalizing the DNS structure for many organizations was an issue five years ago, it tends to be a much smaller issue now. Because of the growth in TCP/IP and intranets, most organizations have already been forced to deal with that issue over the past years. That said, if the DNS machine name space is chaotic, the DNS structure should be rationalized.

The DNS subdomains that are rationalized must consider the relative locations of clients and services and their interaction. Putting Kerberos into two different subdomains — where clients and servers cross between those subdomains — without first rationalizing the name space in both domains will usually result in problems. Again, this is usually best done incrementally, one subdomain at a time, with rationalization preceding deployment within a subdomain. However, it is not unusual to find that rationalizing one subdomain causes unexpected problems elsewhere. It would be wise to let those perturbations settle before embarking on a Kerberos deployment.

Identities

Typically, the most significant problem encountered in large-scale deployments is rationalizing the identifier spaces for people. Everyone in most organizations has at least one, and typically many more than one, ID.

Rationalizing those spaces in the form of secure single sign-on can itself be the justification for a Kerberos deployment. However, no technology provides a solution to the fundamental problem: people are known by different identities within different and discrete name spaces within the enterprise, and the binding of those multiple identities to a specific individual cannot be known. That problem is the result of years of evolution. Binding of multiple identities to a specific individual can be inferred in some cases. The cost and effort of solving this problem, and level of trust in the resulting environment, depend on the level of assurance provided by that inference.

If there is at least one identifier that is relatively universal, and that identity can be trusted, or there are discrete sets of identifiers with little or no overlap, then the job is much easier. If, on the other hand, the identifier space is chaotic, then more time and energy will be required to rationalize IDs. That time and energy can be due to several factors, including the need to change some names; the need to gain user acceptance when names are changed; and the need to rectify any problems caused by name changes (e.g., systems or applications that are hard-wired with specific names or groups). The actual implementation of the solution is best performed incrementally. This implies an extended deployment, or at least an extended period over which the system is enabled and visible to users. While possible, changing even a relatively small fraction of 100,000 user or system identifiers all at once will likely result in chaos and mass hysteria.

The problem is not eliminated if identity mapping is used to map local identifiers (e.g., a local host or application user ID) to a more uniform identifier, such as a Kerberos principal identifier. Identity mapping may obscure or hide that uniform identifier from users, and thus obviate at least some of the issues with changing identifiers. However, although this approach has an intuitive appeal, it does not eliminate the need for someone or something to go through and map identifiers between different name spaces (the uniform name space being one of those). Building such an "identity map" can be a labor-intensive, time-consuming, and error-prone process. The cost and effort of such a solution should be weighed against the cost and effort in promoting a visible uniform identifier before an approach is selected. Note that Kerberos does not provide implicit capabilities for identifier mapping. Using multiple realms may help but can bring additional issues. Also note that when mapping identities, more-trusted identities should always be used to derive less-trusted identities; less-trusted identities should never be used to derive more-trusted identities.

Enrollment

Even with a rational identifier space, users must still be enrolled in the Kerberos database. That is, the principal database must be populated with the names and the passwords of users. There are several ways of populating the principal database depending on what information is available from existing sources, such as legacy user databases, and the form of that information. Depending on what is available, initially populating the principal database can be either a very trivial or a very significant effort.

If a legacy database exists with IDs and passwords, that legacy database can be used to bulk-load the principal database. That database must have clear-text passwords, or keys that are based on an algorithm that is compatible with Kerberos. If clear-text passwords exist in the legacy database, bulk loading is a simple and straightforward process. If the password algorithm used for the legacy database is incompatible with Kerberos, the keys must be transformed to an algorithm that is acceptable to Kerberos, which can be difficult or impossible, depending on the legacy algorithm used.

If keys that use a standard Kerberos algorithm are unavailable, an alternative is to add support for the legacy algorithms to Kerberos, specifically for the purpose of deployment or initially loading the principal database. This requires creating local-use encryption types within the Kerberos implementation (which the protocol allows for). The Kerberos principal database is then loaded with the existing password values from the legacy databases. Those principal entries would also be flagged to require a change-password operation the first time the user logs in. As part of that change-password operation, the new password would be used to update the principal database entry using a standard Kerberos algorithm. After all users have been registered in this manner, support for the legacy algorithm should be removed.

The use of a legacy algorithm as the basis for initial authentication can reduce the security of the system, and thus its use should be limited to enrollment or deployment. Although this approach may expose a weak derivation of the password on the network, that exposure is limited. Moreover, if clear-text passwords or a weak derivation is currently being used and transmitted across the network, this approach does not make the situation any worse and allows us to rapidly improve the situation. If no legacy databases exist, an existing interface (e.g., the existing login process) can be modified to capture and use passwords to enroll those users

and populate the principal database with their passwords. As a last resort, new passwords/keys can be issued to users.

Realm Design

Other than environmental factors and provisioning requirements discussed previously, the greatest effect on the operation and deployment of a Kerberos implementation will depend on realm design. As always, the rule should be to keep it simple. Unless there is a reason for multiple realms, a single realm should be used. The reasons for using multiple realms might include separation of duties or trust between realms, or the need to distribute the number of primary KDCs (one per realm) for availability of administrative services.

The ability of clients to automatically determine the realm of a service, locate a KDC within a realm, and traverse realms will determine the additional management overhead of a multiple-realm design. If services are available to automate those client needs, multiple realms will not add measurable management overhead. Performance issues due to additional cross-realm authentication operations may also affect the design, but that is usually a distant second behind management overhead. DNS informational records and redirection and referral capability by KDCs can be used to significantly reduce the management overhead of multiple realms. The following discussion assumes that those facilities are unavailable to, or unused by, the Kerberos implementation.

If automated services are not available to mitigate client realm issues, multiple realms should be arranged in a hierarchy, or tree, and that tree should follow the organization's existing DNS domain structure in order to simplify the association of a service name with, or locating a KDC within, a realm. This argues for realms that map directly to each and every subdomain that provides services that clients in other domains (and hence realms) access. This also implies that when a new subdomain is created, a new realm is created as well. This typically implies a large number of realms, which may not be feasible due to the number of KDCs required. An implementation that allows multiple realms to be serviced by a single KDC can mitigate KDC provisioning issues but does not address separation of security or trust, or the availability of a primary KDC.

The key to the success of this strategy is maintaining congruency between realms and DNS domains to whatever depth of the DNS hierarchy is appropriate. This is required in order to minimize the amount of information required by clients and to maximize the amount of information that can be inferred by clients. For example, if congruency to first-level subdomains is appropriate, then each and every first-level subdomain must have a realm; if congruency to second-level subdomains is appropriate, then each and every second-level subdomain must also have a realm. This also implies that creation or removal of a subdomain implies creation or removal of the corresponding realm.

Maintaining realm–domain congruency allows clients to infer a realm implicitly given a DNS name; the client would have to be explicitly told to what depth the realm–domain structure is congruent (e.g., first, second, etc., level of subdomains). Note that this does not provide any information as to the name of a KDC within a realm. KDC-location by clients can be handled using appropriate naming conventions. For example, using KDC's with names such as "kerberos.sub.domain" might be used to locate KDCs within "sub.domain," and implicitly "sub.realm." If secondary KDCs are used, a DNS rotary can be used, or additional conventions such as "kerberos*n*.sub.domain" (where *n* denotes secondary KDCs).

Ongoing Development

This section give\s a snapshot of ongoing development efforts surrounding Kerberos and related technologies. Given the rapid development of security technology today, this discussion can only be illustrative and is by no means complete or definitive.

Standards

This section provides an overview of standards efforts relating to Kerberos. Some of these efforts are ongoing and have not yet been approved by the IETF.

Authorization

Ongoing standards efforts are intended to define commonly used authorization data types for identifying the source of authorization information[15] (for example, to distinguish between client- and KDC-supplied autho-

rization information). This effort is also aimed at standardizing the behavior of servers in the presence, or absence, of certain authorization information.

PKINIT

The Public Key Initial Authentication (PKINIT) effort is designed to standardize the use of Public Key credentials (certificates and key pairs) and asymmetric-key cryptography for authentication as part of the Kerberos initial authentication exchange.[16] Using PKINIT, users with Public Key credentials can gain access to Kerberos services within the enterprise. Simple public–private key pairs, without credentials (i.e., issued by a CA), may also be used. PKINIT uses the preauthentication facility of the initial authentication process to incorporate public key capabilities.

PKCROSS

The Public Key Cross-Realm (PKCROSS) effort is based on the PKINIT effort and is designed to standardize the use of Public Key credentials and asymmetric-key cryptography for cross-realm authentication.[17] PKCROSS allows *ad hoc* and direct trust relationships to be established between different realms, thus eliminating the key management required of current implementations, as well as minimizing trust issues associated with transited realms for clients. This minimizes the need for clients or transited realms to have information about realm topology or relationships.

PKTAPP

Public Key Utilizing Tickets for Application Servers (PKTAPP) allows the use of the Kerberos ticketing mechanism without the requirement for a central KDC.[18] PKTAPP proposes a variation of the PKINIT mechanism for allowing application servers to issue tickets for themselves, instead of having the tickets issued by a KDC.

Related Technologies

These technologies are related to Kerberos or are commonly integrated with, or interact with, Kerberos implementations. As of this writing, all of these technologies have ongoing Kerberos-related development efforts associated with them, either within the standards community or by specific vendors.

Public Key

Public key may describe a system that uses certificates or the underlying public key (i.e., asymmetric-key) cryptography on which such a system is based, or both. A public key system implies asymmetric-key cryptography; asymmetric-key cryptography does not imply a public key system. (By the same token, Kerberos implies support for DES, whereas DES does not imply Kerberos.)

In the traditional public key (PK) model, clients are issued credentials, or "certificates," by a "Certificate Authority" (CA). The CA is a trusted third party. PK certificates contain the user's name, the expiration date of the certificate, etc. The most prevalent certificate format is X.509, which is an international standard. PK certificates typically have lifetimes measured in months or years. Because of the long-lived nature of PK certificates, certificate revocation is a key element in PK infrastructures (PKIs). The authentication process in PK authentication systems also provides the information necessary for a client and server to establish a session key for subsequent data encryption (that is, encryption of application data).

PK credentials, in the form of certificates and public–private key pairs, can provide a strong, distributed authentication system. The private key, which is the most important secret possessed by an individual, runs to hundreds or thousands of bits in length. Thus, a persistent storage system is required to hold the private key, and access to this storage must be protected using a more mundane and conventional mechanism, such as a password. Conventional PK systems still suffer from lack of tools and techniques for managing client credentials. Smart cards hold some promise for secure and mobile private key storage. However, that technology is still relatively new and expensive to deploy on any but a limited scale. Lower-cost solutions, which store the credentials on a local (e.g., work station) disk file, have mobility or security issues. Revocation of PK credentials is still a problem, and standard, scalable and efficient solutions have yet to be provided.

The Kerberos and PK trust models are very similar. A Kerberos ticket is analogous to a PK certificate. However, Kerberos tickets usually have lifetimes measured in hours or days, instead of months or years. Because of their relatively short lifetime, Kerberos tickets are typically allowed to expire instead of being explicitly revoked. The Kerberos session key is analogous to the private key associated with the public key contained in a PK certificate. Possession of the private key is required to prove the authenticity of the sender in a PK system.

That is typically done by signing, or encrypting, information with the private key. That signed or encrypted information, along with the certificate, allows a receiver to verify the association between that information and the certificate. As with Kerberos, the trust the receiver places in the identity of the sender is a function of the trust the receiver places in the issuing authority. In the public key systems, that issuing authority is the certificate authority (CA); in Kerberos, that issuing authority is the KDC.

The use of authentication mechanisms such as public key has the potential for minimizing the need for a central online authentication service such as Kerberos. However, authentication is only one of the functions required of an enterprise security service, and the removal of authentication is unlikely to affect Kerberos' role in supporting access control, authorization, and delegation. Moreover, applications where the performance of asymmetric-key cryptography is unacceptable will still require the use of a system that can provide robust services based on symmetric-key cryptography. Advances in cryptography, such as optimizations of elliptic curve algorithms and hardware acceleration, promise improvements in the performance and cost-effectiveness of asymmetric-key cryptography. When the cost will reach a level that allows wide-scale adoption is unclear. In any case, Kerberos can incorporate that technology today for those who can afford it.

PK systems have been integrated into Kerberos using the preauthentication facility of the initial authentication exchange. For example, the client can provide a signed message, with or without an X.509 certificate, as a preauthentication element in the request to the Kerberos authentication service. The result of that exchange is a standard Kerberos 5 credential.

OSF DCE

The Open Software Foundation, Distributed Computing Environment (OSF DCE) uses Kerberos 5 as the underlying security mechanism.[19] DCE extends the basic Kerberos credential to include other information, such as authorization, and defines an authorization system that is separate but typically co-located with the authentication and ticket-granting services on the DCE security server. DCE clients also use RPC (Remote Procedure Call) as their basic communication mechanism, which requires that both client and server utilize the same secure RPC to be interoperable; the RPC is secured using Kerberos 5.

DCE applications are not interoperable with Kerberos 5 applications. However, many DCE implementations also provide support for standard Kerberos 5 clients. That is, the DCE security server may also provide a standard Kerberos 5 authentication service (AS) and ticket-granting service (TGS). That support for standard Kerberos 5 clients does not make DCE and Kerberos 5 applications interoperable; authorization and RPC transport are still barriers to interoperability between applications. As the term "computing environment" implies, DCE requires additional infrastructure components beyond the basic security service, such as a cell directory service, time service, etc.

Kerberos 4

Kerberos 4 is the predecessor of Kerberos 5. Kerberos 5 addresses many Kerberos 4 security issues, as well as other scalability and portability issues associated with Kerberos 4. Although conceptually similar, Kerberos 5 and Kerberos 4 are quite different. Kerberos 4 has seen fairly extensive use in educational and commercial environments, and in a few key applications. One of the most widely used applications is AFS (Andrew File System), which is a secure distributed file system (similar to the OSF DCE distributed file service, DFS).

Kerberos 5 and Kerberos 4 applications are not interoperable. Some Kerberos 5 implementations also include support for Kerberos 4 and provide facilities to improve interoperation between Kerberos 4 and Kerberos 5 environments. Interoperation may be achieved by direct support for Kerberos 4 authentication and ticket-granting services by the KDC, or by allowing a Kerberos 4 ticket to be used to obtain a Kerberos 5 ticket (or vice versa).

GSS-API

The Generic Security Service Applications Programming Interface (GSS-API) is a standard that provides applications with a standard API for using different security mechanisms. The objective of the GSS-API is to shield applications from variations in the underlying security mechanisms. In its simplest form, the GSS-API is a thin veneer that sits above an underlying mechanism; that mechanism, such as Kerberos 5, provides the actual security services. Although applications are shielded from the underlying mechanism, the infrastructure for each security mechanism is still required.

The original GSS-API specification is referred to as V1.[20] V1 of the GSS-API does not support mechanism negotiation. V2 of the GSS-API specification provides the ability for implementations to support multiple mechanisms.[21] As an API, the GSS-API must define specific language bindings, and there are separate standards

for each language binding, such as Java.[22] As of this writing, only "C" language bindings are standardized.[23] GSS-API mechanism specifications may also encapsulate existing mechanisms, in which case a protocol, and not just an API, is defined as part of the GSS-API mechanism standard.

Kerberos 5 was one of the first mechanisms implemented under the GSS-API. Several other mechanisms have also been implemented, including SPKM[24] (Simple Public Key Mechanism) and IDUP[25] (Independent Data Unit Protocol). Two GSS-API applications are compatible only if the underlying GSS-API mechanisms are compatible. GSS-API applications using a Kerberos 5 mechanism and "native" Kerberos 5 applications are not interoperable, because the GSS-API defines not only an API, but a protocol as well.[26] Although the GSS-API Kerberos 5 mechanism uses messages that are the same as Kerberos 5, those messages are encapsulated in a protocol that is different from Kerberos 5.

Microsoft SSPI

The Microsoft Security Service Provider Interface (SSPI) is the Microsoft equivalent of the GSS-API.[27] A mechanism such as Kerberos 5 is a "security provider," and applications use security providers through the "provider interface" (the API). The SSPI Kerberos 5 mechanism is wire-level compatible with the GSS-API Kerberos 5 mechanism. The SSPI API is not compatible with the GSS-API. Thus, although the APIs differ, clients and servers written to use either SSPI or GSS-API can interoperate using a common Kerberos 5 mechanism.

SNEGO

The Simple and Protected GSS-API Negotiation Mechanism (SNEGO), is a special GSS-API mechanism that allows the secure negotiation of the mechanism to be used by two different GSS-API implementations.[28] In essence, SNEGO defines a universal but separate mechanism, solely for the purpose of negotiating the use of other security mechanisms. SNEGO itself does not define or provide authentication or data protection, although it can allow negotiators to determine if the negotiation has been subverted, once a mechanism is established. GSS-API implementations that do not support SNEGO cannot negotiate, and therefore the client and server must agree *a priori* what mechanism or mechanisms will be used.

SSL

Secure Sockets Layer (SSL), and the related Transport Layer Security (TLS), are secure point-to-point protocols that define both authentication and message confidentiality protection.[29] SSL uses public key authentication. Because SSL is point-to-point, it is suitable only as a low-level transport protocol. An SSL authentication exchange results in the establishment of a shared secret key on both the client and server. That key, and conventional symmetric-key cryptography, is used to provide message confidentiality protection.

SSL has also been used to provide an initial authentication exchange between a client and a Kerberos KDC. In essence, SSL is used to replace the standard Kerberos initial authentication exchange, and a special authentication service (AS) is used on the KDC. SSL authentication is used in place of the client's initial authentication request, which may or may not involve the use of a password by the client. SSL is then used to securely transport the TGT back to the client. SSL is presently one of the few protocols that do not have a standard way of integrating Kerberos authentication to provide message integrity and confidentiality, although such integration has been proposed.[30]

SASL

Simple Authentication and Security Layer (SASL) is a framework for negotiating a security mechanism for session-oriented protocols.[31] SASL specifies a naming convention for registered mechanisms, as well as profile information required for clients and servers to use a mechanism to protect a specific protocol. Registered SASL mechanisms include Kerberos 4 and GSS-API, among others.

IPSec

Internet Protocol Security (IPSec), provides integrity or confidentiality services at the network layer.[32] All data protection is performed using symmetric-key cryptography. Establishment of the session keys for data protection is also defined by IPSec, and may use both symmetric- and asymmetric-key cryptography.

Although IPSec provides data protection, it does not provide the key management infrastructure necessary for a large number of IPSec systems to authenticate and establish the session keys needed for data protection. As a network layer protection service, IPSec is targeted primarily at machine-to-machine security; authentication of individuals and applications is outside the scope of IPSec, and depends entirely on the key manage-

ment infrastructure used, and the integration of that key management infrastructure with the IPSec implementation.

Kerberos can provide key management for IPSec implementations, and this has been proposed through the use of the GSS-API mechanism.[33] In essence, the Kerberos principals are simply machines, or more accurately, the service on each machine that provides IPSec network layer protection. Kerberos can also provide the key management for binding individuals and applications to IPSec implementations.

RADIUS

The Remote Authentication Dial-In User Service (RADIUS) allows a RADIUS client (typically a network access device, such as a terminal server), to authenticate a user on a remote computer and control that user's access to the network.[34] The RADIUS client uses the RADIUS protocol to talk to a RADIUS server to authenticate the user. The RADIUS server may contain a simple database containing IDs and passwords, or may use another server to authenticate the client, such as a token card server, or a Kerberos KDC. RADIUS has gained significant acceptance among network and token card vendors.

RADIUS protects the communication between a RADIUS client (e.g., a terminal server), and a RADIUS server. RADIUS does not protect the communications between a remote client and a RADIUS client. Thus, information passed between the remote client (e.g., a laptop computer) and the RADIUS client is unprotected. RADIUS does not have the concept of a credential, and the result of authentication using RADIUS is a yes–no answer. Thus, RADIUS is primarily used as a simple access control mechanism. DIAMETER, part of the AAA (Authentication, Authorization, and Accounting) effort in the IETF, is working to address some of the limitations of RADIUS.[35]

RADIUS has been integrated with Kerberos by using the RADIUS server as a surrogate Kerberos client. That is, the RADIUS server acts as a client to verify an ID and password against a KDC; that ID and password come from the end user at the remote computer system. Although the RADIUS server obtains a Kerberos credential as the result of that authentication, there is no way to send that credential back to the end client through the RADIUS client. The benefit of using RADIUS in this manner is that a single authentication database can be used (the KDC's principal database), even though the result of authentication does not provide the client a credential. Note that RADIUS does not protect the user's password between the end client and the RADIUS, and the RADIUS client and server have access to the user's Kerberos ID and password. Thus, use of RADIUS as part of a Kerberos implementation should ensure that the resulting exposure is acceptable.

CDSA

Common Data Security Architecture (CDSA) provides a standard API for many security services, including encryption, authentication, and credential storage and management.[36] CDSA also defines standard methods for incorporating a variety of security service providers, both hardware and software, and a variety of mechanisms, including public key and biometrics. CDSA is similar to Microsoft's Cryptographic API (MS CAPI) in purpose. CDSA was originally developed by Intel and has now been adopted by the Open Group.[37]

Token Cards

Token cards are an example of a very simple trusted third party authentication system. A user, in possession of a token, keys in information from the token. That information is then sent to the application, which verifies the information with a token card server (the trusted third party) provided by the token card vendor. Typically, the value presented by the token is usable only once (to prevent replays) or has a very limited life, and is generated using a key contained within the token card (which is tamper-proof) and a key known to the vendor's token card server.

Token cards secure only the authentication to the application and do not provide any security for the application's data. That is, no information in the authentication process is available for establishing a session key for subsequently encrypting application data. Moreover, token cards must be used for authentication to each application, just as a password is. While the user is not required to remember passwords — the token card in effect generates the passwords — the user must still key a "password" in for each application authentication.

There are three basic types of token cards: challenge–response, time synchronous, and event synchronous. Regardless of type, all have a common attribute: the card is (or should be) tamper-proof, and the card contains a secret key shared between the card and the security server. Use of the card typically requires both physical possession of the card (something you have) and a PIN (something you know). The requirement that those two factors be present for authentication to succeed is the basis for the term "two-factor authentication."

Software may also be used to achieve the same effect as a hardware token card. Obviously a software "token card" does not provide the two factors provided by a hardware token.

A variety of token card systems have been integrated into Kerberos using the preauthentication facility of the initial authentication service. The KDC then contacts the token card server, instead of the client contacting the token card server. This allows a mix of token card technologies to be used. The result of the initial authentication exchange is a standard Kerberos 5 credential.

Smart Cards

Smart cards are so named because they have processing intelligence on a card that is the same form factor as a credit card. The processing power and memory capacity varies depending on the card. Smart cards have received prominent attention recently, primarily because of the promise they hold for addressing public key client credential management and security issues, by holding the user's private key in tamper-proof storage, and performing cryptographic operations on the card. Thus, the user's private key never leaves the card.

Smart card costs are dropping rapidly. However, a wide-scale smart card deployment requires not only cards, but also readers. As of this writing, cards with the necessary processing power and storage, and the associated readers, are still too expensive for wide-scale deployment. Although smart cards are most often associated with public key systems, smart cards are also used to provide symmetric-key cryptography. Symmetric-key smart cards may provide secure key storage and associated cryptographic functions for use as challenge–response devices, for example.

Public key smart cards have been integrated into Kerberos using the preauthentication mechanism. This allows users with smart cards to authenticate to the Kerberos authentication service using the public key credentials on a smart card.

Encryption Algorithms

The two broad classifications of cryptographic systems are symmetric-key and asymmetric-key. Both Kerberos and public key systems (as well as other authentication systems) may incorporate one or both cryptographic systems. Common symmetric-key systems include DES (Data Encryption Standard), and the triple-DES variant.[38] Common asymmetric-key systems include ECC[39] (elliptic curve) and RSA[40] (Rivest–Shamir–Adleman). The strength of these different systems is difficult to compare and is only one element that determines their application. For example, based on exhaustive key search, a triple-DES (112-bit) key is approximately equal to a 1792-bit RSA key (i.e., key modulus);[41] and a 1024-bit RSA key is approximately equal to a 160-bit ECC key.[42]

The distinguishing characteristic of these systems is the symmetry of the keys used for encryption and decryption. Symmetric-key systems use the same key for encryption and decryption. Thus, two parties must share the same key (presumably secret) in order to encrypt and decrypt information. Asymmetric-key systems use different, but related, keys for encryption and decryption: information encrypted with one key can only be decrypted with the other key. That key pair is typically referred to as a public–private key pair. One of the keys is public and known to many people; the other key is private (presumably secret) and known to only one person.

Another distinguishing characteristic of these systems is the CPU speed or hardware complexity for encryption and decryption operations. Symmetric-key systems tend to be quite fast. Asymmetric-key systems tend to be CPU intensive and are typically used only for encrypting small amounts of data — typically only that needed for authentication (as with digital signatures). Because of its speed advantages, symmetric key cryptography is still used by all security systems for encrypting application data. Symmetric- and asymmetric-key are often used together. For example, asymmetric-key is used to establish a session key for symmetric-key by encrypting a symmetric session key (that symmetric-key usually being a very a small amount of data). Higher-performance symmetric-key is then used to encrypt and decrypt the application data. The speed of cryptographic operations in symmetric-key systems is typically symmetric. That is, encrypt and decrypt speeds are generally the same (for the same implementation running on the same hardware). The speed of cryptographic operations in asymmetric-key systems is typically asymmetric, and depends on what function is being performed.

Cryptographic systems alone do not constitute a secure authentication system. Kerberos and public key are secure, distributed, authentication systems that use cryptographic systems, define the rules of how cryptography is used, and that define the syntax and semantics for various protocol messages and data formats. Although the rules and protocols for different authentication systems tend to be very different, the problems that must be solved to build a practical, secure, distributed, authentication system are largely invariant.

Kerberos defines the use of symmetric-key cryptography, including both DES and triple-DES, for both authentication and data encryption. Asymmetric-key cryptography has also been integrated into Kerberos using the preauthentication facility of the initial authentication service.

Secure Hash Algorithms

Secure distributed authentication systems require secure hash functions and not just encryption and decryption, although secure hash functions are often built using a cryptographic algorithm. A secure hash function takes a large amount of data and hashes it down to a small amount of data (e.g., 128 bits), or the "hash value." The attributes of a secure hash function are no two inputs should produce the same output ("collision proof"), and you cannot work backwards from the hash value to the input. Think of the secure hash value as a fingerprint: the hash value uniquely defines the input but does not tell you anything about the input. Note that a simple checksum, such as CRC32, is not a secure hash function — too many inputs produce the same output. A secure hash is sometimes referred to as a message digest or cryptographic checksum.

A secure hash is typically used to provide integrity protection and is also used in digital signature applications. The hash value of a document is generated, and that value is encrypted using an individual's key. Encrypting only the hash value, or signature, eliminates the need to encrypt the entire document for integrity protection. That encrypted value is also the digital signature of the individual applied to a document. Verifying the signature against the document simply regenerates the hash value of the document, decrypts the encrypted hash value, and compares the two. If someone changes either the signature or the document, the hash will change, and verification will fail. The most common hash functions are MD5[43] (Message Digest 5) and SHA-1[44] (Secure Hash Algorithm 1).

Kerberos defines the use of several secure hash functions, including DES and triple-DES message authentication code (MAC) hashing functions, as well as MD5 and SHA-1.

Lessons Learned

As discussed in previous sections, most of the technical issues surrounding the implementation and deployment of Kerberos are tractable, and when properly understood, those issues should not present serious problems. The significant technical issues that remain — such as fragmented or dysfunctional namespaces — and their solutions are dependent on the environment. Various methods can minimize those issues, but there is little that Kerberos, or any security system, can do to fix the underlying problems. And as with all security systems, the primary obstacles to success are not technical, but fundamental to the role of information security in today's business and organizational environments. Kerberos does what it can technically by providing a robust and cost-effective distributed security system. The rest is up to us.

Risk, Fear, and Value

Kerberos is fundamentally a strong distributed authentication system. It can be used for a single application within a single group or a set of applications that span an enterprise. Whatever the use, successful deployments usually address applications that can benefit from what Kerberos has to offer. That applies whether Kerberos is being used for a single application or to implement enterprise wide secure single sign-on. As obvious as it may seem, the security that Kerberos brings with it must be perceived to be of value to the organization. Although security practitioners may appreciate the intrinsic value of strong authentication, the broader community within most organizations generally does not perceive that value. Without perceived value, cost and effort will be viewed as wasted. To put it another way, without perceived value, any deployment problems will be magnified, and the probability of success will rapidly approach zero.

Applications that can benefit from a distributed security system such as Kerberos are growing more common than in the past. However, the fundamentals still hold true. As enterprises move to more distributed environments, services are often pushed out toward the consumer. For example, providing on-demand access to human resources data (typically some of the most sensitive information in an organization) by employees from individual desktops. Such "self-service" applications require a strong, distributed authentication system that can also provide data encryption, and provide those capabilities at reasonable cost. The cost of the security infrastructure can often be justified by the cost savings obtained by removing the "human firewall" of clerks that typically guard access to those applications' data.

Because the intrinsic value of a system such as Kerberos is not always appreciated, it is up to security practitioners to identify the applications that can benefit. That requires more than an understanding of security. It also requires understanding the application, and the business needs that surround the application. It requires knowledge sufficient to make the benefits of security intrinsically obvious to the application owners, or sufficient knowledge to quantify the risks and costs to the application owners. Risk and cost are a business decision. Making an informed decision requires understanding both. Risk is often difficult to quantify, and unquantified risk, in the form of fear, can sometimes be a great motivator. However, decisions based on fear are often subject to reversal and second-guessing, and are poor substitutes for informed decision making.

Security based on value and informed decisions will find a more accepting audience, and much easier deployment, than those based on fear.

Distributed Security

The rules that a security system enforces represent demands and assumptions made of the environment. If those rules are too onerous, the security implementation will fail as predictably, and for the same reasons, as any technology that makes unrealistic assumptions or resource demands on its environment. As a security *technology*, Kerberos provides very good performance and makes relatively modest demands and assumptions on its environment. As a security *system*, the demands and assumptions made by Kerberos are entirely dependent on an organization's definition of acceptable security.

The tradeoff between acceptable security and what is practical in an organization, is the first question that the security practitioner must answer. The answer to that question varies from organization to organization, and technology generally plays a minor role in the equation. Moreover, the organic nature of most distributed environments is not receptive to the introduction of a broad-based security system. Introduction of such a system into those environments — with implicitly greater uniformity and rigidity — will cause friction. If Kerberos is used to enforce draconian security measures in environments that have previously had very informal or isolated security practices, problems are very likely to occur. Technology cannot solve those problems.

The very nature of distributed environments increases diversity and indeterminacy. That introduces a greater degree of uncertainty into the security equation. That uncertainty is something the security community has historically been very uncomfortable with. Probabilistic models of security require quantification and analysis. Today, that quantification and analysis are extremely difficult at best, impossible at worst, and so rare as to be nonexistent. Thus we are left to make a value judgment, and for most it is far easier to retreat into the absolutes of the past than to risk uncertainty. After all, risk reduction and aversion is what security is all about.

While the level of certainty that we are historically accustomed to is achievable in distributed environments, it is not achievable at a cost that any organization can afford. That is extremely unlikely to change. Diversity and indeterminacy are increasing with every passing day. Successful distributed security implementations recognize and embrace those changes, making incremental improvements as organizations and technology adapt and converge on an acceptable paradigm. Unsuccessful distributed security implementations shun those changes and attempt to impose unrealistic demands based on time-worn assumptions about what is feasible, necessary, or desirable.

The one lesson that stands out from years of Kerberos implementations is that uncertainty is a fact of life in distributed security. Learn to deal with it.

Notes

1. Project Athena is a model of "next-generation distributed computing" in the academic environment. It began in 1993 as an eight-year project with DEC and IBM as its major industrial) sponsors. Their pioneering model is based on client–server technology and it includes such innovations as authentication based on Kerberos and X Windows. An excellent reference — George Champine, *MIT Project Athena, A Model for Distributed Campus Computing*, Digital Press, 1991. Other definitive works on Kerberos include B. Clifford Neuman and Theodore Ts'o, Kerberos: an authentication service for computer networks, *IEEE Communications,* 32(9):33-38. September 1994; available at http://gost.isi.edu/publications/kerberos-neuman-tso.html and http://nii.isi.edu/publications/kerberos-neuman-tso.html.
2. R. Needham and M. Schroeder, Using encryption for authentication in large networks of computers, *Communications of the ACM 21*, December 1978.

3. D.E. Denning and G.M. Sacco, Time-stamps in key distribution protocols, *Communications of the ACM 24*, August 1981.
4. J. Kohl and C. Neuman, The Kerberos Network Authentication Service(V5), Internet Request for Comments 1510, September 1993. http://www.rfc-editor.org.
5. Current revisions to the Kerberos protocol can be found in C. Neuman, J. Kohl and T. Ts'o, "The Kerberos Network Authentication Service (V5)," Internet Draft, November 1998.
6. IETF RFC information can be found at various Internet sites. The reference sites are ds.internic.net (US East Coast), nic.nordu.net (Europe), ftp.isi.edu (US West Coast), and munnari.oz.au (Pacific Rim).
7. Microsoft Corporation, "Microsoft Windows 2000 Product Line Summary," http://www.microsoft.com/presspass/features/1998/winntproducts.htm.
8. Sun Microsystems, "Sun Enterprise Authentication Mechanism for Solaris Enterprise Server Datasheet," http://www.sun.com/solaris/ds/ds-seamss.
9. B. Blakley, "Security Requirements for DCE", Open Software Foundation Request for Comments 8.1, October 1995.
10. S. M. Bellovin and M. Merritt, Limitations of the Kerberos authentication system, *Proceedings of the Winter 1991 Usenix Conference*, January 1991.
11. B. Clifford Neuman, Proxy-based authorization and accounting for distributed systems, in *Proceedings of the 13th International Conference on Distributed Computing Systems*, Pittsburgh, May 1993.
12. In his treatise on distributed systems security, Morrie Gasser categorizes the security services that a distributed system can provide for its users and applications as: secure channels, authentication, confidentiality, integrity, access control. nonrepudiation, and availability. M. Gasser, Security in distributed systems, in *Recent Developments in Telecommunications*, North-Holland, Amsterdam, The Netherlands, Elsevier Science Publishers, 1992.
13. J. Pato, "Using Pre-Authentication to Avoid Password Guessing Attacks," Open Software Foundation DCE Request for Comments 26, December 1992.
14. See Reference 11.
15. C. Neuman, J. Kohl, T. Ts'o, "The Kerberos Network Authentication Service (V5)," Internet Draft, November 1998.
16. C. Neuman, J. Wray, B. Tung, J. Trostle, M. Hur, A. Medvinsky, and S. Medvinsky, "Public Key Cryptography for Initial Authentication in Kerberos," Internet Draft, November 1998.
17. G. Tsudik, C. Neuman, B. Sommerfeld, B. Tung, M. Hur, T. Ryutov, and A. Medvinsky, "Public Key Cryptography for Cross-Realm Authentication in Kerberos," Internet Draft, November 1998.
18. C. Neuman, M. Hur, A. Medvinsky, Alexander Medvinsky, "Public Key Utilizing Tickets for Application Servers (PKTAPP)," Internet Draft, March 1998. See also: M. Sirbu, J. Chuang. "Distributed Authentication in Kerberos Using Public Key Cryptography," Symposium On Network and Distributed System Security, 1997.
19. B. Blakley, "Security Requirements for DCE," Open Software Foundation Request for Comments 8.1, October 1995.
20. J. Linn, "Generic Security Service Application Program Interface," Internet Request for Comments 1508, September 1993. http://www.rfc-editor.org
21. J. Linn, "Generic Security Service Application Program Interface, Version 2," Internet Request for Comments 2078 (January 1997). http://www.rfc-editor.org
22. J. Kabat, "Generic Security Service API Version 2: Java bindings," Internet Draft, August 1998.
23. J. Wray, "Generic Security Service API: C-bindings," Internet Request for Comments 1509, September 1993. http://www.rfc-editor.org
24. C. Adams, "The Simple Public-Key GSS-API Mechanism (SPKM)," Internet Request for Comments 2025, October 1996. http://www.rfc-editor.org
25. C. Adams, "Independent Data Unit Protection Generic Security Service Application Program Interface (IDUP-GSS-API)," Internet Request for Comments 2479, December 1998. http://www.rfc-editor.org
26. J. Linn, "The Kerberos Version 5 GSS-API Mechanism," Internet Request for Comments 1964, June 1996.
27. D. Chappell, NT 5.0 in the enterprise, *Byte Magazine*, May 1997.
28. E. Baize, D. Pinkas, "The Simple and Protected GSS-API Negotiation Mechanism," Internet Request for Comments 2478, December 1998. http://www.rfc-editor.org
29. T. Dierks, C. Allen, "The TLS Protocol Version 1.0," Internet Request for Comments 2246, January 1999. http://www.rfc-editor.org

30. M. Hur, A. Medvinsky, "Addition of Kerberos Cipher Suites to Transport Layer Security (TLS)," Internet Draft, September 1998.
31. J. Myers, "Simple Authentication and Security Layer (SASL)," Internet Request for Comments 2222, October 1997. http://www.rfc-editor.org
32. R. Thayer, N. Doraswamy, R. Glenn, "IP Security Document Roadmap," Internet Request for Comments 2411, November 1998. http://www.rfc-editor.org
33. D. Piper, "A GSS-API Authentication Mode for IKE", Internet Draft, December 1998.
34. C. Rigney, A. Rubens, W. Simpson, S. Willens. "Remote Authentication Dial In User Service (RADIUS)," Internet Request for Comments 2138, April 1997. http://www.rfc-editor.org
35. A. Rubens, P. Calhoun, "DIAMETER Base Protocol," Internet Draft, November 1998.
36. Intel Corporation, "Making PC Interaction Trustworthy for Communications, Commerce and Content," Intel Security Program, July 1998.
37. The Open Group, "New Security Standard from The Open Group Brings the Realization of High-Value E-Commerce for Everyone a Step Further" Press Release January 6, 1998.
38. National Bureau of Standards, U.S. Department of Commerce, "Data Encryption Standard (DES)," Federal Information Processing Standards Publication 46-2, Washington, DC (December 1993). National Bureau of Standards, U.S. Department of Commerce, "DES Modes of operation," Federal Information Processing Standards Publication 81 (December 1980). Information on triple-DES can be found in: National Institute of Standards and Technology, U.S. Department of Commerce, "Data Encryption Standard (DES)," Draft Federal Information Processing Standards Publication 46-3, (January 1999).
39. V.S. Miller, Use of elliptic curves in cryptography, *Advances in Cryptology — Proceedings of CRYPTO85,* (Springer Verlag Lecture Notes in Computer Science 218, pp. 417-426, 1986). For a more contemporary treatment, see: Jurisic and A.J. Menezes, Elliptic curves and cryptography, *Dr. Dobb's Journal*, pp. 26-35, (April 1997).
40. R.L. Rivest, A. Shamir, and L.M. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM 21*, February 1978.
41. B. Schneier, *Applied Cryptography*, John Wiley & Sons, New York, 1996.
42. "Remarks on the Security of the Elliptic Curve Cryptosystem," Certicom Corporation ECC whitepaper (September 1997).
43. R. Rivest, "The MD5 Message Digest Algorithm," Internet Request for Comments 1321, MIT Laboratory for Computer Science, April 1992.
44. National Institute of Standards and Technology, U.S. Department of Commerce, "Secure Hash Standard (SHS)," Federal Information Processing Standard Publication 180-1, April 1995.

119

Methods of Attacking and Defending Cryptosystems

Joost Houwen, CISSP

Encryption technologies have been used for thousands of years and, thus, being able read the secrets they are protecting has always been of great interest. As the value of our secrets have increased, so have the technological innovations used to protect them. One of the key goals of those who want to keep secrets is to keep ahead of techniques used by their attackers. For today's IT systems, there is increased interest in safeguarding company and personal information, and therefore the use of cryptography is growing. Many software vendors have responded to these demands and are providing encryption functions, software, and hardware. Unfortunately, many of these products may not be providing the protection that the vendors are claiming or customers are expecting. Also, as with most crypto usage throughout history, people tend to defeat much of the protection afforded by the technology through misuse or inappropriate use. Therefore, the use of cryptography must be appropriate to the required goals and this strategy must be constantly reassessed. To use cryptography correctly, the weaknesses of systems must be understood.

This chapter reviews various historical, theoretical, and modern methods of attacking cryptographic systems. Although some technical discussion is provided, this chapter is intended for a general information technology and security audience.

Cryptography Overview

A brief overview of definitions and basic concepts is in order at this point. Generally, *cryptography* refers to the study of the techniques and methods used to hide data, and *encryption* is the process of disguising a message so that its meaning is not obvious. Similarly, decryption is the reverse process of encryption. The original data is called *cleartext* or *plaintext,* and the encrypted data is called *ciphertext*. Sometimes, the words *encode*/*encipher* and *decode*/*decipher* are used in the place of *encrypt* and *decrypt*. A cryptographic algorithm is commonly called a *cipher*. *Cryptanalysis* is the science of breaking cryptography, thereby gaining knowledge about the plaintext. The amount of work required to break an encrypted message or mechanism is call the *work factor*. *Cryptology* refers to the combined disciplines of cryptography and cryptanalysis.

Cryptography is one of the tools used in information security to assist in ensuring the primary goals of confidentiality, integrity, authentication, and non-repudiation.

Some of the things a cryptanalyst needs to be successful are:

- Enough ciphertext
- Full or partial plaintext
- Known algorithm
- Strong mathematical background
- Creativity

- Time, time, and more time for analysis
- Large amounts of computing power

Motivations for a cryptanalyst to attack a cryptosystem include:

- Financial gain, including credit card and banking information
- Political or espionage
- Interception or modification of e-mail
- Covering up another attack
- Revenge
- Embarrassment of vendor (potentially to get them to fix problems)
- Peer or open-source review
- Fun/education (cryptographers learn from others' and their own mistakes)

It is important to review the basic types of commonly used ciphers and some historical examples of cryptosystems. The reader is strongly encouraged to review cryptography books, but especially Bruce Schneier's essential *Applied Cryptography*[1] and *Cryptography and Network Security*[2] by William Stallings.

Cipher Types

Substitution Ciphers

A simple yet highly effective technique for hiding text is the use of substitution cipher, where each character is switched with another. There are several of these types of ciphers with which the reader should be familiar.

Monoalphabetic Ciphers

One way to create a substitution cipher is to switch around the alphabet used in the plaintext message. This could involve shifting the alphabet used by a few positions or something more complex. Perhaps the most famous example of such a cipher is the Caesar cipher, used by Julius Caesar to send secret messages. This cipher involves shifting each letter in the alphabet by three positions, so that "A" becomes "D," and "B" is replaced by "E," etc. Although this may seems simple today, it is believed to have been very successful in ancient Rome. This is probably due, in large part, to the fact the even the ability to read was uncommon, and therefore writing was probably a code in itself.

A more modern example of the use of this type of cipher is the UNIX *crypt* utility, which uses the ROT13 algorithm. ROT13 shifts the alphabet 13 places, so that "A" is replaced by "N," "B" by "M," etc. Obviously, this cipher provides little protection and is mostly used for obscuration rather than encryption, although with a utility named *crypt*, some users may assume there is actually some real protection in place. Note that this utility should not be confused with the UNIX *crypt( )* software routine that is used in the encryption of passwords in the password file. This routine uses the repeated application of the DES algorithm to make decrypting these passwords extremely difficult.[3]

Polyalphabetic Ciphers

By using more than one substitution cipher (alphabet), one can obtain improved protection from a frequency analysis attack. These types of ciphers were successfully used in the American Civil War[4] and have been used in commercial word-processing software. Another example of this type of cipher is the Vigenère cipher, which uses 26 Caesar ciphers that are shifted. This cipher is interesting as well because it uses a keyword to encode and decode the text.

One-Time Pad

In 1917, Joseph Mauborgne and Gilbert Vernam invented the unbreakable cipher called a one-time pad. The concept is quite effective, yet really simple. Using a random set of characters as long as the message, it is possible to generate ciphertext that is also random and therefore unbreakable even by brute-force attacks. In practice, having — and protecting — shared suitably random data is difficult to manage but this technique has been

successfully used for a variety of applications. It should be understood by the reader that a true, and thus unbreakable, one-time pad encryption scheme is essentially a theoretical concept as it is dependent on true random data, which is very difficult to obtain.

Transposition Cipher

This technique generates ciphertext by performing some form of permutation on plaintext characters. One example of this technique is to arrange the plaintext into a matrix and perform permutations on the columns. The effectiveness of this technique is greatly enhanced by applying it multiple times.

Stream Cipher

When large amounts of data need to enciphered, a cipher must be used multiple times. To efficiently encode this data, a stream is required. A stream cipher uses a secret key and then accepts a stream of plaintext producing the required ciphertext.

Rotor Machines

Large numbers of computations using ciphers can be time-consuming and prone to errors. Therefore, in the 1920s, mechanical devices called rotors were developed. The rotors were mechanical wheels that performed the required substitutions automatically. One example of a rotor machine is the Enigma used by the Germans during World War II. The initial designs used three rotors and an operator plugboard. After the early models were broken by Polish cryptanalysts, the Germans improved the system only to have it broken by the British.

RC4

Another popular stream cipher is the Rivest Cipher #4 (RC4) developed by Ron Rivest for RSA.

Block Cipher

A block cipher takes a block of plaintext, a key, and produces a block of ciphertext. Current block ciphers produce ciphertext blocks that are the same size as the corresponding plaintext block.

DES

The Data Encryption Standard (DES) was developed by IBM for the National Institute of Standards and Technology (NIST) as Federal Information Processing Standard (FIPS) 46. Data is encrypted using a 56-bit key and 8 parity bits with 64-bit blocks.

3DES

To improve the strength of DES-encrypted data, the algorithm can be applied in the triple-DES form. In this algorithm, the DES algorithm is applied three times, either using two keys (112-bit) encrypt-decrypt-encrypt, or using three keys (168-bit) encrypt-encrypt-encrypt modes. Both forms of 3DES are considered much stronger than single DES. There have been no reports of breaking 3DES.

IDEA

The International Data Encryption Algorithm (IDEA) is another block cipher developed in Europe. This algorithm uses 128-bit keys to encrypt 64-bit data blocks. IDEA is used in Pretty Good Privacy (PGP) for data encryption.

Types of Keys

Most algorithms use some form of secret key to perform encryption functions. There are some differences in these keys that should be discussed.

1. *Private/Symmetric.* A private, or symmetric, key is a secret key that is shared between the sender and receiver of the messages. This key is usually the only key that can decipher the message.
2. *Public/Asymmetric.* A public, or asymmetric, key is one that is made publicly available and can be used to encrypt data that only the holder of the uniquely and mathematically related private key can decrypt.

3. *Data/Session.* A symmetric key, which may or may not be random or reused, is used for encrypting data. This key is often negotiated using standard protocols or sent in a protected manner using secret public or private keys.
4. *Key Encrypting.* Keys that are used to protect data encrypting keys. These keys are usually used only for key updates and not data encryption.
5. *Split Keys.* To protect against intentional or unintentional key disclosure, it is possible to create and distribute parts of larger keys which only together can be used for encryption or decryption.

Symmetric Key Cryptography

Symmetric key cryptography refers to the use of a shared secret key that is used to encrypt and decrypt the plaintext. Hence, this method is sometimes referred to as secret key cryptography. In practice, this method is obviously dependent on the "secret" remaining so. In most cases, there needs to be a way that new and updated secret keys can be transferred. Some examples of symmetric key cryptography include DES, IDEA, and RC4.

Asymmetric Key Cryptography

Asymmetric key cryptography refers to the use of public and private key pairs, and hence this method is commonly referred to as public key encryption. The public and private keys are mathematically related so that only the private key can be used to decrypt data encrypted with the public key. The public key can also be used to validate cryptographic signatures generated using the corresponding private key.

Examples of Public Key Cryptography

RSA

This algorithm was named after its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman, and based on the difficulty in factoring large prime numbers. RSA is currently the most popular public key encryption algorithm and has been extensively cryptanalyzed. The algorithm can be used for both data encryption and digital signatures.

Elliptic Curve Cryptography (ECC)

ECC utilizes the unique mathematical properties of elliptic curves to generate a unique key pair. To break the ECC cryptography, one must attack the "elliptic curve discrete logarithm problem." Some of the potential benefits of ECC are that it uses significantly shorter key lengths and that is well-suited for low bandwidth/CPU systems.

Hash Algorithms

Hash or digest functions generate a fixed-length hash value from arbitrary-length data. This is usually a one-way process, so that it impossible to reconstruct the original data from the hash. More importantly, it is, in general, extremely difficult to obtain the same hash from two different data sources. Therefore, these types of functions are extremely useful for integrity checking and the creation of electronic signatures or fingerprints.

MD5

The Message Digest (MD) format is probably the most common hash function in use today. This function was developed by Ron Rivest at RSA, and is commonly used as a data integrity checking tool, such as in Tripwire and other products. MD5 generates a 128-bit hash.

SHA

The Secure Hash Algorithm (SHA) was developed by the NSA. The algorithm is used by PGP, and other products, to generate digital signatures. SHA produces a 160-bit hash.

Steganography

Steganography is the practice used to conceal the existence of messages. That is different from encryption, which seeks to make the messages unintelligible to others.[5]

A detailed discussion of this topic is outside the scope of this chapter, but the reader should be aware that there are many techniques and software packages available that can be used to hide information in a variety of digital data.

Key Distribution

One of the fundamental problems with encryption technology is the distribution of keys. In the case of symmetric cryptography, a shared secret key must be securely transmitted to users. Even in the case of public key cryptography, getting private keys to users and keeping public keys up-to-date and protected remain difficult problems. There are a variety of key distribution and exchange methods that can be used. These range from manual paper delivery to fully automated key exchanges. The reader is advised to consult the references for further information.

Key Management

Another important issue for information security professionals to consider is the need for proper key management. This is an area of cryptography that is often overlooked and there are many historical precedents in North America and other parts of the world. If an attacker can easily, or inexpensively, obtain cryptographic keys through people or unprotected systems, there is no need to break the cryptography the hard way.

Public versus Proprietary Algorithms and Systems

It is generally an accepted fact among cryptography experts that closed or proprietary cryptographic systems do not provide good security. The reason for this is that creating good cryptography is very difficult and even seasoned experts make mistakes. It is therefore believed that algorithms that have undergone intense public and expert scrutiny are far superior to proprietary ones.

Classic Attacks

Attacks on cryptographic systems can be classified under the following threats:

- Interception
- Modification
- Fabrication
- Interruption

Also, there are both passive and active attacks. Passive attacks involve the listening-in, eavesdropping, or monitoring of information, which may lead to interception of unintended information or traffic analysis where information is inferred. This type of attack is usually difficult if not impossible to detect. However, active attacks involve actual modification of the information flow. This may include[6]:

- Masquerade
- Replay
- Modification of messages
- Denial of service

There are many historical precedents of great value to any security professional considering the use of cryptography. The reader is strongly encouraged to consult many of the excellent books listed in the bibliography, but especially the classic, *The Codebreakers: The Story of Secret Writing*, by David Kahn.[7]

Standard Cryptanalysis

Cryptanalysis strives to break the encryption used to protect information, and to this end there are many techniques available to the modern cryptographer.

Reverse Engineering

Arguably, one of the simplest forms of attack on cryptographic systems is reverse engineering, whereby an encryption device (method, machine, or software) is obtained through other means and then deconstructed to learn how best to extract plaintext. In theory, if a well-designed crypto hardware system is obtained and even its algorithms are learned, it may still be impossible to obtain enough information to freely decrypt any other ciphertext.[8] During World War II, efforts to break the German Enigma encryption device were greatly aided when one of the units was obtained. Also, today when many software encryption packages that claim to be foolproof are analyzed by cryptographers and security professionals, they are frequently found to have serious bugs that undermine the system.

Guessing

Some encryption methods may be trivial for a trained cryptanalyst to decipher. Examples of this include simple substitutions or obfuscation techniques that are masquerading as encryption. A common example of this is the use of the logical XOR function, which when applied to some data will output seemingly random data, but in fact the plaintext is easily obtained. Another example of this is the Caesar cipher, where each letter of the alphabet is shifted by three places so that A becomes D, B becomes E, etc. These are types of cryptograms that commonly present in newspapers and puzzle books.

The *Principle of Easiest Work* states that one cannot expect the interceptor to choose the hard way to do something.[9]

Frequency Analysis

Many languages, especially English, contain words that repeatedly use the same patterns of letters. There have been numerous English letter frequency studies done that give an attacker a good starting point for attacking much ciphertext. For example, by knowing that the letters E, T, and R appear the most frequently in English text, an attacker can fairly quickly decrypt the ciphertext of most monoalphabetic and polyalphabetic substitution ciphers. Of course, critical to this type of attack is the ready supply of sufficient amounts of ciphertext from which to work. These types of frequency and patterns also appear in many other languages, but English appears particularly vulnerable. Monoalphabetic ciphers, such as the Caesar cipher, directly transpose the frequency distribution of the underlying message.

Brute Force

The process of repeatedly trying different keys to obtain the plaintext are referred to as brute-force techniques. Early ciphers were made stronger and stronger in order to prevent human "computers" from decoding secrets; but with the introduction of mechanical and electronic computing devices, many ciphers became no longer usable. Today, as computing power grows daily, it has become a race to improve the resistance, or work factor, to these types of attacks. This of course introduces a problem for applications that may need to protect data that may be of value for many years.

Ciphertext-Only Attack

The cryptanalyst is presented only with the unintelligible ciphertext, from which she tries to extract the plaintext. For example, by examining only the output of a simple substitution cipher, one is able to deduce patterns and ultimately the entire original plaintext message. This type of attack is aided when the attacker has multiple pieces of ciphertext generated from the same key.

Known Plaintext Attack

The cryptanalyst knows all or part of the contents of the ciphertext's original plaintext. For example, the format of an electronic funds transfer might be known except for the amount and account numbers. Therefore, the work factor to extract the desired information from the ciphertext is significantly reduced.

Chosen Plaintext Attack

In this type of attack, the cryptanalyst can generate ciphertext from arbitrary plaintext. This scenario occurs if the encryption algorithm is known. A good cryptographic algorithm will be resistant even to this type of attack.

Birthday Attack

One-way hash functions are used to generate unique output, although it is possible that another message could generate an identical hash. This instance is called a collision. Therefore, an attacker can dramatically reduce the work factor to duplicate the hash by simply searching for these "birthday" pairs.

Factoring Attacks

One of the possible attacks against RSA cryptography is to attempt to use the public key and factor the private key. The security of RSA depends on this being a difficult problem, and therefore takes significant computation. Obviously, the greater the key length used, the more difficult the factoring becomes.

Replay Attack

An attacker may be able to intercept an encrypted "secret" message, such as a financial transaction, but may not be able to readily decrypt the message. If the systems are not providing adequate protection or validation, the attacker can now simply send the message again, and it will be processed again.

Man-in-the-Middle Attack

By interjecting oneself into the path of secure communications or key exchange, it possible to initiate a number of attacks. An example that is often given is the case of an online transaction. A customer connects to what is thought to be an online bookstore; but in fact, the attacker has hijacked the connection to monitor and interact with the data stream. The customer connects normally because the attacker simply forwards the data onto the bookstore, thereby intercepting all the desired data. Also, changes to the data stream can be made to suit the attacker's needs.

In the context of key exchange, this situation is potentially even more serious. If an attacker is able to intercept the key exchange, he may be able to use the key at will (if it is unprotected) or substitute his own key.

Dictionary Attacks

A special type of known-plaintext and brute-force attack can be used to guess the passwords on UNIX systems. UNIX systems generally use the *crypt( )* function to generate theoretically irreversible encrypted password hashes. The problem is that some users choose weak passwords that are based on real words. It is possible to use dictionaries containing thousands of words and to use this well-known function until there is a match with the encoded password. This technique has proved immensely successful in attacking and compromising UNIX systems. Unfortunately, Windows NT systems are not immune from this type of attack. This is accomplished by obtaining a copy of the NT SAM file, which contains the encrypted passwords, and as in the case of UNIX, comparing combinations of dictionary words until a match is found. Again, this is a popular technique for attacking this kind of system.

Attacking Random Number Generators

Many encryption algorithms utilize random data to ensure that an attacker cannot easily recognize patterns to aid in cryptanalysis. Some examples of this include the generation of initialization vectors or SSL sessions. However, if these random number generators are not truly random, they are subject to attack. Furthermore, if the random number generation process or function is known, it may be possible to find weaknesses in its implementation. Many encryption implementations utilize pseudorandom number generators (PRNGs), which as the name the name suggests, attempt to generate numbers that are practically impossible to predict. The basis of these PRNGs is the initial random seed values, which obviously must be selected properly. In 1995, early versions of the Netscape Navigator software were found to have problems with the SSL communication security.[10] The graduate students who reverse engineered the browser software determined that there was a problem with the seeding process used by the random number generator. This problem was corrected in later versions of the browser.

Inference

A simple and potential low-tech attack on encrypted communication can be via simple inference. Although the data being sent back and forth is unreadable to the interceptor, it is possible that the mere fact of this communication may mean there is some significant activity. A common example of this is the communication between military troops, where the sudden increase in traffic, although completely unreadable, may signal the start of an invasion or major campaign. Therefore, these types of communications are often padded so as not to show any increases or decreases in traffic. This example can easily be extended to the business world by considering a pending merger between two companies. The mere fact of increased traffic back and forth may signal the event to an attacker. Also, consider the case of encrypted electronic mail. Although the message data is well encrypted, the sender and recipient are usually plainly visible in the mail headers and message. In fact, the subject line of the message (e.g., "merger proposal") may say it all.

Modern Attacks

Although classical attacks still apply and are highly effective against modern ciphers, there have been a number of recent cases of new and old cryptosystems failing.

Bypass

Perhaps one of the simplest attacks that has emerged, and arguably is not new, is to simply go around any crypto controls. This may be as simple as coercion of someone with access to the unencrypted data or by exploiting a flaw in the way the cipher is used. There are currently a number of PC encryption products on the market and the majority of these have been found to have bugs. The real difference in these products has been the ways in which the vendor has fixed the problem (or not). A number of these products have been found to improperly save passwords for convenience or have backdoor recovery mechanisms installed. These bugs were mostly exposed by curious users exploring how the programs work. Vendor responses have ranged from immediately issuing fixes to denying there is a problem.

Another common example is the case of a user who is using some type of encryption software that may be protecting valuable information or communication. An attacker could trick the user into running a Trojan horse program, which secretly installs a backdoor program, such as BackOrifice on PCs. On a UNIX system, this attack may occur via an altered installation script run by the administrator. The administrator can now capture any information used on this system, including the crypto keys and passphrases. There have been several demonstrations of these types of attacks where the target was home finance software or PGP keyrings. The author believes that this form of attack will greatly increase as many more users begin regularly using e-mail encryption and Internet banking.

Operating System Flaws

The operating system running the crypto function can itself be the cause of problems. Most operating systems use some form of virtual memory to improve performance. This "memory" is usually stored on the system's hard disk in files that may be accessible. Encryption software may cache keys and plaintext while running, and

this data may remain in the system's virtual memory. An attacker could remotely or physically obtain access to these files and therefore may have access to crypto keys and possibly even plaintext.

Memory Residue

Even if the crypto functions are not cached in virtual memory or on disk, many products still keep sensitive keys in the system memory. An attacker may be able to dump the system memory or force the system to crash, leaving data from memory exposed. Hard disks and other media may also have residual data that may reside on the system long after use.

Temporary Files

Many encryption software packages generate temporary files during processing and may accidentally leave plaintext on the system. Also, application packages such as word processors leave many temporary files on the system, which may mean that even if the sensitive file is encrypted and there are no plaintext versions of the file, the application may have created plaintext temporary files. Even if temporary files have been removed, they usually can be easily recovered from the system disks.

Differential Power Analysis

In 1997, Anderson and Kuhn proposed inexpensive attacks against through which knowledgeable insiders and funded organizations could compromise the security of supposed tamper-resistant devices such as smart cards.[11] While technically not a crypto attack, these types of devices are routinely used to store and process cryptographic keys and provide other forms of assurance. Further work in this field has been done by Paul Kocher and Cryptographic Research, Inc. Basically, the problem is that statistical data may "leak" through the electrical activity of the device, which could compromise secret keys or PINs protected by it. The cost of mounting such an attack appears to be relatively low but it does require a high technical skill level. This excellent research teaches security professionals that new forms of high-security storage devices are highly effective but have to be used appropriately and that they do not provide *absolute* protection.

Parallel Computing

Modern personal computers, workstations, and servers are very powerful and are formidable cracking devices. For example, in *Internet Cryptography*,[12] Smith writes that a single workstation will break a 40-bit export crypto key, as those used by Web browsers, in about ten months. However, when 50 workstations are applied to this problem processing in parallel, the work factor is reduced to about six days. This type of attack was demonstrated in 1995 when students using a number of idle workstations managed to obtain the plaintext of an encrypted Web transaction.

Another example of this type of processing is *Crack* software, which can be used to brute-force guess UNIX passwords. The software can be enabled on multiple systems that will work cooperatively to guess the passwords.

Parallel computing has also become very popular in the scientific community due the fact that one can build a supercomputer using off-the-shelf hardware and software. For example, Sandia National Labs has constructed a massively parallel system called Cplant, which was ranked the 44th fastest among the world's 500 fastest supercomputers (http://www.wired.com/ news/technology/0,1282,32706,00.html). Parallel computing techniques mean that even a moderately funded attacker, with sufficient time, can launch very effective and low-tech brute-force attacks against medium to high value ciphertext.

Distributed Computing

For a number of years, RSA Security has proposed a series of increasingly difficult computation problems. Most of the problems require the extraction of RSA encrypted messages and there is usually a small monetary award. Various developers of elliptic curve cryptography (ECC) have also organized such contests. The primary reason for holding these competitions is to test current minimum key lengths and obtain a sense of the "real-world" work factor.

Perhaps the most aggressive efforts have come from the Distributed.Net group, which has taken up many such challenges. The Distributed team consists of thousands of PCs, midrange, and high-end systems that

collaboratively work on these computation problems. Other Internet groups have also formed and have spawned distributed computing rivalries. These coordinated efforts show that even inexpensive computing equipment can be used in a distributed or collaborative manner to decipher ciphertext.

DES Cracker

In 1977, Whitfield Diffie and Martin Hellman proposed the construction of a DES-cracking machine that could crack 56-bit DES keys in 20 hours. Although the cost of such a device is high, it seemed well within the budgets of determined attackers. Then in 1994, Michael Weiner proposed a design for a device built from existing technology which could crack 56-bit DES keys in under four hours for a cost of $1 million. The cost of this theoretical device would of course be much less today if one considers the advances in the computer industry.

At the RSA Conferences held in 1997 and 1998, there were contests held to crack DES-encrypted messages. Both contests were won by distributed computing efforts. In 1998, the DES message was cracked in 39 days. Adding to these efforts was increased pressure from a variety of groups in the United States to lift restrictive crypto export regulations. The Electronic Freedom Foundation (EFF) sponsored a project to build a DES cracker. The intention of the project was to determine how cheap or how expensive it would be to build a DES cracker.

In the summer of 1998, the EFF DES cracker was completed, costing $210,000 and taking only 18 months to design, test, and build. The performance of the cracker was estimated at about five days per key. In July 1998, EFF announced to the world that it had easily won the RSA Security "DES Challenge II," taking less than three days to recover the secret message. In January 1999, EFF announced that in a collaboration with Distributed.Net, it had won the RSA Security "DES Challenge III," taking 22 hours to recover the plaintext. EFF announced that this "put the final nail into the Data Encryption Standard's coffin." EFF published detailed chip design, software, and implementation details and provided this information freely on the Internet.

RSA-155 (512bit) Factorization

In August 1999, researchers completed the factorization of the 155-digit (512-bit) RSA Challenge Number. The total time taken to complete the solution was around five to seven months without dedicating hardware. By comparison, RSA-140 was solved in nine weeks. The implications of this achievement in relatively short time may put RSA keys at risk from a determined adversary. In general, it means that 768- or 1024-bit RSA keys should be used as a minimum.

TWINKLE RSA Cracker

In summer 1999, Adi Shamir, co-inventor of the RSA algorithm, presented a design for The Weizmann Institute Key Locating Engine (TWINKLE), which processes the "sieving" required for factoring large numbers. The device would cost about $5000 and provide processing equivalent to 100 to 1000 PCs. If built, this device could be used similarly to the EFF DES Cracker device. This device is targeted at 512-bit RSA keys, so it reinforces the benefits of using of 768- or 1024-bit, or greater keys.

Key Recovery and Escrow

Organizations implementing cryptographic systems usually require some way to recover data encrypted with keys that have been lost. A common example of this type of system is a public key infrastructure, where each private (and public) key is stored on the Certificate Authority, which is protected by a root key(s). Obviously, access to such a system has to be tightly controlled and monitored to prevent a compromise of all the organization's keys. Usually, only the private data encrypting, but not signing, keys are "escrowed."

In many nations, governments are concerned about the use of cryptography for illegal purposes. Traditional surveillance becomes difficult when the targets are using encryption to protect communications. To this end, some nations have attempted to pursue strict crypto regulation, including requirements for key escrow for law enforcement.

In general, key recovery and escrow implementations could cause problems because they are there to allow access to all encrypted data. Although a more thorough discussion of this topic is beyond the scope of this chapter, the reader is encouraged to consult the report entitled "The Risks of Key Recovery, Key Escrow, and

Trusted Third Party Encryption," which was published in 1997 by an *ad hoc* group of cryptographers and computer scientists. Also, Whitfield Diffie and Susan Landau's *Privacy on the Line* is essential reading on the topic.

Protecting Cryptosystems

Creating effective cryptographic systems requires balancing business protection needs with technical constraints. It is critical that these technologies be included as part of an effective and holistic protection solution. It is not enough to simply implement encryption and assume all risks have been addressed. For example, just because an e-mail system is using message encryption, it does not necessarily mean that e-mail is secure, or even any better than plaintext. When considering a protection system, not only must one look at and test the underlying processes, but one must also look for ways around the solutions and address these risks appropriately. It is vital to understand that crypto solutions can be dangerous because they can easily lead to a false sense of information security.

Design, Analysis, and Testing

Fundamental to the successful implementation of a cryptosystem are thorough design, analysis, and testing methodologies. The implementation cryptography is probably one of the most difficult and most poorly understood IT fields. Information technology and security professionals must fully understand that cryptographic solutions that are simply dropped into place are doomed to failure.

It is generally recommended that proprietary cryptographic systems are problematic and usually end up being not quite what they appear to be. The best algorithms are those that have undergone rigorous public scrutiny by crypto experts. Just because a cryptographer cannot break his or her own algorithm, this does not mean that this is a safe algorithm. As Bruce Schneier points out in "Security Pitfalls in Cryptography," the output from a poor cryptographic system is very difficult to differentiate from a good one.

Smith[13] suggests that preferred crypto algorithms should have the following properties:

- No reliance on algorithm secrecy
- Explicitly designed for encryption
- Available for analysis
- Subject to analysis
- No practical weaknesses

When designing systems that use cryptography, it is also important to build in proper redundancies and compensating controls, because it is entirely possible that the algorithms or implementation may fail at some point in the future or at the hands of a determined attacker.

Selecting Appropriate Key Lengths

Although proper design, algorithm selection, and implementation are critical factors for a cryptosystem, the selection of key lengths is also very important. Security professionals and their IT peers often associate the number of "bits" a product uses with the measure of its level of protection. As Bruce Schneier so precisely puts it in his paper "Security Pitfalls in Cryptography": "…reality isn't that simple. Longer keys don't always mean more security."[14] As stated earlier, the cryptographic functions are but part of the security strategy. Once all the components and vulnerabilities of a encryption strategy have been reviewed and addressed, one can start to consider key lengths.

In theory, the greater the key length, the more difficult the encryption is to break. However, in practice, there are performance and practical concerns that limit the key lengths to be used. In general, the following factors will determine what key sizes are used:

- Value of the asset it is protecting (compare to cost to break it)
- Length of time it needs protecting (minutes, hours, years, centuries)
- Determination of attacker (individual, corporate, government)
- Performance criteria (seconds versus minutes to encrypt/decrypt)

Therefore, high value data that needs to protected for a long time, such as trade secrets, requires long key lengths. Whereas, a stock transaction may only be of value for a few seconds, and therefore is well protected with shorter key lengths. Obviously, it is usually better to err toward longer key sizes than shorter. It is fairly common to see recommendations of symmetric key lengths, such as for 3DES or IDEA, of 112 to 128 bits, while 1024- to 2048-bit lengths are common for asymmetric keys, such as for RSA encryption.

Random Number Generators

As discussed earlier, random number generators are critical to effective cryptosystems. Hardware-based RNG are generally believed to be the best, but more costly form of implementation. These devices are generally based on random physical events, and therefore should generate data that is nearly impossible to predict.

Software RNGs obviously require additional operating system protection, but also protection from covert channel analysis. For example, systems that use system clocks may allow an attacker access to this information via other means, such as remote system statistics or network time protocols. Bruce Schneier has identified software random number generators as being a common vulnerability among crypto implementations [SOURCE], and to that end has made an excellent free PRNG available, with source code, to anyone. This PRNG has undergone rigorous independent review.

Source Code Review

Even if standard and publicly scrutinized algorithms and methods are used in an application, this does not guarantee that the application will work as expected. Even open-source algorithms are difficult to implement correctly because there are many nuances (e.g., cipher modes in DES and proper random number generation) that the programmer may not understand. Also, as discussed in previous sections, many commercial encryption packages have sloppy coding errors such as leaving plaintext temporary files unprotected. Cryptographic application source code should be independently reviewed to ensure that it actually does what is expected.

Vendor Assurances

Vendor assurances are easy to find. Many products claim that their data or communications are encrypted or are secure; however, unless they provide any specific details, it usually turns out that this protection is not really there or is really just "obfuscation" at work. There are some industry evaluations and standards that may assist in selecting a product. Some examples are the Federal Information Processing Standards (FIPS), the Common Criteria evaluations, ICSA, and some information security publications.

New Algorithms

Advanced Encryption Algorithm (AES)

A new robust encryption algorithm was needed to replace the aging Data Encryption Standard (FIPS 46-3), which had been developed in the 1970s. In September 1997, NIST issued a Federal Register notice soliciting an unclassified, publicly disclosed encryption algorithm that would be available royalty-free, worldwide. Following the submission of 15 candidate algorithms and three publicly held conferences to discuss and analyze the candidates, the field was narrowed to five candidates:

- MARS (IBM)
- RC6TM (RSA Laboratories)
- RIJNDAEL (Joan Daemen, Vincent Rijmen)
- Serpent (Ross Anderson, Eli Biham, Lars Knudsen)
- Twofish (Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson)

NIST continued to study all available information and analyses about the candidate algorithms, and selected one of the algorithms, the Rijndael algorithm, to propose for the AES. The Secretary of Commerce approved FIPS 197, Advanced Encryption Standard (AES), which, effective May 26, 2002, makes it compulsory and binding on federal agencies for the protection of sensitive, unclassified information. The development and public review process has proven very interesting, showing the power of public review of cryptographic algorithms.

Conclusion

The appropriate use of cryptography is critical to modern information security, but it has been shown that even the best defenses can fail. It is critical to understand that cryptography, while providing excellent protection, can also lead to serious problems if the whole system is not considered. Ultimately, practitioners must understand not only the details of the crypto products they are using, but what they are in fact protecting, why these controls are necessary, and who they are protecting these assets against.

Notes

1. Schneier, Bruce, *Applied Cryptography*, New York: John Wiley, 1995, p. 19.
2. Stallings, William, *Cryptography and Network Security*: *Principles and Practice,* Englewood Cliffs: Prentice-Hall, 2002, p. 19.
3. Spafford, *Practical UNIX and Internet Security, Sebastapol*: O'Reilly & Associates, 2003, p. 19.
4. Schneier, Bruce, *Applied Cryptography,* New York: John Wiley, 1995, p. 11.
5. Stallings, William, *Cryptography and Network Security: Principles and Practices,* Englewood Cliffs: Prentice-Hall, 2002, p. 26.
6. Stallings, William, *Cryptography and Network Security: Principles and Practice,* Englewood Cliffs: Prentice-Hall, 2002, pp. 7–9.
7. Kahn, David, *The Codebreakers: The Story of Secret Writing,* New York: Scribner, 1983, p. 19.
8. Smith, Richard, E., *Internet Cryptography,* Reading, MA: Addison-Wesley, 1997, p. 95.
9. Pfleeger, E., Charles, *Security in Computing,* Englewood Cliffs: Prentice-Hall, 1996, p. 19.
10. Smith, Richard, E., *Internet Cryptography,* Reading, MA: Addison-Wesley, 1997, p. 91.
11. Anderson, Ross, Kuhn, and Markus, Low Cost Attacks on Tamper Resistant Devices, *Security Protocols, 5th Int. Workshop,* 1997.
12. Smith, Richard E., *Internet Cryptography,* p. 19.
13. Smith, Richard E., *Internet Cryptography,* , p. 52.
14. Schneier, Bruce, *Security Pitfalls in Cryptography,* http://www.counterpane.com/pitfalls. html.

Domain 6 Enterprise Security Architecture

The Enterprise Security Architecture Domain contains the concepts, principles, structures, and standards used to design, implement, monitor, and secure operating systems, equipment, networks, applications, and those controls used to enforce various levels of confidentiality, integrity, and availability.

Building an information system requires a balance among various requirements, such as capability, flexibility, performance, ease of use, cost, business requirements, and security. Security should be considered a requirement from the beginning — it is simply another feature that needs to be included. Attempting to retrofit the required and desired security controls after the fact can lead to user frustration, a lowered security posture, and significantly increased implementation costs. Based on the importance of each requirement, various trade-offs may be necessary during the design of the system. Thus, it is important to identify what security features must be included. Then if a performance or flexibility requirement means downgrading or not including a security feature, the architecture designers can keep the primary goals of the system in check and make compromises on the nonessential points.

Security architecture is simply a view of an overall system architecture from a security perspective. It provides some insight into the security services, mechanisms, technologies, and features that can be used to satisfy system security requirements. It provides recommendations on where, within the context of the overall system architecture, security mechanisms should be placed. The security view of a system architecture focuses on the system security services and high-level mechanisms, allocation of security-related functionality, and identified interdependencies among security related components, services, mechanisms, and technologies, and at the same time reconciling any conflict among them. The security architecture is only one aspect of the enterprise or system architecture, which may also include network architecture or physical connectivity architecture.

Security architecture describes how the system is put together to satisfy the security requirements. It is not a description of the functions of the system; it is more of a design overview, describing at an abstract level the relationships between key elements of the hardware, operating systems, applications, network, and other required components to protect the organization's interests. It should also describe how the functions in the system development process follow the security requirements. For example, if the security requirements specify that the system must have a given level of assurance as to the correctness of the security controls, the security architecture must prescribe these specifications in the development process.

Security requirements are not added steps to the development process; instead, the specifications or guidelines of the security architecture provide an influence during all development processes. During the beginning stages, the security architecture should outline high-level security issues, such as the system security policy, the level of assurance required, and any potential impacts security could have on the design process. As the system is developed, the security architecture should evolve in parallel, and may even need to be slightly ahead of the development process so that the security requirements will guide the development process.

The chapters presented here provide the necessary breadth to address the challenges of developing a security architecture and the insight to evaluate the existing or legacy architecture of an organization.

Contents

Enterprise Security Architecture

William Hugh Murray

INTRODUCTION

Sometime during the 1980s we crossed a line from a world in which the majority of computer users were users of multi-user systems to one in which the majority were users of single-user systems. We are now in the process of connecting all computers in the world into the most complex mechanism that humans have ever built. While for many purposes we may be able to do this on an ad hoc basis, for purposes of security, audit, and control it is essential that we have a rigorous and timely design. We will not achieve effective, much less efficient, security without an enterprise-wide design and a coherent management system.

Enterprise

If you look in the dictionary for the definitions of enterprise, you will find that an enterprise is a project, a task, or an undertaking; or, the readiness for such, the motivation, or the moving forward of that undertaking. The dictionary does not contain the definition of the enterprise as we are using it here. For our purposes here, the enterprise is defined as the largest unit of business organization, that unit of business organization that is associated with ownership. If the institution is a government institution, then it is the smallest unit headed by an elected official. What we need to understand is that it is a large, coordinated, and independent organization.

ENTERPRISE SECURITY IN THE 1990s

Because the scale of the computer has changed from one scaled to the enterprise to one scaled to the application or the individual, the computer security requirements of the enterprise have changed. The new requirement can best be met by an architecture or a design.

We do not do design merely for the fun of it or even because it is the "right" thing to do. Rather, we do it in response to a problem or a set of requirements. While the requirements for a particular design will be those

for a specific enterprise, there are some requirements that are so pervasive as to be typical of many, if not most, enterprises. This section describes a set of observations by the author to which current designs should respond.

Inadequate expression of management intent — One of these is that there is an inadequate expression of management's intent. Many enterprises have no written policy at all. Of those that do, many offer inadequate guidance for the decisions that must be made. Many say little more than "do good things." They fail to tell managers and staff how much risk general management is prepared or intends to accept. Many fail to adequately assign responsibility or duties or fix the discretion to say who can use what resources. This results in inconsistent risk and inefficient security, i.e., some resources are overprotected and others are underprotected.

Multiple sign-ons, IDs, and passwords — Users are spending tens of minutes per day logging on and logging off. They may have to log on to several processes in tandem in order to access an application. They may have to log off of one application in order to do another. They may be required to remember multiple user identifiers and coordinate many passwords. Users are often forced into insecure or inefficient behavior in futile attempts to compensate for these security measures. For example, they may write down or otherwise record identifiers and passwords. They may even automate their use in macros. They may postpone, or even forget tasks so as not to have to quit one application in order to do another. This situation is often not obvious to system managers. They tend to view the user only in the context of the systems that they manage rather in the context of the systems he uses. He may also see this cost as "soft money," not easily reclaimed by him. On the other hand, it is very real money to the enterprise which may have thousands of such users and which might be able to get by with fewer if they were not engaged in such activity. Said another way, information technology management overlooks what general management sees as an opportunity.

Multiple points of control — Contrary to what we had hoped and worked for in the 1980s, data are proliferating and spreading throughout the enterprise. We did not succeed in bringing all enterprise data under a single access control system. Management is forced to rely upon multiple processes to control access to data. This often results in inconsistent and incomplete control. Inconsistent control is usually inefficient. It means that management is spending too much or too little for protection. Incomplete control is ineffective. It means that some data are completely unprotected and unreliable.

Unsafe defaults — In order to provide for ease of installation and avoid deadlocks, systems are frequently shipped with security mechanisms set to the unsafe conditions by default. The designers are concerned that even

before the system is completely installed, management may lose control. The administrator might accidentally lock himself out of his own system with no remedy but to start over from scratch. Therefore, the system may be shipped with controls defaulted to their most open settings. The intent is that after the systems are configured and otherwise stable, the administrator will reset the controls to the safe condition. However, in practice and so as not to interfere with running systems, administrators are often reluctant to alter these settings. This may be complicated by the fact that systems which are not securely configured are, by definition, unstable. The manager has learned that changes to an already unstable system tend to aggravate the instability.

Complex administration — The number of controls, relations between them, and the amount of special knowledge required to use them may overwhelm the training of the administrator. For example, in order to properly configure the password controls for a Novell server, the administrator may have to set four different controls. The setting of one requires not only knowledge of how the others are set but how they relate to each other. The administrator's training is often focused on the functionality of the systems rather than on security and control. The documentation tends to focus on the function of the controls while remaining silent on their use to achieve a particular objective or their relationship to other controls.

Late recognition of problems — In part because of the absence of systematic measurement and monitoring systems, many problems are being detected and corrected late. Errors that are not detected or corrected may be repeated. Attacks are permitted to go on long enough to succeed. If permitted to continue for a sufficient length of time without corrective action, any attack will succeed. The cost of these problems is greater than it would be if they were detected on a more timely basis.

Increasing use, users, uses, and importance — Most important for our purposes here, security requirements arise in the enterprise as the result of increasing use of computers, increasing numbers of users, increasing numbers of uses and applications, and increasing importance of those applications and uses to the enterprise. All of these things can be seen to be growing at a rate that dwarfs our poor efforts to improve security. The result is that relative security is diminishing to the point that we are approaching chaos.

ARCHITECTURE DEFINED

In response to these things we must increase not only the effectiveness of our efforts but also their efficiency. Because we are working on the scale of the enterprise, ad hoc and individual efforts are not likely to be successful. Success will require that we coordinate the collective efforts of the enterprise according to a plan, design, or architecture.

Architecture can be defined as that part of design that deals with what things look like, what they do, where they are, and what they are made of. That is, it deals with appearance, function, location, and materials. It is used to agree on what is to be done and what results are to be produced so that multiple people can work on the project in a collaborative and cooperative manner and so that we can agree when we are through and the results are as expected.

The design is usually reflected in a picture, model, or prototype; in a list of specified materials; and possibly in procedures to be followed in achieving the intended result. When dealing in common materials, the design usually references standard specifications. When using novel materials the design must describe these materials in detail.

In information technology we borrow the term from the building and construction industry. However, unlike this industry, we do not have 10,000 years of tradition, conventions, and standards behind us. Neither do we share the rigor and discipline that characterize them.

TRADITIONAL IT ENVIRONMENT

Computing environments can be characterized as traditional and modern. Each has its own security requirements but, in general and all other things being equal, the traditional environment is easier to secure than its modern equivalent.

Closed — Traditional IT systems and networks are closed. Only named parties can send messages. The nodes and links are known in advance. The insertion of new ones requires the anticipation and cooperation of others. They are closed in the sense that their uses or applications are determined in advance by their design, and late changes are resisted.

Hierarchical — Traditional IT can be described as hierarchical. Systems are organized and controlled top down, usually in a hierarchical or tree structure. Messages and controls flow vertically better than they do horizontally. Such horizontal traffic as exists is mediated by the node at the top of the tree, for example, a mainframe.

Point-to-point — Traffic tends to flow directly from point to point along nodes and links which, at least temporarily, are dedicated to the traffic. Traffic flows directly from one point to another; what goes in at node A will come out only at node B.

Connection switched — The resources that make up the connection between two nodes are dedicated to that connection for the life of the communication. When either is to talk to another, the connection is torn down and a new one is created. The advantage is in speed of communication and security, but capacity may not be used efficiently.

Host-dependent workstations — In traditional computing, workstations are incapable of performing independent applications. They are dependent upon cooperation with a host or master in order to be able to perform any useful work.

Homogeneous components — In traditional networks and architectures, there is a limited number of different component types from a limited number of vendors. Components are designed to work together in a limited number of ways. That is to say part of the design may be dictated by the components chosen.

MODERN IT ENVIRONMENT

Open — By contrast, modern computing environments are open. Like the postal system, for the price of a stamp anyone may send a message. For the price of an accommodation address, anyone can get an answer back. For not much more, anyone can open his own post office. Modern networks are open in the sense that nodes can be added late and without the permission or cooperation of others. They are open in the sense that their applications are not predetermined.

Flat — The modern network is flat. Traffic flows with equal ease between any two points in the network. It flows horizontally as well as it does vertically. Traffic flows directly and without any mediation. If one were to measure the bandwidth between any two points in the network, chosen arbitrarily, it would be approximately equal to that between any other two points chosen the same way. While traffic my flow faster between two points that are close to each other, taken across the collection of all pairs, it flows with the same speed.

Broadcast — Modern networks are broadcast. While orderly nodes accept only that traffic which is intended for them, traffic will be seen by multiple nodes in addition to the one for which it is intended. Thus, confidentiality may depend in part upon the fact that a large number of otherwise unreliable devices all behave in an orderly manner.

Packet-switched — Modern networks are packet-switched rather than circuit-switched. In part this means that the messages are broken into packets and each packet is sent independent of the others. Two packets sent from the same origin to the same destination may not follow the same path and may not arrive at the destination in the same order that they were sent. The sender cannot rely upon the safety of the path or the arrival of the message at the destination and the receiver cannot rely upon the return address. In part, it means that a packet may be broadcast to multiple nodes, even to all nodes, in an attempt to speed it to its destination. By design it will be heard by many nodes other than the ones for which it is intended.

Intelligent workstations — In modern environments, the workstations are intelligent, independently programmable, and capable of performing independent work or applications. They are also vulnerable both to the leakage of sensitive information and to the insertion of malicious programs. These malicious programs may be untargeted viruses or they may be password grabbers that are aimed at specific workstations, perhaps those used by privileged users.

Heterogeneousness — The modern network is composed of a variety of nodes and links from many different vendors. There may be dozens of different workstations, servers, and operating systems. The links may be of many speeds and employ many different kinds of signaling. This makes it difficult to employ an architecture that relies upon the control or behavior of the components.

OTHER SECURITY ARCHITECTURE REQUIREMENTS

IT architecture — The information security architecture is derivative of and subordinate to the information technology architecture. It is not independent. One cannot do a security architecture except in the context of and in response to an IT architecture. An information technology architecture describes the appearance, function, location, and materials for the use of information technology. Often one finds that the IT architecture is not sufficiently well thought out or documented to support the development of the security architecture. That is to say, it describes fewer than all four of the things that an architecture must describe. Where it is documented at all, one can expect to find that it describes the materials but not appearance, location, or function.

Policy or management intent — The security architecture must document and respond to a policy or an expression of the level of risk that management is prepared to take. This will influence materials chosen, the roles assigned, the number of people involved in sensitive duties, etc.

Industry and institutional culture — The architecture must document and respond to the industry and institutional culture. The design that is appropriate to a bank will not work for a hospital, university, or auto plant.

Other — Likewise, it must respond to the management style — authoritarian or permissive, prescriptive or reactive — of the institution, to law and regulation, to duties owed to constituents, and to good practice.

SECURITY ARCHITECTURE

The security architecture describes the appearance of the security functions, what is to be done with them, where they will be located within the

organization, its systems, and its networks, and what materials will be used to craft them. Among other things, it will describe the following.

Duties, roles, and responsibilities — It will describe who is to do what. It specifies who management relies upon and for what. For every choice or degree of freedom within the system, the architecture will identify who will exercise it.

How objects will be named — It will describe how objects are named. Specifically, it will describe how users are named, identified or referred to. Likewise it will describe how information resources are to be named within the enterprise.

What authentication will look like — It must describe how management gains sufficient confidence in these names or identifiers. How does it know that a user is who he says he is and that the data returned for a name are the expected data? Specifically, the architecture describes what evidence the user will present to demonstrate her identity. For example, if the user is to be authenticated based upon something that he knows, what are the properties (length and character set) of that knowledge?

Where it will be done — Similarly, the architecture will describe where the instant data are to be collected, where the reference data will be stored, and what process will reconcile the two.

What the object of control will be — The architecture must describe what it is that will be controlled. In the traditional IT architecture this was usually a file or a dataset, or sometimes a procedure such as a program or a transaction type. In modern systems it is more likely to be a data base object such as a table or a view.

Where access will be controlled — The architecture will describe where, i.e., what processes, will exercise control over the objects. In the traditional IT architecture we tried to centralize all access control in a single process, scaled to the enterprise. In more modern systems access will be controlled in a large number of places. These places will be scaled to departments, applications, and other ways of organizing resources. They may be exclusive or they may overlap. How they are related and where they are located is the subject of the design.

Generation and distribution of warnings and alarms — Finally, the design must specify what events or combinations of events require corrective action, what process will detect them, who is responsible for the action, and how the warning will be communicated from the detecting process to the party responsible for the correction.

POLICY

A Statement of Management's Intent

Among other things, a policy is a statement of management's intent. Among other things, a security policy describes how much risk management intends to take. This statement must be adequate for managers to be able to figure out what to do in a given set of circumstances. It should be sufficiently complete that two managers will read it the same way, reach similar conclusions, and behave in similar ways.

It should speak to how much risk management is prepared to take. For example, management expects to take normal business risk, or acceptable and accepted risk. Alternately or in addition, management can specify the intended level of control. For example, management can say that controls must be such that multiple people must be involved in sensitive duties or material fraud.

The policy should state what management intends to achieve, for example, data integrity, availability, and confidentiality, and how it intends to do it. It should clearly state who is to be responsible for what. It should state who is to have access to what information. Where such access is to be restricted or discretionary, then the policy should state who will exercise the discretion.

The policy should be such that it can be translated into an access control policy. For example, it might say that read access to confidential data must be restricted to those authorized by the owner of the data. The architecture will describe how a given platform or a network of platforms will be used to implement that policy.

IMPORTANT SECURITY SERVICES

The architecture will describe the security mechanisms and services that will be used to implement the access control policy. These will include but not be limited to the following.

User name service — The user name service is used for assigning unique names to users and to resolve aliases where necessary. It can be thought of as a data base, data base application, or data base service. The server can encode and decode user names into user identifiers. For the distinguished user name it returns a system user identifier or identifiers. For the system user identifier it returns a distinguished user name. It can be used to store information about the user. It is often used to store other descriptive data about the user. It may store office location, telephone number, department name, and manager's name.

Group name service — The group name service is used for assigning unique group names and for associating users with those groups. It

permits the naming of any arbitrary but useful group such as member of department m, employees, vendors, consultants, users of system 1, users of application A, etc. It can also be used to name groups of one, such as the payroll manager. For the group name, it returns the names, identifiers, or aliases of members of the group. For a user name, it returns a list of the groups of which that user is a member. A complete list of the groups of which a user is a member is a description of his role or relationship to the enterprise. Administrative activity can be minimized by assigning authority, capabilities, and privileges to groups and assigning users to the groups. While this is indirect it is also usually efficient.

Authentication server — The authentication server reconciles evidence of identity. Users are enrolled along with the expectation, i.e., the reference data, for authenticating their identity. For a user identifier and an instance of authenticating data, the server returns *true* if the data meets its expectation, i.e., matches the reference data, and *false* if it does not. If *true*, the server will vouch to its clients for the identity of the user. The authentication server must be trusted by its client and the architecture must provide the basis for that trust. The server may be attached to its client by a trusted path or it may give its client a counterfeit-resistant voucher (ticket or encryption-based logical token).

Authentication service products — A number or authentication services are available off the shelf. These include Kerberos, SESAME, NetSP, and Open Software Foundation Distributed Computing Environment (OSF/DCE). These products can meet some architectural requirements in whole or in part.

Single point of administration — One implication of multiple points of control is that there may be multiple controls that must be administered. The more such controls there are, the more desirable it becomes to minimize the points of administration. Such points of administration may simply provide for a common interface to the controls or may provide for a single data base of its own. There are a number of standard architectures that are useful here. These include SESAME and the Open Software Foundation Distributed Computing Environment.

RECOMMENDED ENTERPRISE SECURITY ARCHITECTURE

This section makes some recommendations about enterprise security architecture. It describes those choices which, all other things equal, are to be preferred over others.

Single-user name space for the enterprise — Prefer a single-user name space across all systems. Alternatively, have an enterprise name server that relates all of a user's aliases to his distinguished name. This server

should be the single point of name assignment. In other words it is a data base application or server for assigning names.

Prefer strong authentication — Strong authentication should be preferred by all enterprises of interest. Strong authentication is characterized by two kinds of evidence, at least one of which is resistant to replay. Users should be authenticated using two kinds of evidence. Evidence can be something that only one person knows, has, is, or can do. The most common form of strong authentication is something that the user knows such as a password, pass-phrase, or personal identification number (PIN), plus something that they carry such as a token. The token generates a one-time password that is a function of time or a challenge. Other forms in use include a token plus palm geometry or a PIN plus the way the user speaks.

Prefer single sign-on — Prefer single sign-on. A user should have to log on only once per workstation per enterprise per day. A user should not be surprised that if he changes workstations, crosses an enterprise boundary, or leaves for the day, that he should have to log on again. However, he should not have to log off one application to log on to another or log on to multiple processes to use one application.

Application or service as point of control — Prefer the application or service as the point of control. The first applicable principle is that the closer to the data that the control is, the fewer instances of it that there will be, the less subject it will be to user interference, the more difficult it will be to bypass, and consequently, the more reliable it will be. This principle can be easily understood by contrasting it to the worst case — the one where the control is on the desktop. Multiple copies must be controlled, they are very vulnerable to user interference, not to say complete abrogation, and the more people there are who are already behind the control. The second principle is that application objects are both specific, i.e., their behavior is intuitive, predictable from their name, and obvious as to their intended use. Contrast "update name and address of customer" to "write to customer data base." One implication of the application as the point of control is that there will be more than one point of control. However, there will be fewer than if the control were even closer to the user.

Multiple points of control — Each server or service should be responsible for control of access to all of its dynamically allocated resources. Prefer that all such resources be of the same resource type. To make its access decision, the server may use local knowledge or data or it may use a common service that is sufficiently abstract to include its rules. One implication of the server or service as the point of control is that there will be multiple points of control. That is to say, there are multiple repositories of data and multiple mechanisms that management must manipulate to exercise control. This may increase the requirement for special knowledge, communication, and coordination.

Limited points of administration — Therefore, prefer a limited number of points of administration that operate across a number of points of control. These may be relatively centralized to respond to a requirement for a great deal of special knowledge about the control mechanism. Alternatively it can be relatively decentralized to meet a requirement for special knowledge about the users, their duties, and responsibilities.

Single resource name space for enterprise data — Prefer a single name space for all enterprise data. Limit this naming scheme to enterprise data; i.e., data that are used and meaningful across business functions or that are related to the business strategy. It is not necessary to include all business functional data, project data, departmental data, or personal data.

Object, table, or view as unit of control — Prefer capabilities, objects, tables, views, rows, columns, and files, in that order as objects of control. This is the order in which the data are most obvious as to meaning and intended use.

Arbitrary group names with group-name service — It is useful to be able to organize people into affinity groups. These may include functions, departments, projects, and other units of organization. They may also include such arbitrary groups as employees, nonemployees, vendors, consultants, contractors, etc. The architecture should deal only with enterprise-wide groups. It should permit the creation of groups which are strictly local to a single organizational unit or system. Enterprise group names should be assigned and group affinities should be managed by a single service across the enterprise and across all applications and systems. This service may run as part of the user name service. Within reasonable bounds any user should be able to define a group for which he is prepared to assume ownership and responsibility. Group owners should be able to manage group membership or delegate it. For example, the human resources manager might wish to restrict the ability to add members to the group *payroll department* while permitting any manager to add users to the group *employee* or the group *nonemployee*.

Rules-based (as opposed to list-based) access control — Prefer rules-based to list-based access control. For example, prefer "access to data labelled confidential is limited to employees" should be preferred to "user A can access dataset 1." While the latter is more granular and specific, the former covers more data in a single rule. The latter will require much more administrative activity to accomplish the same result as the former. Similarly, it can be expressed in far less data. While the latter may permit only a few good things to happen, the former forbids a large number of bad things. This recommendation is counterintuitive to those of us who are part of the tradition of "least possible privilege." This rule implies that a user should be given access to only those resources required to do their job and that all access should be explicit. The rule of least privilege worked

well in a world in which the number of users, data objects, and relations between them was small. It begins to break down rapidly in the modern world of tens of millions of users and billions of resources.

Data-based rules — Access control rules should be expressed in terms of the name and other labels of the data rather than in terms of the procedure to be performed. They should be independent of the procedures used to access the data or the environment in which they are stored. That is, it is better to say that a user has *read* access to *filename* than to say that he has *execute* access to *word.exe*. It makes little sense to say that a user is restricted to a procedure that can perform arbitrary operations on an unbounded set of objects. This is an accommodation to the increase in the number of data objects and the decreasing granularity of the procedures.

Prefer single authentication service — Evidence of user identity should be authenticated by a single central process for the entire enterprise and across all systems and applications. These systems and applications can be clients of the authentication server or the server can issue trusted credentials to the user that can be recognized and honored by the using systems and applications.

Prefer a single standard interface for invoking security services — All applications, services, and systems should invoke authentication, access control, monitoring, and logging services via the same programming interface. The generalized system security application programming interface (GSSAPI) is preferred in the absence of any other overriding considerations. Using a single interface permits the replacement or enhancement of the security services with a minimum of disruption.

Encryption services — Standard encryption services should be available on every platform. These will include encryption, decryption, key management, and certificate management services. The Data Encryption Standard algorithm should be preferred for all applications save key management, where RSA is preferred. A public key server should be available in the network. This service will permit a user or an application to find the public key of any other.

Automate and hide all key management functions — All key management should be automated and hidden from users. No keys should ever appear in the clear or be transcribed by a user. Users should reference keys only by name. Prefer dedicated hardware for the storage of keys. Prefer smart cards, tokens, PCMCIA cards, other removable media, laptops, or access-controlled single user desktops in that order. Only keys belonging to the system manager should be stored on a multi-user system.

Use firewalls to localize and raise the cost of attacks — The network should be compartmented with firewalls. These will localize attacks, prevent them from spreading, increase their cost, and reduce the value of success.

Firewalls should resist attack traffic in both directions. That is, each subnetwork should use a firewall to connect to any other. A subnet manager should be responsible for protecting both his own net and connecting nets from any attack traffic. A conservative firewall policy is indicated. That is, firewalls should permit only that traffic which is necessary for the intended applications and should hide all information about one net from the other.

Access control begins on the desktop — Access control should begin on the desktop and be composed up rather than begin on the mainframe and spread down. The issue here is to prevent the insertion of malicious programs more than to prevent the leakage of sensitive data.

APPENDIX I

PRINCIPLES OF GOOD DESIGN

Prefer broad solutions to point solutions — Prefer broad security solutions which work across the enterprise, multiple applications, multiple resources, and against multiple hazards to those which are limited to or specific to one of these. Such practices are almost always more efficient than a collection of mechanisms that are specific to applications, resources, or hazards.

Prefer end-to-end solutions to point-by-point solutions — Similarly, prefer encryption-based end-to-end security solutions that are independent of the network. The more sensitive the application and the more hostile the network, the greater this preference. Such solutions are more robust and more efficient than those that attempt to identify and fix all of the vulnerabilities between the ends of the path.

Design top-down, implement bottom up — Design by functional decomposition and successive refinement. Implement by composition from the bottom. Prefer early deployment of those services and servers which will be required over the long haul.

Do it right the first time — When building infrastructure, build for the ages. Do it right the first time. This strategy is more effective and more efficient than the "assess and patch" strategy that has been the approach to security in the past.

Prefer planning to fixing — Similarly, work by plan and design rather than by experimentation. Necessary experimentation should be carefully identified, contained, and controlled.

Prefer long term to short — Applications are becoming more sensitive and the environment more hostile. While one may consent to a plan that permits an early deployment of an application with a plan to deploy the

agreed upon security function by a date certain, do not take a "wait and see" approach.

Justify across the enterprise and time — Security measures must be justified across the entire enterprise and across the life of the application or the mechanism. By definition, security prefers predictable, regular, prevention costs to unpredictable, irregular, remedial costs. They should be justified across a time frame that is consistent with the normal frequency of the events that it addresses. Security measures are relatively easy to justify in this manner and difficult to justify locally or in the short term. In justifying security measures, weight should be given to the fact that applications are becoming more sensitive, more interoperable, and more important, and that the environment in which they operate is becoming less reliable and more hostile.

Provide economy of safe use — Using the system safely should require as little user effort as possible. For example, a user should have to log on only once per enterprise, per workstation, per day.

Provide consistent presentation and appearance — Security should look the same across the enterprise, i.e., applications, systems, and platforms.

Make control predictable and intuitive — Systems should be supportive. They should encapsulate the special knowledge required by the manager and user to operate them. They should make this information available to the manager and user at the time of use.

Provide ease of safe use — Design in such a way that it is easy to do the right thing. Penalties should be associated with doing the wrong thing (e.g., economy of log-on, user should have to log on only once per workstation, per enterprise, per day.)

Prefer mechanisms that are obvious as to their intent — Avoid mechanisms which are complex or obscure, which might cause error, or be used to conceal malice. For example, prefer online transactions, EDI, secure formatted E-mail, formatted E-mail, E-mail, and file transfer in that order. The online transaction is always obvious and predictable; for a given set of inputs one can predict the outputs. While the intent of a file transfer may be obvious, it is not necessarily so.

Encapsulate necessary special knowledge — Necessary special knowledge should be included in documentation or programs.

Prefer simplicity; hide complexity — For example, all other things being equal, simple mechanisms should be preferred to complex ones. Prefer a single mechanism to two, a single instance of a mechanism should be preferred to multiple ones. For example, prefer a single appearance of administration, like CA Unicenter Star to the appearance of all the systems

which may be hidden by it. Similarly, prefer a single point of administration such as SAM or RAS to Unicenter Star.

Place controls close to the resource — As a rule and all other things being equal, controls should be as close to the resource as possible. The closer to the resource, the more reliable the control, the more resistant to interference, and the more resistant to bypass. Controls should be server-based, rather than client-based.

Place operation of the control as close as possible to where the knowledge is and where the effect can be observed — For example, prefer controls operated by the owner of the resource, the manager of the group, the manager of the system, and the manager of the user rather than by a surrogate such as a security administrator. While a surrogate has the necessary special knowledge to operate the control, he knows less about the intent and the effect of the control. He cannot observe the effect and take corrective action. Surrogates are often compensation for a missing, complex, or poorly designed control.

Prefer localized control and data — As a general rule and all other things being equal, prefer solutions that place reliance on as few controls in as few places as possible. Not only are such solutions more effective and efficient but they are also more easily apprehended, comprehended, and demonstrated. Distribute function and data as required or indicated for performance, reliability, availability, and use or control.

APPENDIX II

REFERENCES

IBM Security Architecture [SC28-8135-01]
ECMA 138 (SESAME) (see http://www.esat.kuleuven.ac.be/cosic/sesame3_2.html)
Open Systems Foundation Distributed Computing Architectures
(see http://www.osf.org/tech_foc.htm)

APPENDIX III

GLOSSARY

Architecture — That part of design that deals with appearance, function, location, and materials.

Authentication — The testing or reconciliation of evidence; reconciliation of evidence of user identity

Cryptography — The art of secret writing; the translation of information from a public code to a secret one and back again for the purpose of limiting access to it to a select few.

Distinguished User Name — User's full name so qualified as to be unique within a population. Qualifiers may include such things as enterprise name, organization unit, date of birth, etc.

Enterprise — The largest unit of organization; usually associated with ownership. (In government it is associated with sovereignty or democratic election.)

Enterprise Data — Data which are defined, meaningful, and used across business functions or for the strategic purposes of the enterprise.

Name Space — All of the possible names in a domain, whether used or not.

PIN — Personal Identification Number; evidence of personal identity when used with another form.

APPENDIX IV

PRODUCTS OF INTEREST

Secure authentication products — A number of clients and servers share a protocol for secure authentication. These include Novell Netware, Windows NT and Oracle Secure Network Services. A choice of these may meet some of the architectural requirements.

Single sign-on products — Likewise, there are a number of products on the market that meet some or all of the requirements for limited or single sign-on. These include SSO DACS from Mergent International, NetView Access Services from IBM, and NetSP.

- SSO DACS (Mergent International) (see http://www.pilgrim.umass.edu/pub/security/mergent.html)
- NetView Access Services (IBM) (see http://www.can.ibm.com/mainframe/software/sysman/p32.html)
- SuperSession (see http://www.candle.com/product_info/solutions/SOLCL.HTM)
- NetSP (IBM) (see http://www.raleigh.ibm.com/dce/dcesso.html)

Authentication services — A number of standard services are available for authenticating evidence of user identity. These include:

- Ace Server (see http://www.securid.com/ID188.100543212874/Security/ACEdata.html)
- TACACS (see http://sunsite.auc.dk/RFC/rfc/rfc1492.html)
- Radius (see http://www.tribe.com/support/TribeLink/RADIUS/RADIUSpaper.html)

Administrative services — There are a number of products that are intended for creating and maintaining access control data across a distributed computing environment. These include:

- Security Administration Manager (SAM) (Schumann, AG) (see http://www.schumann-ag.de/deutsch/sam/sam.html)
- RAS (Technologic) (see http://www.technologic.com/RAS/rashome.html)
- Omniguard Enterprise Security Manager (Axent) (http://www.axent.com:80/axent/products/products.html)
- Mergent Domain DACS (http://www.mergent.com/html/products.html)
- RYO (“Roll yer own”)

120

Security Infrastructure: Basics of Intrusion Detection Systems

Ken M. Shaurette, CISSP, CISA, NSA, IAM

An intrusion detection system (IDS) inspects all inbound and outbound network activity. Using signature and system configuration, it can be set up to identify suspicious patterns that may indicate a network or system attack. Unusual patterns, or patterns that are known to generally be attack signatures, can signify someone attempting to break into or compromise a system. The IDS can be a hardware- or software-based security service that monitors and analyzes system events for the purpose of finding and providing real-time or near-real-time warning of events that are identified by the configuration to be attempts to access system resources in an unauthorized manner (see Exhibit 120.1).

There are many ways that an IDS can be categorized:

- *Misuse detection.* In misuse detection, the IDS analyzes the information it gathers and compares it to databases of attack signatures. To be effective, this type of IDS depends on attacks that have already been documented. Like many virus detection systems, misuse detection software is only as good as the databases of attack signatures that it can use to compare packets.
- *Anomaly detection.* In anomaly detection, a baseline, or normal, is established. This consists of things such as the state of the network's traffic load, breakdown, protocol, and typical packet size. With anomaly detection, sensors monitor network segments to compare their present state against the baseline in order to identify anomalies.
- *Network-based system.* In a network-based system, or NIDS, the IDS sensors evaluate the individual packets that are flowing through a network. The NIDS detects malicious packets that are designed by an attacker to be overlooked by the simplistic filtering rules of many firewalls.
- *Host-based system.* In a host-based system, the IDS examines the activity on each individual computer or host. The kinds of items that are evaluated include modifications to important system files, abnormal or excessive CPU activity, and misuse of root or administrative rights.
- *Passive system.* In a passive system, the IDS detects a potential security breach, logs the information, and signals an alert. No direct action is taken by the system.
- *Reactive system.* In a reactive system, the IDS can respond in several ways to the suspicious activity such as by logging a user off the system, closing down the connection, or even reprogramming the firewall to block network traffic from the suspected malicious source.

Defense-in-Depth

Hacking is so prevalent that it is wrong to assume that it will not happen. Similar to insurance statistics, "the longer we go without being compromised, the closer we are to an incident." You do not buy flood insurance

EXHIBIT 120.1 Definitions

To better understand the requirements and benefits of an intrusion detection system, it is important to understand and be able to differentiate between some key terms. Some of that terminology is outlined below.

Anomaly — This is a technique used for identifying intrusion. It consists of determining deviations from normal operations. First, normal activity is established that can be compared to current activity. When current activity varies sufficiently from previously set normal activity, an intrusion is assumed.

Audit Logs — Most operating systems can generate logs of activity, often referred to as audit logs. These logs can be used to obtain information about authorized and unauthorized activity on the system. Some systems generate insufficient or difficult-to-obtain information in their audit logs and are supplemented with third-party tools and utilities (i.e., Top Secret for MVS). The term *audit* as it pertains to these logs is generally associated with the process to assess the activity contained in the logs. Procedures should exist to archive the logs for future review, as well as review security violations in the logs for appropriateness. As it pertains to intrusion detection, an audit approach to detection is usually based on batch processing of after-the-fact data.

False Negative/Positive — These are the alerts that may not be desired. Not identifying an activity when it actually was an intrusion is classified as a false negative. Crying wolf on activity that is not an actual intrusion is a false positive.

File Integrity Checking (FIC) — File integrity checking employs a cryptographic mechanism to create a signature of each file to be monitored. The signature is stored for further use for matching against future signatures of the same file. When a mismatch occurs, the file has been modified or deleted; and it must be determined whether intrusive activity has occurred. FIC is valuable for establishing a "golden" unmodified version of critical software releases or system files.

Hackers — The popular press has established this term to refer to individuals who gain unauthorized access to computer systems for the purpose of stealing and corrupting data. It is used to describe a person who misuses someone else's computer or communications technology. Hackers maintain that the proper term for such individuals is *cracker*, and they reserve the term *hacker* for people who look around computer systems to learn with no intent to damage or disrupt.

Honeypot — A honeypot is a system or file designed to look like a real system or file. It is designed to be attractive to the attacker to learn their tools and methods. It can also be used to help track the hacker to determine their identity and to help find out vulnerabilities. It is used to help keep an attacker off of the real production systems.

Intrusion Detection Systems — By definition, an intrusion detection system consists of the process of detecting unauthorized use of, or attack on, a computer or network. An IDS is software or hardware that can detect such misuse. Attacks can come from the Internet, authorized insiders who misuse privileges, and insiders attempting to gain unauthorized privileges. There are basically two kinds of intrusion detection — host-based and network-based — described below. Some products have become hybrids that combine features of both types of intrusion detection.

IDS System Types

Host Based — This intrusion detection involves installing pieces of software on the host to be monitored. The software uses log files and system auditing agents as sources of data. It looks for potential malicious activity on a specific computer in which it is installed. It involves not only watching traffic in and out of the system but also integrity checking of the files and watching for suspicious processes and activity. There are two major types: application specific and OS specific.

OS Specific — Based on monitoring OS log files and audit trails.

Application Specific — Designed to monitor a specific application server such as a database server or Web server.

Network Based — This form of intrusion detection monitors and captures traffic (packets) on the network. It uses the traffic on the network segment as its data source. It involves monitoring the packets on the network as they pass by the intrusion detection sensor. A network-based IDS usually consists of several single-purpose hosts that "sniff" or capture network traffic at various points in the network and report on the attacks based on attack signatures.

Incident Response Plan — This is the plan that has been set up to identify what is to be done when a system is suspected of being compromised. It includes the formation of a team that will provide the follow-up on the incident and the processes that are necessary to capture forensic evidence for potential prosecution of any criminal activity.

Penetration Testing — Penetration testing is the act of exploiting known vulnerabilities of systems and users. It focuses on the security architecture, system configuration, and policies of a system. Penetration tests are often purchased as a service from third-party vendors to regularly test the environment and report findings. Companies can purchase the equivalent software used by these service organizations to perform the penetration tests themselves. Penetration testing and vulnerability analysis (see below) are often confused and used by people to mean the same thing, differentiated technically by whether you are attempting to penetrate (access) vs. simply reporting on vulnerabilities (test, for existence) such as the presence or absence of security-related patches. Some penetration test software can identify an apparent vulnerability and provide the option of attempting to exploit it for verification.

Vulnerability Scanner — This tool collects data and identifies potential problems on hosts and network components. Scanners are the tools often used to do a vulnerability analysis and detect system and network exposures. A scanner can identify such things as systems that do not have current patch levels, software and installation bugs, or poor configuration topology and protocols. A scanner does not enforce policy or fix exposures; it purely identifies and reports on them.

Vulnerability Analysis (also called vulnerability assessment) — Vulnerability analysis is the act of checking networks or hosts to determine if they are susceptible to attack, not attempting to exploit the vulnerability. The process consists of scanning servers or networks for known vulnerabilities or attack signatures to determine whether security mechanisms have been implemented with proper security configuration, or if poor security design can be identified. A form of vulnerability assessment is to use a product to scan sets of servers for exposures that it can detect.

in the 99th year before the 100-year flood. Although keeping hackers away from your company data is virtually impossible, much can be done to reduce vulnerabilities. Hacker have the easiest task; they need find only one open door. As the defenders, a company must check every lock, monitor every hallway. A company will implement a variety of sound security mechanisms such as authentication, firewalls, and access control; but there is still the potential that systems are unknowingly exposed to threats from employees and nonemployees (from inside and from outside). Layering security or using generally accepted practices for what is today often called a *defense-in-depth* requires more.

The complexity of the overall corporate environment and disparity of knowledge for security professionals subject implemented protection mechanisms to improper configuration, poor security design, or malicious misuse by trusted employees or vendor/contract personnel. Today's intrusions are attacks that exploit the vulnerabilities that are inherent in operating systems such as NT or UNIX. Vulnerabilities in network protocols and operating system utilities (i.e., telnet, FTP, traceroute, SNMP, SMTP, etc.) are used to perform unauthorized actions such as gaining system privileges, obtaining access to unauthorized accounts, or rerouting network traffic.

The hacker preys on systems that:

- Do not lock out users after unsuccessful log-in attempts
- Allow users to assign dictionary words as passwords
- Lack basic password content controls
- Define generic user IDs and assign password defaults that do not get changed
- Do not enforce password aging

Two-factor authentication is still expensive and slow to gain widespread adoption in large organizations. Using two factors — something you have and something you know — is one of the best methods to improve basic access control and thwart many simple intrusions.

A company that does not have a comprehensive view of where its network and system infrastructure stands in terms of security lacks the essentials to make informed decisions. This is something that should be resolved with the cooperation and support of all a company's IS technology areas. A baseline identifying gaps or places for improvement must be created. An IDS requirements proposal or any other security improvement proposal will require coordination with all infrastructure technicians to be effective. Companies need to have a dynamic information security infrastructure.

Although no organization relishes the idea of a system intrusion, there is some comfort that, with the right tools, it is possible to reduce exposures and vulnerabilities — but not necessarily eliminate all of the threats. There will always be some exposure in the environment. It is virtually impossible to remove them all and still have a functional system. However, measures to reduce impact of compromise can be put in place, such as incident response (what to do when), redundancy, traps (honeypots), prosecution (forensic evidence), and identification (logging). In order for it to be easier to track a hacker's activity, proper tools are needed to spot and plug vulnerabilities as well as to capture forensic evidence that can be used to prosecute the intruder. Intrusion detection systems are complex to implement, especially in a large environment. They can generate enormous quantities of data and require significant commitments in time to configure and manage properly. As such, an IDS has limitations that must be considered when undertaking selection and deployment. Even so, intrusion detection is a critical addition to an organization's security framework; but do not bother without also planning at least rudimentary incident response.

What to Look for in an IDS

Vendors are searching for the next generation, a predictive IDS — an IDS that can flag an attack without being burdened by the weight of its own logs and can operate worry-free with minimal false alarms. There are many shapes, sizes, and ways to implement an IDS. A rule-based model relies on preset rules and attack signatures to identify what to alert on and review. Anomaly-based systems build their own baselines over time by generating a database of usage patterns: when usage is outside the identified norm, an alert or alarm is set off. In addition, placement of an IDS is important especially when it comes to determining host- or network-based or the need for both.

A typical weakness in rule-based systems is that they require frequent updates and risk missing new or yet-unidentified attack patterns. An anomaly system attempts to solve this but tends to be plagued by false alarms. Often, companies install and maintain the host-based IDS on only production systems. Test hosts are often

the entry point for an attacker and, as such, require monitoring for intrusion as well. The next generation of IDS will correlate the fact that an intrusion has occurred, is occurring, or is likely to occur. It will use indicators and warnings, network monitoring and management data, known vulnerabilities, and threats to arrive at a recommended recovery process.

Some intrusion detection systems introduce the ability to have a real-time eye on what is happening on the network and operating systems. Many of the leading products offer similar features, so the choice of product can boil down to the fine details of how well the product will integrate into a company's environment as well as meet the company's incident response procedures. For example, one vendor's product may be a good fit for network detection in a switched network, but does not provide any host intrusion detection, or it misses traffic on other segments of the network.

For intrusion detection to be a useful tool, the network and all of the hosts under watch should have a known security state. A company must be first willing to apply patches for known vulnerabilities. Most of the vulnerability assessment tools can find the vulnerabilities, and these are what the intrusion detection tools monitor for exploitation. The anomaly-based system relies on the fact that most attacks fit a known profile. Usually this means that by the time the IDS system can detect an attack, the attack is preventable and patches are available. Security patches are a high priority among most if not all product vendors, and they appear rapidly if they are actively exploited. Therefore, it might be more effective to first discover the security posture of the network and hosts, bring them up to a base level of security, and identify maintenance procedures to stay at that desired level of security. Once that is accomplished, IDS can more effectively contribute to the overall security of the environment. It becomes a layer of the defense that has value.

Getting Ready

Although many organizations are not aware of them, there are laws to address intrusion and hacking. There are an even greater number of organizations that are not prepared to take advantage of the laws. For example, the Federal Computer Fraud and Abuse Act was updated in 1996 to reflect problems such as viruses sent via e-mail (Melissa, Bubble-Boy). In fact, the law was used to help prosecute the Melissa virus author. In addition, this same law addresses crimes of unauthorized access to any computer system, which would include nonvirus-related intrusions. DoS (denial-of-service) attacks have become very common, but they are no joking matter. In the United States, they can be a serious federal crime under the National Infrastructure Protection Act of 1996 with penalties that include years of imprisonment. Many countries have similar laws. For more information on computer crimes, refer to www.usdoj.gov/criminal/cybercrime/compcrime.html.

Laws are of little help if a company is unable to recognize an event is occurring, react to it, and produce forensic evidence of the crime. Forensic computer evidence is required for prosecution of a crime. Not every system log is appropriate as forensic evidence. Logs must maintain very specific qualities and should document system activity and log-in/log-out type activity for all computers on the network. These allow a prosecutor to identify who has accessed what and when. Also important is the process for gathering and protecting any collected information (the chain of custody) in order for the information to retain forensic value. This process should be part of a comprehensive incident response plan. IDS without intrusion response, including an incident response plan, essentially reduces its value. The IDS effectively becomes merely another set of unused log data.

Even more important than prosecution as a reason for maintaining forensic data, the company's network technicians can use the forensic evidence to determine how a hacker gained access in order to close the hole. The data can also be necessary to determine what was done when the attacker was inside the network. It can be used to help mitigate the damage. In many cases, companies are still rarely interested in the expense, effort, and publicity involved in prosecution.

A company must perform a thorough requirements analysis before selecting an intrusion detection system strategy and product. A return on investment (ROI) can be difficult to calculate; but in any case, costs and benefits need to be identified and weighted. Refer to Exhibit 120.2 for a discussion on cost/benefit analyses (CBA) and ROI. A solution must be compatible with the organization's network infrastructure, host servers, and overall security philosophy and security policies. There can be a big variance of resource (especially human) requirements among the different tools and methodologies. Both network and server teams must work together to analyze the status of an organization's security posture. (i.e., systems not patched for known vulnerabilities, weak password schemes for access control, poor control over root or administrative access). There may be

EXHIBIT 120.2 Cost/Benefit Analysis — Return on Investment

Risk Management to Improve Enterprise Security Infrastructure

Effective protection of information assets identifies the information used by an area and assigns primary responsibility for its protection to the management of the respective functional area that the data supports. These functional area managers can accept the risk to data that belongs to them, but they cannot accept exposures that put the data of other managers at risk.

Every asset has value. Performing an analysis of business assets — and the impact of any loss or damage resulting from the loss — is necessary to determine the benefits of any actual dollar or human time expenditures to improve the security infrastructure. A formal quantitative risk analysis is not necessary, but generally assessing the risks and taking actions to manage them can pay dividends. It will never be possible to eliminate all risks; the trick is to manage them. Sometimes it may be desirable to accept the risks, but it is a must to identify acceptance criteria. The most difficult part of any quantifiable risk management is assigning value and annual loss expectancy (ALE) to intangible assets like a customer's lost confidence, potential embarrassment to the company, or various legal liabilities. To provide a risk analysis, a company must consider two primary questions.

1. What is the probability that something will go wrong (*probability* of one event)?
2. What is the cost if something does go wrong (the *exposure* of one event)?

Risk is determined by getting answers to the above questions for various vulnerabilities and assessing the probability and impact of the vulnerability on each risk.

A quantifiable way to determine the risk and justify the cost associated with purchase of an IDS or any other security software or costs associated with mitigating risks is as follows:

- Risk becomes the probability times the exposure (risk = probability × exposure). Cost justification becomes the risk minus the cost to mitigate the vulnerabilities (justification = risk minus cost of security solution). If the justification is a positive number, then it is costjustified. For example, if the potential loss (exposure) on a system is $100,000, and the chance that the loss will be realized (probability) is about once in every ten years, the annual frequency rate (AFR) would be 1/10 (0.10). The risk (ALE) would be $100,000 × 0.10 = $10,000 per year. If the cost is $5000 to minimize the risk by purchasing security software, the cost justification would be $10,000 less $5000 = $5000, or payback in six months.
- Using a less quantifiable method, it would be possible to assign baseline security measures used in other similar sized companies, including other companies in the same industry. Setting levels of due diligence that are accepted in the industry would then require implementation of controls that are already proven, generally used, and founded on the "standard of due care." For example, for illustration purposes, say that 70 percent of other companies the size of your company are implementing intrusion detection systems and creating incident response teams. Management would be expected to provide similar controls as a "standard of due care." Unless it can be clearly proven that implementation costs of such measures are above the company's expected risks and loss expectancies, management would be expected to provide due diligence in purchasing and implementing similar controls.

many areas of basic information security infrastructure that require attention before IDS cost can be justified. The evaluations could indicate that simply selecting and implementing another security technology (IDS) is wasted money. A company may already own technologies that are not fully implemented or properly supported that could provide compensating controls and for which cost could be more easily justified.

When it comes to a comprehensive IDS, integration between server and network environments is critical. A simple decision such as whether the same tool should provide both network and host IDS is critical in the selection process and eliminates many tools from consideration that are unable to provide both. Even simply identifying integration requirements between operating systems will place limitations and requirements on technology selection. Does a company want to simply detect an intrusion, or is it desirable to also track the activity such as in a honeypot? Honeypots are designed to be compromised by an attacker. Once compromised, they can be used for a variety of purposes, such as an alert, an intrusion detection mechanism, or as a deception. Honeypots were first discussed in a couple of very good books: Cliff Stoll's *Cuckoo's Egg*[1] and Bill Cheswick's *An Evening with Berferd*.[2] These two reviews used a capture-type technology to gather an intruder's sessions. The sessions were then monitored in detail to determine what the intruder was doing.

Steps for Protecting Systems

To continue improving the process of protecting the company systems, three fundamental actions are required.

Action 1

The company must demonstrate a willingness to commit resources (money, people, and time) to patching the basic vulnerabilities in current systems and networks as well as prioritize security for networks and hosts.

Making use of an IDS goes way beyond simply installing the software and configuring the sensors and monitors. It means having necessary resources, both technical and human, to customize, react, monitor, and correct. Nearly all systems should meet basic levels of security protection. Simple standards such as password aging, improved content controls, and elimination of accounts with fixed passwords or default passwords are a step in that direction. It is also critical that all network and operating systems have current security patches installed to address known vulnerabilities and that maintenance procedures exist to keep systems updated as new alerts and vulnerabilities are found.

Action 2

All systems and network administrators must demonstrate the security skills and focus to eliminate basic vulnerabilities by maintaining and designing basic secure systems — which, poorly done, account for the majority of attacks.

Nearly all system and network administrators want to know how to secure their systems, but in many cases they have never received actual security training or been given security as a priority in their system design. Often, security is never identified as a critical part of job responsibility. It should be included in employee job descriptions and referenced during employee performance reviews. However, before this can be used as a performance review measurement, management must provide staff with opportunity (time away from office) and the priority to make security training part of job position expectations. Training should be made available in such topics as system security exposures, vulnerability testing, common attacks and solutions, firewall design and configuration, as well as other general security skills. For example, the effectiveness of any selected IDS tool is dependent on who monitors the console — a skilled security expert or an inexperienced computer operator. Even a fairly seasoned security expert may not know how to respond to every alert.

Action 3

Once security expectations are in place, tasks must be given proper emphasis. Staff members must recognize that security is part of their job and that they must remain properly trained in security. Security training should receive the same attention as the training they receive on the system and network technologies they support. Security must be given similar time and resources as other aspects of the job, especially defining and following maintenance procedures so that systems remain updated and secure.

Network and system administrators will need to stay current with the technology they support. Often they will attend training to stay current, but not to understand security because it is not sufficiently recognized as important to their job responsibilities.

These tasks will not stop all attacks but they will make a company a lot less inviting to any criminal looking for easy pickings. Typical attackers first case their target. When they come knocking, encourage them to go knocking on your neighbor's door — someone who has not put security measures in place. Putting the fundamentals in place to monitor and maintain the systems will discourage and prevent common external intrusion attempts as well as reduce internal incidents.

Types of Intrusion

Intrusions can be categorized into two main classes:

1. *Misuse intrusions* are well-defined attacks on known weak points of a system. They can be detected by watching for certain actions performed on certain objects. A set of rules determines what is considered misuse.

2. *Anomaly intrusions* are based on the observation of the deviation from normal system activity. An anomaly is detected by building a profile of the system monitored, followed by using some methodology for detecting significant deviations from this profile.

Misuse intrusions can be detected by doing pattern matching on audit-trail information because they follow well-defined patterns. For example, examining log messages of password failures can catch an attempt to log on or set user ID to root from unauthorized accounts or addresses.

Anomalous intrusions are a bit more difficult to identify. The first difficulty is identifying what is considered normal system activity. The best IDS is able to learn system and network traffic and correlate it to the time of day, day of week, and recognize changes. Exploitation of a system's vulnerabilities usually involves the hacker performing abnormal use of the system; therefore, certain kinds of system activity would be detected from normal patterns of system usage and flagged as potential intrusion situations. To detect an anomaly intrusion, it is necessary to observe significant deviations from the normal system behavior from the baseline set in a *profile*. A quantitative measure of normal activity can be identified over a period of time by measuring the daily activity of a system or network. For example, the average or a range of normal CPU activity can be measured and matched against daily activity. Significant variations in the number of network connections, an increase or decrease in average number of processes running in the system per minute, or a sudden sustained spike in CPU utilization when it does not normally occur could signify intrusion activity. Each anomaly or deviation may signal the symptoms of a possible intrusion. The challenge is mining the captured data and correlating one element of data to other captured data and determining what the two together might signify.

Characteristics of a Good Intrusion Detection System

There are several issues an IDS should address. Regardless of the mechanism on which it is based, it should include the following:

- Run continually with minimal human interaction. It should run in the background. The internal workings should be able to be examined from outside, so it is not a black box.
- Fault tolerance is necessary so that it can survive a system crash and not require that its knowledge base be rebuilt at restart.
- It must be difficult to sabotage. The system should be self-healing in the sense that it should be able to monitor itself for suspicious activities that might signify attempts to weaken the detection mechanism or shut it off.
- Performance is critical. If it creates performance problems, it will not get used.
- Deviations from normal behavior need to be observed.
- The IDS must be easy to configure to the system it is monitoring. Every system has a different usage pattern, and the defense mechanism should adapt easily to these patterns.
- It should be like a chameleon, adapting to its environment and staying current with the system as it changes — new applications added, upgrades, and any other modifications. The IDS must adapt to the changes of the system.
- To be effective, an IDS must have built-in defense mechanisms, and the environment around it should be hardened to make it difficult to fool.

Watch Out for Potential Network IDS Problems

ACIRI (AT&T Center for Internet Research at the International Computer Science Institute) does research on Internet architecture and related networking issues. Research has identified that a problem for a NIDS is its ability to detect a skilled attacker who desires to evade detection by exploiting the uncertainty or ambiguity in the traffic's data stream. The ability to address this problem introduces a network-forwarding element called a *traffic normalizer*. The normalizer needs to sit directly in the path of traffic coming into a site. Its purpose is to modify the packet stream to eliminate potential ambiguities before the monitor sees the traffic. Doing this removes evasion opportunities. There are a number of tradeoffs in designing a normalizer. Mark Handley and Vern Paxson discuss these in more detail in their paper titled "*Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics*." In the paper they emphasize the important question pertaining to the degree to which normalizations can undermine end-to-end protocol semantics. Also discussed

are the key practical issues of "cold start" and attacks on the normalizer. The paper shows how to develop a methodology for systematically examining the ambiguities present in a protocol based on walking the protocol's header. Refer to the notes at the end of this chapter to find more information on the paper.

Methodology for Choosing and Implementing an IDS

To choose the best IDS, evaluation is necessary of how well the tool can provide recognition of the two main classes of intrusion. Specific steps should be followed to make the best selection. Some of the steps are:

1. Form a team representing impacted areas, including network and server teams.
2. Identify a matrix of intrusion detection requirements and prioritize, including platform requirements, detection methodology (statistical or real-time), cost, resource commitments, etc.
3. Determine preferences for purchasing IDS software versus using a managed service.
4. Determine if the same product should provide both network- and host-based IDS.
5. Formulate questions that need to be answered about each product.
6. Diagram the network to understand what hosts, subnets, routers, gateways, and other network devices are a part of the infrastructure.
7. Establish priority for security actions such as patching known vulnerabilities.
8. Identify IDS sensor locations (critical systems and network segments).
9. Identify and establish monitoring and maintenance policies and procedures.
10. Create an intrusion response plan, including creation of an incident response team.

Suspicion of Compromise

Before doing anything, *define an incident*. Incident handling can be very tricky, politically charged, and sensitive. The IDS can flag an incident, but next is determining what first-level support will do when an alert is received or identifying what to do in case of a *real* incident. This is critical to the system reaching its full value.

An IDS can be configured to take an action based on the different characteristics of the types of alerts, their severity, and the targeted host. In some cases it may be necessary to handle an incident like a potential crime. The evidence must be preserved similar to a police crime scene. Like a police crime scene that is taped off to prevent evidence contamination, any logs that prove unauthorized activity and what was actually done must be preserved. Inappropriate actions by anyone involved can cause the loss of valuable forensic evidence, perhaps even tip off the intruder, and cause a bigger problem. An incident response program can be critical to proper actions and provide consistency when reacting to intrusion activity. Without documented procedures, the system and network administrators risk taking the wrong actions when trying to fix what might be broken and contaminating or even eliminating evidence of the incident.

The following outlines considerations for incident response:

- Scream loudly and get hysterical — your system has been compromised.
- Brew up a few pots of strong coffee.
- Actually, you need to remain calm — do not hurry.
- Create a documented incident handling procedure, including options if possible.
- Notify management and legal authorities as outlined in the incident response plan.
- Apply the need-to-know security principle — only inform those personnel with a need to know. The fewer the people who are informed about the incident, the better; but be sure to prevent rumors by supplying enough information to the right people.
- Use out-of-band communications and avoid e-mail and other network-based communication channels — they may be compromised.
- Determine the items you need to preserve as forensic evidence (i.e., IDS log files, attacked system's hard drive, snapshot of system memory, and protection and safety logs).
- Take good notes — the notes may be needed as evidence in a court of law. Relying on your memory is not a good idea. This will be a stressful time, and facts may become fuzzy after everything calms down.
- Back up the systems; collect forensic evidence and protect it from modification. Ensure a chain of custody for the information.

- Contain the problem and pull the network cable? Is shutting off the system appropriate at this point? Is rebooting the system appropriate? It might not be!
- Eradicate the problem and get back to business.
- Use what has been learned from the incident to apply modifications to the process and improve the incident response methodology for future situations.

Summary

Before doing anything, define an incident. Know what you are detecting so that you know what you are handling.

Every year thousands of computers are illegally accessed because of weak passwords. How many companies have users who are guilty of any of the following?

- Writing down a password on a sticky note placed on or near their computer
- Using a word found in a dictionary.
- Using a word from a dictionary followed by two or less numerics
- Using the names of people, places, pets, or other common items
- Sharing their password with someone else
- Using the same password for more than one account, and for an extended period of time
- Using the default password provided by the vendor

Chances are, like the majority of companies, the answer is yes to one or more of the above. This is a more basic flaw in overall security infrastructure and requires attention. The problem is, hackers are aware of these problems as well and target those who do not take the correct precautions. This makes systems very vulnerable, and more than simple technology is necessary to correct these problems.

If a company's current security posture (infrastructure) is unacceptable, it must be improved for additional security technology to provide much added benefit. Performing an assessment of the present security posture provides the information necessary to adequately determine a cost–benefit analysis or return on investment. Implementing all the best technology does not eliminate the basic exposure introduced by the basic problem described above. A team should be created to identify current protection mechanisms as well as other measures that could be taken to improve overall security infrastructure for the company. Immediate benefits could be realized by enhancement to procedures, security awareness, and better implementation of existing products (access control and password content) with minimum investment. The overall security improvement assessment could include a project to select and implement an intrusion detection system (IDS) and incident response (IR) programs. IDS without IR is essentially worthless. First steps are for management to identify a team to look into necessary security infrastructure improvements. From this team, recommendations will be made for security improvements and the requirements against which products can be judged to help reduce security vulnerabilities while being an enabler of company business objectives.

Now that you have the IDS deployed and working properly, it is possible to kick back and relax. Not yet — in fact, the cycle has just begun. IDS, although a critical component of the defense-in-depth for an organization's security infrastructure, is just that — only a component.

References

1. C. Stoll, *The Cuckoo's Egg: Tracking a Spy through the Maze of Computer Espionage,* New York: Pocket Books, 1990.
2. Bill Cheswick, An Evening with Berferd in which a Cracker is Lured, Endured, and Studied, http://www.securityfocus.com/library/1793.
3. Intrusion Detection Pages, http://www.cerias.purdue.edu/coast/intrusion-detection/.
4. Mark Handley and Vern Paxson, Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics, http://www.aciri.org/vern/papers/norm-usenix-sec-01-html/norm.html.
5. http://www.aciri.org/vern/papers/norm-usenix-sec-01.ps.gz.
6. http://www.aciri.org/vern/papers/norm-usenix-sec-01.pdf.
7. http://www.usenix.org/events/sec01/handley.html.

Systems Integrity Engineering

Don Evans

INTRODUCTION

The primary goal of any enterprise-wide security program is to support user communities by providing cost-effective protection to information system resources at appropriate levels of integrity, availability, and confidentiality without impacting productivity, innovation, and creativity in advancing technology within the corporation's overall objectives.

Ideally, information systems security enables management to have confidence that their computational systems will provide the information requested and expected, while denying accessibility to those who have no right to it. The analysis of incidents resulting in damage to information systems show that most losses were still due to errors or omissions by authorized users, actions of disgruntled employees, and an increase in external penetrations of systems by outsiders. Traditional controls are normally inadequate in these cases or are focused on the wrong threat, resulting in the exposure of a vulnerability.

There are so many factors influencing security in today's complex computing environments that a structured approach to managing information resources and associated risk(s) is essential. New requirements for using distributed processing capabilities introduces the need to change the way integrity, reliability, and security are applied across diverse, cooperative information systems environments. The demand for high-integrity systems that ensure a sustained level of confidence and consistency must be instituted at the inception of a system design, implementation, or change. The formal process for managing security must be linked intrinsically to the existing processes for designing, delivering, operating, and modifying systems to achieve this objective.

Unfortunately, the prevalent attitude toward security by management and even some security personnel is that the confidentiality of data is still the primary security issue. That is, physical isolation, access control,

audit, and sometimes encryption are the security tools most needed. While data confidentiality may be an issue in some cases, it is usually more important that data and/or process integrity and availability be assured. Integrity and availability must be addressed as well as ensuring that the total security capability keeps current with technology advancements that make it easier to share geographically distributed computing resources.

As the complexity of today's distributed computing environments continues to evolve independently, with respect to geographical and technological barriers, the demand for a dynamic, synergistically integrated, and comprehensive information systems security control methodology increases.

Business environments have introduced significant opportunity for process reengineering, interdisciplinary synergism, increased productivity, profitability, and continuous improvement. With each introduction of a new information technology, there exists the potential for an increased number of threats, vulnerabilities, and risk. This is the added cost of doing business. These costs focus on systems failure and loss of critical data. These costs may be too great to recover with respect to mission- and/or life-critical systems. Enterprise-wide security programs, therefore, must be integrated into a systems integrity engineering discipline carried out at each level of the organization and permeated throughout the organization.

The purpose of this document is to provide an understanding of risk accountability issues and management's responsibility for exercising due care and due diligence in developing and protecting enterprise-wide, interoperable information resources as a synergistic organizational function.

UNDERSTANDING DISTRIBUTED PROCESSING CONCEPTS AND CORRESPONDING SECURITY-RELEVANT ISSUES

Distributed systems are an organized collection of programs, data, and processes implemented in software, firmware, or hardware that are specifically designed to integrate separate operational systems into a single, logical information system infrastructure. This structure provides the flexibility of segmenting management control into domains or nodes of processing that are physically required or are operationally more effective and efficient, while satisfying the overall goals of the information processing community.

The operational environment for distributed systems is a combination of multiple separate environments that may individually or collectively store and process information. The controls over each operational environment must be based on a common integrated set of security controls that constitute the foundation for overall information security of the distributed systems.

The foundation of security-relevant requirements for distributed systems is derived from the requirements specified in the following areas:

- Operating systems and support software,
- Information access control,
- Application software development and maintenance,
- Application controls and security,
- Telecommunications,
- Satisfaction of the need for cost-effective business objectives.

Distributed systems must also address a common set of security practices, procedures, and processes because of the interaction of separate operational environments which include:

1. A multiplicity of components, including both physical and logical resources, that can be assigned freely to specific tasks on a dynamic basis. (Homogeneity of physical resources is not essential.) However, in general, there should be more than one resource capable of supporting any given task to maintain referential integrity of the information and the complexity of the connectivity interrelationships of heteromorphic processing environments.
2. A physical distribution of these physical and logical components intercommunicating through a network. Within the distributed system environment, a network is an information transmission mechanism that uses a cooperative protocol to control the transfer of information.
3. A high-level operating system that unifies and integrates the control of the distribution components. This high-level operating system may not exist as distinctly identifiable blocks of code. It may be merely a set of specifications or an overall, integrating philosophy incorporated into the design of the operating system for each component.
4. System transparency, permitting services to be requested by name only. The resource to provide the service may not need to be uniquely identified.
5. Cooperative autonomy, characterizing the operation as an interaction of both physical and logical resources.

These five criteria form an indivisible set that defines a fully distributed system. The degree of distribution of a system depends upon the distribution of data, programs, physical hardware location, and control. This is depicted in Exhibit 1.

To simplify this three-dimensional continuum, distributed systems may be classified into three nonoverlapping parts of the continuum, ranging from simple interactions to complex interactions of the environments. The three types of distributed systems, illustrated in Exhibit 1, are

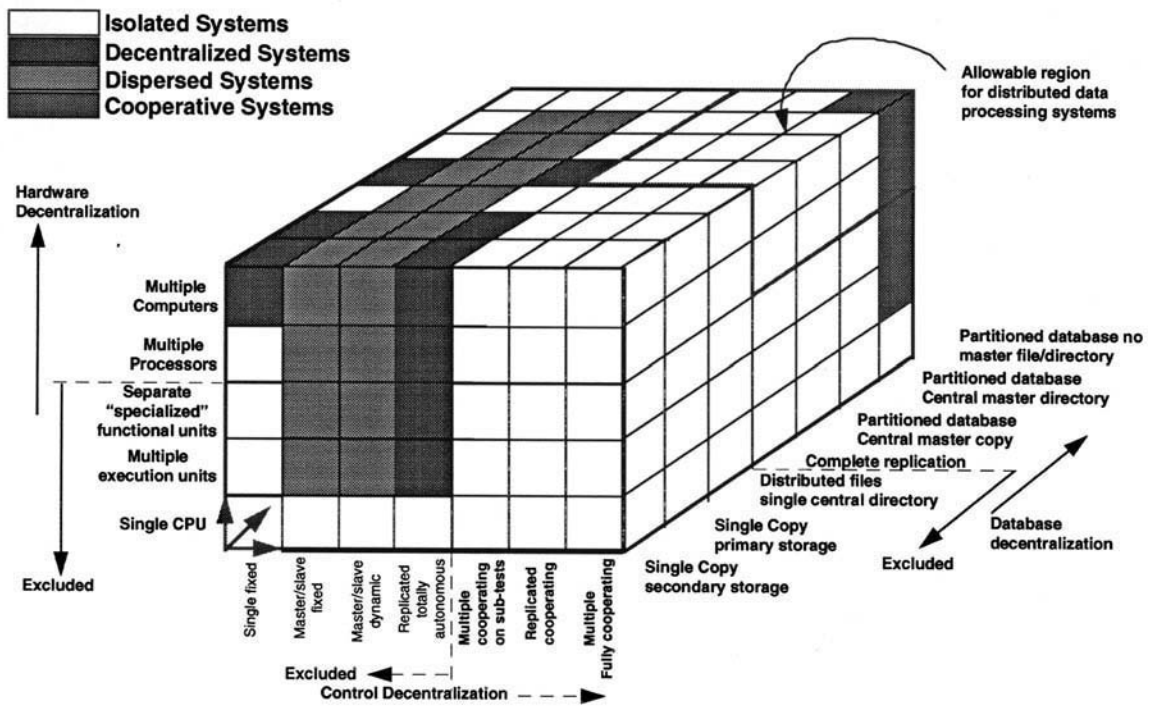


Exhibit 1. Distribution Continuum

- Decentralized systems
- Dispersed systems
- Interoperable or Cooperative systems

Decentralized systems are characterized by a group of related but not necessarily interconnected platforms running independent copies of the same (or equivalent) applications with independent copies of data. The current state of the group is not automatically maintained. Instead of a single (central) processor with multiple users, the decentralized system has multiple (distributed) processors with single or multiple users (Exhibit 2). The processors do not necessarily communicate electronically. This characteristic prevents the system from automatically maintaining the state of the distributed system and is the primary distinction between the decentralized model and the other two distributed system models.

Dispersed systems (Exhibit 3) are characterized by a group of related, interconnected platforms in which either the data or the software (but not both) is centralized. A dispersed system offers advantages over centralized systems in its capabilities to:

- Accommodate organizational change
- More effectively deploy resources through resource sharing
- Improve performance through intelligent matching of applications, media, access schemes, and grouping of related members
- Lower risk of overall system failure due to hardware failures

The dispersed system may have centralized data with dispersed processors (as in a system with a central file server) or centralized processing with dispersed data (as with remote transaction collection and central data processing). Dispersed systems may exist on multiple platforms in a single location or on platforms in multiple locations. The hardware may be homogeneous or heterogeneous.

The processors communicate electronically, usually to request or provide data. This characteristic allows the system to automatically maintain a single, collective, real-time state of the distributed system.

Interoperable or cooperative systems (Exhibit 4) are characterized by a group of related, interconnected platforms in which both the data and the software are distributed throughout the system. The interoperable system differs from the dispersed system by eliminating the dependency of centralized data or centralized applications. The interoperable system offers the same advantages over centralized systems as the dispersed system. The difference is in the degree to which the system can cooperatively exploit these advantages.

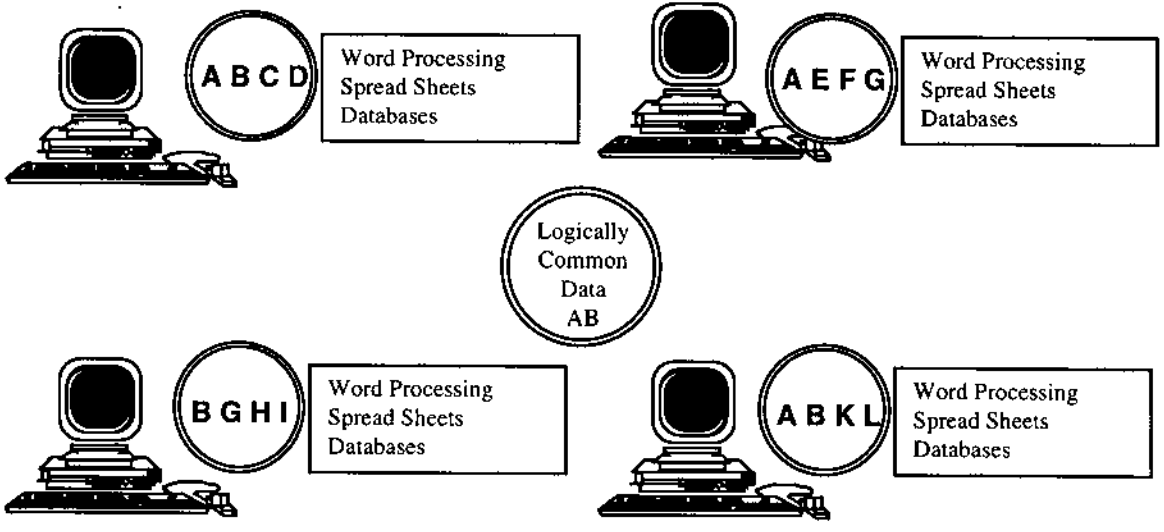


Exhibit 2. Decentralized Systems

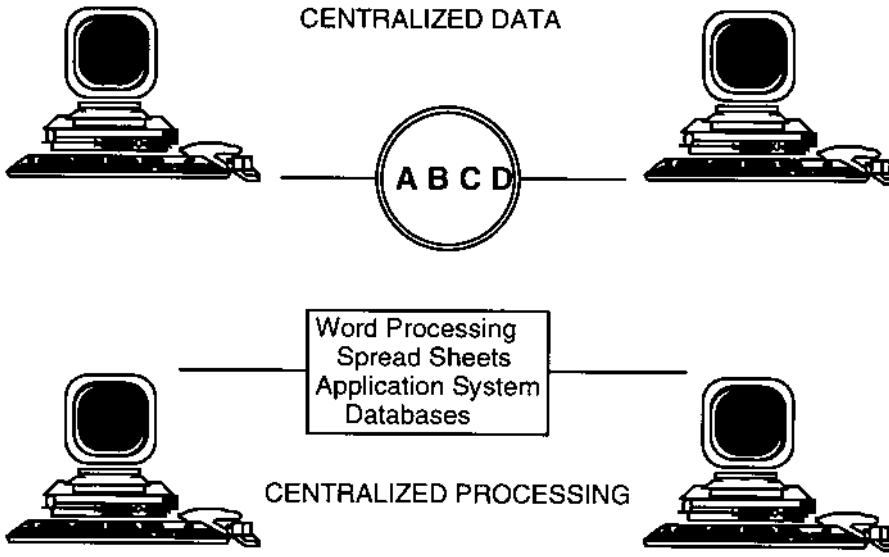


Exhibit 3. Dispersed Systems

Additionally, an interoperable system offers advantages over centralized systems in its capabilities to:

- Combine data from dissimilar hardware platforms
- Independently execute and test each component

Interoperable systems represent the highest level of the distributed processing continuum. In a fully interoperable system, each component is independent of all other components. Interfaces and data dependencies are implemented as messaging schemes or as data objects (consisting of data and operations). Interoperable systems may exist on multiple platforms in a single location, on platforms in multiple locations, or on multiple networks in multiple locations.

The hardware may be homogeneous or heterogeneous. The processors communicate electronically. Each component automatically maintains its own state and can provide its state on request. The existence of multiple states is the primary discriminant between the interoperable model and the other two distributed system models.

A distributed system may include characteristics of each of the three models described above. The application of security-relevant requirements from each model is necessary to build a complete security requirements set.

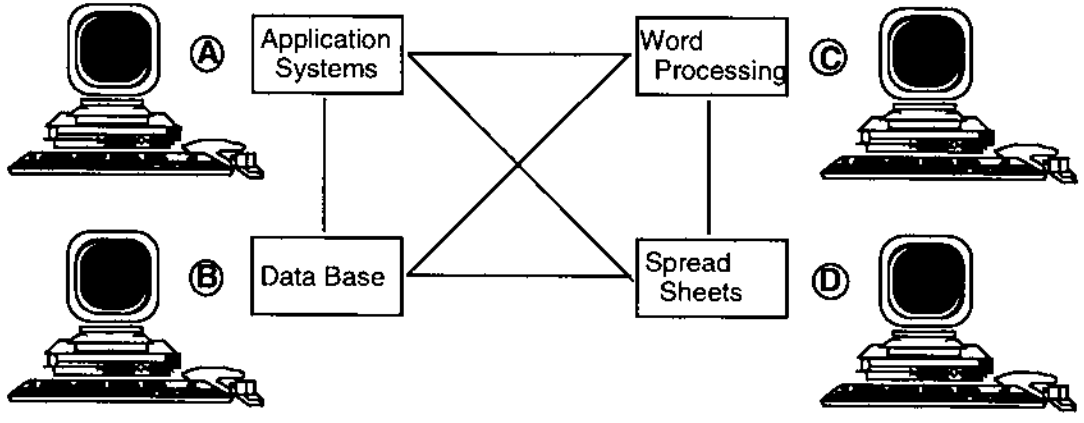


Exhibit 4. Interoperable Systems

Distributed Systems Integrity Control Issues

A system of controls for distributed (i.e., decentralized, dispersed, and cooperative) systems will need to be developed that addresses:

- Multisystem configuration management
- Establishing and maintaining connectivity
- Prevention of exploitation of connectivity
- Multilevel, multisite information transfers
- Contingency planning, backup, and recovery

Distributed systems are depicted in the three-dimensional continuum (Exhibit 5) represented by the simplest decentralized case in one bottom corner (centralized remote processing) and the most complicated cooperative case (fully interoperable system of systems) in the opposite top corner. Decentralized systems represent a stepwise departure from centralized processing and isolated system(s) controls.

For any two related systems, there generally exists some data common to the two systems. The larger the amount of common data and the more dynamic the data are, the more vulnerable the decentralized system is to integrity loss. Configuration management of the changes to common data, applications, and hardware can reduce the vulnerability to integrity loss. In addition, the processes for updating common data, applications, and hardware require controls to ensure that the approved changes and only the approved changes are received and installed.

Analysis from multiple systems may produce erroneous or tainted results caused by the inability to synchronize the data. If any correlation of time-based transactions from different platforms is required, these systems require either a synchronous time source or manual synchronization and periodic verification.

In implementations of a decentralized system where two identical (or equivalent) software applications and/or hardware platforms exist, users must periodically switch processing roles as part of planning, training, and disaster preparedness. The following suggestions are provided as guidelines for establishing a baseline set of controls that ensure high integrity and minimal risk accountability for managing distributed systems.

All common data, hardware, software, and each component system should be identified formally in a Distributed System Configuration Management (CM) Plan. Distributed System CM Plans must document system-level policies, standards and procedures, responsibilities, and requirements. For distributed systems where the nodes are not located at one site or where the components are not covered in a single CM Plan, management will need to appoint a Configuration Control Authority for all distributed

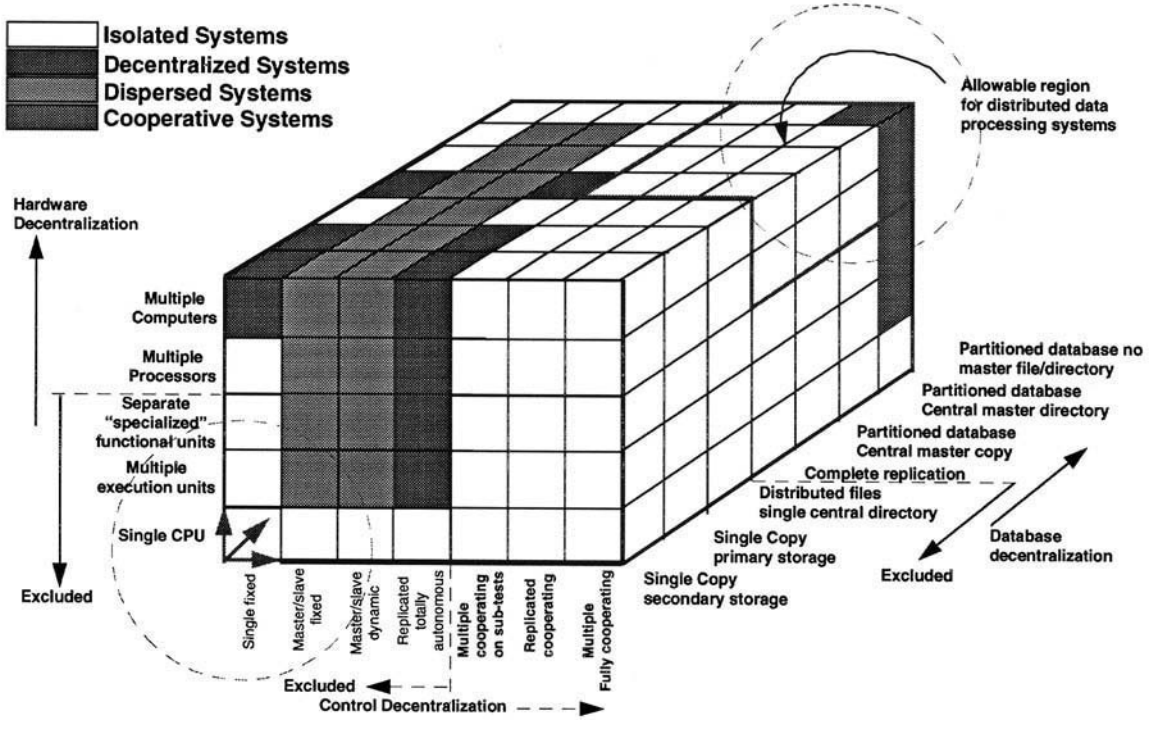


Exhibit 5. Decentralized Processing Complexities

system-level changes. Management must ensure that sufficient resources and personnel are provided for the Configuration Control Authority to manage distributed system-level changes. Additionally,

1. Site-level CM Plans should be hierarchically subordinate to distributed system-level CM Plans.
2. All changes at the site level need to be reviewed by a site Configuration Control Authority for potential impact at the distributed system level.
3. The Distributed System CM Plan should describe the distribution controls and audit checks that are used to ensure that the common data and applications are the same version across the decentralized system.

For distributed systems where the managers of components do not report to (are not managed by) the same organization, the Configuration Control Authority needs to enter into a more formal agreement with each of the managers. A memorandum of agreement should be generated that establishes policies, standards and procedures, roles, responsibilities, and requirements for the total system. At a minimum a memorandum of agreement must identify, document, and provide a detailed description of the information to be provided from each component and the recipient of that information. It must also provide a description of each level of sensitivity or criticality for each data item, delineating the levels of sensitivity or criticality at which the data will be used, and the process for moving each data item to each operation level.

All memoranda of agreement should include a description by component and interface, of all security countermeasures required of each component. This description should focus on:

1. Security countermeasures to ensure confidentiality, integrity, and availability during the transfer of data and applications software.
2. Access control countermeasures to ensure that the transfer process is not used to gain unauthorized access to each component.
3. Countermeasures to ensure that the transferred data and applications are received only by the intended receiver (for data and applications requiring a high level of confidentiality).
4. A description of the overall distributed system security policy.

It is essential to include a detailed description of the transfer process between each component, identifying:

1. A description of any physical and media controls to be used.
2. Electronic transfers (bulletin board systems, communications software not integrated with the decentralized component) must include a description of the software used.
3. The software communications protocol and standards used.
4. Encryption methods and devices used.

5. The security features and limitations of the communications application used.
6. All hardware requirements, hardware settings, and protocols used.
7. Assignment of all decentralized system-level responsibilities and authorities, including network management, performance monitoring and tuning, training, training plan development and management, resource configuration management, software and data configuration management, system access control and audit management.
8. A description of all required components or site-level security roles and responsibilities, including resource, software, and data configuration management; access control; site security management; security awareness training and training management; as well as verification and validation of security relevant issues and audit control management.
9. An identification and needs assessment of the user community, including the levels of sensitivity or functional criticality of the information expected to be created, maintained, accessed, shared, or disseminated in or by the decentralized system.
10. A description of the information required in each component's audit trail and how the audit trail tasks will be divided among the components.
11. Any results of risk assessments and how controls mitigate perceived risks.

For distributed systems managed under a single organization, the Distributed System CM Plan must identify, define, and substantiate distributed system-level policies, standards and procedures, roles, responsibilities, and requirements for the interchange of data, as well as for configuration management at the distributed system-level in accordance with corporate Configuration Management guidelines.

Systems should segregate data and applications according to their organizational and/or functional sensitivity or criticality levels. Transitions between levels should be explicitly controlled. The process for transitioning data or applications from one sensitivity level to another, as well as from office systems and or end-user systems to other systems, must be formally documented and well understood. The transition process must include measures to increase the integrity and reliability of data and/or applications moving from less stringent requirements. Data must not be transitioned from a higher sensitivity level to a lower level that provides insufficient sensitivity protection. Additional application software may need to be developed to remove sensitive data when those data are transitioned to a level that cannot provide adequate protection. Application software must increase and ensure the integrity and reliability required when transitioning data from a component of lower reliability and integrity. A

formal process of transformation, testing, and certification must be developed for each transition.

For systems requiring a high level of integrity, techniques such as digital signature or digital envelope may be used to ensure that the data are not changed in transit. The digital envelope technique will provide a means for implementing the principle of least privilege or need-to-know concept.

Dispersed Distributed Systems Integrity Control Issues and Concerns

The following suggestions are provided as additional guidance for establishing a baseline set of controls that ensure minimal risk accountability encountered in managing the more complex environments of dispersed and/or interoperable systems. Additional controls for dispersed and/or interoperable systems will need to be developed addressing:

- Multisystem configuration management.
- Establishing and maintaining connectivity.
- Multilevel, multisite information transfers.
- Contingency planning, backup, and recovery.
- Maintaining multisystem data and referential integrity.
- Attaining a graceful degradation capability.
- Hardware maintenance.

Change control should be applied to dispersed or interoperable system level data, applications, and hardware to reduce the vulnerability to integrity loss. Periodic verification should be performed to ensure that the common data and applications are the correct version. Techniques (such as digital signature) may be used to assure applications and common data are at their expected version levels.

The functional equivalence claimed between two different software applications executing on different platforms will need to be closely examined during the procurement process due to the possibility of nonhomogeneous hardware being used in the dispersed system.

Network management personnel must maintain connectivity by allowing only authorized, authenticated users to log on, responding to access violation alarms, and auditing access logs for evidence of unauthorized access attempts.

Systems requiring the highest levels of availability must use error correction software during transmissions and redundant transmission of data down multiple communications paths to ensure that at least one is received. Transmission along multiple paths may be simultaneous, as in a broadcast mode, or may be an automatic response to failure detection or performance degradation beyond a predetermined threshold. An automatic response can be implemented to protect specific transmission lines,

or it can be implemented as an overall network scheme for automatic reconfiguration to optimize data transfer. The multiple path approach makes denial of service more difficult and reduces the possibility of a single point of failure.

Dispersed/interoperable systems must be supported by an onsite backup and restore repository for archiving applications and data. Backup procedures should be posted and training given to ensure backup integrity of data. Additionally, backup procedures should be automated to the greatest extent possible. A system of periodic and requested backups should be developed and enforced based upon the functional criticality of the system with respect to availability, accessibility, operational continuity, and responsiveness of recoverability needs. The more dynamic the critical data, the more frequently backups should occur. Intelligent backup systems, which back up only changed data, must have their configuration periodically certified for use.

Contingency planning for dispersed and/or interoperable systems must exist for those failures which are inevitable and those which may be unlikely but may result in catastrophic consequences. Contingency Planning should concentrate on the ability to configure, control and audit, operate, and maintain the data processing equipment to achieve information integrity, availability, and confidentiality. Specifically:

1. Upon failure, critical components should be replaced, repaired, and restarted according to contingency planning procedures.
2. Referential integrity of the data will need to be preserved. In systems where several processes may manipulate a data object, state data must be maintained about the data object so that incorrect sequencing may be prevented.
3. Each component must be capable of executing a controlled shutdown without impacting unrelated functions in other components in the event of a security breach or failure.
4. The dispersed system topology should be designed so that when hardware is taken out of service for maintenance, impact on the rest of the system is minimized.

Cooperative Distributed Systems Integrity Control Issues and Concerns

Additional controls for fully cooperative systems will need to be developed focusing on:

- Establishing and maintaining connectivity.
- Multilevel, multisite information transfers.
- Software development and maintenance.
- Hardware maintenance.

System management will need to conduct an impact analysis to determine the affect of monitoring all transactions involving data, process, and control information without causing degradation of the work in progress.

When transferring data between platforms, the classification access and the identity and authorization of the requester, the accredited classification range of the destination system, and destination level within that system should be authenticated. It is important to document any risks that have been accepted when classifying the level upon which a platform may process. This allows platforms under different management control to be evaluated for risks and have them taken into consideration when making reconfiguration plans. The transfer process must ensure that if the information fails to reach its destination the information is protected at the level required and appropriate warnings are raised.

A process will need to be implemented for introducing new platforms to an existing network. Cooperative processes will need to describe how the access control, security features, and auditability must be ensured prior to operational use of the new platform and how access will be granted. In a cooperative system with diverse platforms, a risk analysis will need to be performed to ensure that the combination of network operating system(s), platform operating system(s), and security software features available on each platform meet the access control and security requirements for that platform's assigned role in the network/system. In cooperative systems, the differences in security software present on or available for each platform must be reconciled to ensure the consistent deployment of the system of controls. The results of this risk analysis must be used when developing reconfiguration and/or recovery options.

A risk assessment of security requirements must be a product of each formal review (i.e., system specification review, preliminary design review, critical design review, etc.) during the software development life cycle. In systems where several processes may manipulate a data object, state data must be maintained about the data object so that incorrect sequencing may be prevented and processing completion can be determined.

Software targeted for use in cooperative systems must be designed using the principle of loose coupling and high cohesion. Loose coupling indicates weak software module-to-module dependency. High cohesion indicates that a module performs a discrete function. In concert, the properties of loose coupling and high cohesion indicate a software module designed for independent performance. Using this principle produces software modules that can execute alone and enable the production of software which may degrade gracefully. Software targeted for use in cooperative systems must be designed so that each component is network topology independent. This will enable components to more readily be installed or reconfigured onto any platform within the network.

Components of cooperative systems must be designed to allow the removal of components to perform maintenance, testing, etc. with minimal impact to operations. Before an element can be removed from the cooperative network, the component must conclude all pending transactions. The work being performed by that component will need to either be done on another platform or the system must continue in a degraded state. Cooperative systems need to be designed with an operational capability for placing the components in a quiescent state. This operation must:

- Cause a component to notify all other components in the system that it is about to terminate.
- Cause all other components in the system to respond by ceasing any transmissions to that component.
- Cause the component to conclude all pending transactions.
- Cause the component to post notification that it is now quiescent.

An operational capability must also exist that allows the component to reenter the network in diagnostic mode for checkout and to notify other components that the component/platform is back in the network but not ready for operational use. Additionally an operational capability will need to exist to allow the component to reenter the system as active from the diagnostic mode and to notify other components that the component is active and fully functional.

INTEROPERABLE RISK ACCOUNTABILITY CONCEPTS

In designing and developing high-integrity interoperable systems, management is faced with the issue that connectivity is still a point-to-point transmission irregardless of the transmission mechanism itself. Unfortunately in today's infrastructure, the majority of attention is focused on adding layers of protection, rather than building controls into the application systems at either end of the transmission. Even with advances in firewall technology, authentication processes, and encryption, management must address the issues of intrusion and infiltration into, as well as exploitation of their information resources by an increasing number of external threat manifestations.

Management must address the following key issues about risk, mitigation of risk, residual risk acceptance, and exercising a standard of due care in protecting its information resources. Additionally, management must recognize that an integrated intrusion detection process and penetration testing are integral components of today's system life cycle. Penetration testing offers the only suite of tests that reflect "real-world" scenarios; and must be integrated into the verification and validation of a system's productional acceptance criteria throughout all life-cycle phases. Intrusion detection, on the other hand, must be instantiated into the overall operational control, similar to, or as a part of the access control and audit.

Risk Accountability Associated with Developing, Maintaining, and Protecting Information Resources

Information security is still largely an unknown entity to most people. Managers can and often do ignore advice offered by security professionals. In the past, when the integrity, availability, or confidentiality of information systems was breached and damages occurred, the majority of damages were internal and simply absorbed by the organization. Limited incident investigation was performed. With the advent of virus infections and the susceptibility of interoperable, intra/Internetworked systems, management must take a proactive approach to managing and protecting its information resources.

Any organization and/or individual is liable when they act in a way that they should not have, or fail to act the way they should, and this act or failure results in harm that could have been prevented. Therefore, it is exceedingly important for management to fully understand the limits of liability associated with managing and protecting corporate information resources and which method of security management to implement.

Compliance-Based Security Management

The compliance-based approach has been an accepted method of protecting information resources. It yields clear requirements that are easy to audit. However, a compliance-based approach to information security does have notable disadvantages when applied to both classified or unclassified information systems.

A compliance-based approach treats every system the same, protecting all systems against the same threats, whether they exist or not. It also eliminates flexibility on the part of a manager who controls and processes the information and who makes reasonable decisions about accepting risks. Utilization of a compliance-based approach may often leave the owners of the information systems with a false impression that a one-time answer to security makes the system secure forever. Usually, the inflexibility of a compliance-based approach significantly increases the cost of the security program, while failing to provide a higher level or more secure information systems.

Risk-Based Security Management

Management often confuses Risk Management with Risk-Based Management. Risk Management is an analytical decision-making process used to address the identification, implementation, and administration of actions and responses, based upon the propensity for an event to occur that would have a negative effect upon an organization or its functional programs or components. Risk Management address probabilistic threats (e.g., natural disasters, human errors, accidents, technology failures, etc.), but fails to

take into account speculative risks (e.g., legal or regulatory changes, economic change, social change, political change, technological change, or management and organizational strategies). In contrast, Risk-Based Management is a methodology that involves the frequent assessment of events (both probabilistic and speculative) affecting an environment.

In managing the security of information systems, a risk-based approach is essentially an integrity failure impact assessment of the environment, program, system, and subsystem components. As such, it must be integrated as a part of the system life cycle. A risk-based approach to security directly places the responsibility for determining the actual threats to a processing environment and for determining how much risk to accept, in the hands of the managers who are most familiar with the environment in which they have to operate.

Both compliance-based security management and risk-based security management take advantage of risk management processes and assessment practices. In contrast to the compliance-based security management discussed above, using a risk-based security management approach allows managers to make decisions based on identified risks rather than on a comprehensive list of risks, many of which may not even exist for the facility in question. Security control requirements for each information system may then be determined throughout the system's life cycle by iterative risk management processes and summarized as a control architecture under configuration management. Implementation of a security control architecture as a primary point of control ensures that each information system is protected in accordance with organizational policy, and at the levels of integrity, availability, and confidentiality appropriate for the functions of the corporation's systems.

Exercising Due Care

A standard of due care is the minimum and customary practice of responsible protection of assets that reflects a community or societal norm. In the private sector this norm is usually based on type or line of business (e.g., banking, insurance, oil and gas, medical, etc.), and within the public sector this norm is determined by legislative, federal, and agency requirements. Efforts to develop a universal norm for both the public and private sectors as well as for the international community have been initiated in response to the National Information Infrastructure and the development of the international Common Criteria.

In either sector, failure to achieve minimum standards would be considered negligent and could lead to litigation, higher insurance rates, and loss of assets. Sufficient care of assets should be maintained such that recognized experts in the field would agree that negligence of care is not apparent.

Due care must be exercised to ensure that the type of control, the cost of control, and the deployment of control are appropriate for the system being managed. Due care implies reasonable care and competence, not infallibility or extraordinary performance, providing assurance that management does not overcontrol nor take an unnecessary reactionary, politically motivated, or emotional position.

Due diligence, on the other hand, is simply the prudent management and execution of due care. Failure to achieve the minimum standards would be considered negligent and could lead to loss of assets, life, and/or litigation.

Understanding the Accountability Associated with Exercising a Standard of Due Care

Although significant strides have been made in criminal prosecution of computer and "high tech" crime in the last few years, the civil concepts (contractual and common law) of negligence and exercising a standard of due care for the protection of information of inter/intranetworked systems and the National Information Infrastructure are still in their embryonic state.

Under the standard of Due Care, managers and their organizations have a duty to provide for information security even though they may not be aware they have such obligations. These obligations arise from the portion of U.S. Common Law that deals with issues of negligence.

Since information systems are relied on by a rapidly increasing number of people outside the organizations providing the services, the lives, livelihood, property, and privacy of more and more individuals may be affected. As a result, an increasing number of users and third-party nonusers are being exposed to and are now actually experiencing damages as a result of failures of information security in information systems. If managers take actions that leave their information resources unreasonably insecure, or if they fail to take actions to make their information resources reasonably secure, and as a result someone suffers damages when those systems are penetrated, usurped, or otherwise corrupted, both the managers and their organizations may be sued for negligence.

Integrity Issues and Associated Policy Concerns

1. Duties and responsibilities must be defined so that security controls are established to ensure separation of logical and physical environments (i.e., maintenance, test, production, quality assurance, and configuration management) for each distributed system node and the interaction between nodes. Policies must also address the various resources, skills, and information requirements that exist for consistent deployment of controls supporting the management and

maintenance of the distributed systems facilities. Additional policies may need to be developed based on the characteristics of a specific distributed system node after the software and hardware for that node have been selected for implementation.

2. Organizational functions and individual duties must be separated. Separation of functions and duties along organizational lines will complicate circumvention of security controls in the acquisition, implementation, and operation of the software at each distributed node or in defining the permissibility of actions between nodes.
3. Configuration Management (CM) plans will need to be developed at the system level, or at a minimum redesigned to include the following:
   - Distributed system CM plans must document system-level and site-level policies, standards, procedures, responsibilities, and requirements for the overall system control of the exchange of data.
   - Distributed system CM plans must document the identification of each individual site's configuration.
   - Distributed system CM plans must include documentation for common data, hardware, and software.
   - Maintenance of each component's configuration must be identified in the CM plan.

   A system-level CM plan is needed that will describe distribution controls and audit checks to ensure common data and application versions are the same across the distributed system in which site-level CM plans are subordinate to distributed-level CM plans. For distributed-level changes, if the components are not documented in a single CM plan, a change control authority will need to be established as a point of control. In distributed systems where nodes are geographically separated or when the components are not documented in a single CM plan, site-level changes must be reviewed by a site's change control authority for potential impacts at the distributed level. Additionally, the change control authority(s) will need to establish agreements with all distributed systems on policies, standards, procedures, roles, responsibilities, and requirements for distributed systems that are not managed by a single organizational department, agency, or entity.

4. If digital signatures are used for configuration management of critical software components; then the digital signature technology must validate the configuration of each node during system validation tests. It is imperative that the signature construct be formulated during node certification.
5. Security control requirements and responsibilities will need to be identified that focus on establishing procedures for owners, users, and custodians of distributed systems hardware and software; as well as procedures for the overall system and for each node to

ensure consistent implementation of security controls for handling data between components of distributed systems.

6. Organizational and functional access controls must be implemented for each node identifying and establishing the relationship between node software and hardware resources, and that periodic assessment of the relationship between node software and hardware resources be performed to ensure that access is limited to a definite minimum.
7. Security controls need to be assessed, by node, at each phase review of the system development life cycle to ensure that as requirements and vulnerabilities are discovered, they are addressed using the design/implementation approach. Additionally, independent testing and verification responsibilities should be assigned, by node, for maintenance and production processes to ensure that safeguards and protection mechanisms are not compromised by special interests.
8. Since distributed systems require network connection for communication with other nodes, network security controls must be considered which address:
   - User authentication
   - Data flow disguise
   - Traffic authentication
   - System attack detection
   - Repudiation protection
9. The level of physical access control depends on the functional criticality or sensitivity level of the information being processed, proprietary process(es) invoked, and/or software/hardware employed. Distributed system components that normally need to be guarded include:
   - Terminals
   - Equipment
   - Nodes
   - Communication lines
   - Connections
10. Intrusion detection processes and mechanisms will need to be deployed to detect, monitor, and control both internal and external intrusion and/or infiltration attempts. Additionally, corresponding controls will need to be established to address all security incidents. A security incident is considered to be an event that is judged unusual enough to warrant investigation to determine if a threat manifestation or vulnerability exploitation has occurred. For distributed systems, security incident detection requires the reporting of and warning to other nodes of the system that such an event has occurred within the control domain.

11. A capability will need to be provided to evaluate the effectiveness of security controls. In order to evaluate the effectiveness, security controls must be modular and measurable.
12. Software with privileged instruction sets that can override security controls within the system must be identified, certified, and controlled.
13. Designers will need to reconcile the differences in security software installed or available on each platform.
14. Designers must be able to ensure a consistent implementation of security controls.
15. Communications subsystem packages for each node must be capable of logging the status of information transfer attempts. Additionally, security management personnel must periodically review these data for evidence of attempts to gain unauthorized access or corrupt data integrity during the transfer process.
16. Distributed system managers will need to maintain connectivity capabilities by allowing only authorized, authenticated users to log on, responding to access violation alarms, and auditing access logs for attempts at unauthorized access.
17. Functions will need to be identified and separated into isolated security domains. These isolated security domains will ensure the confidentiality, integrity, and availability of information for the overall system and for each node. Management may decide that a security control architecture (the composite of all controls within the design of the system addressing security-related requirements) will need to be established that defines isolatable security domains within the environment to ensure integrity within each domain, as well as between levels of sensitivity and domain boundaries.
18. System reconfiguration plans will need to be developed. Additionally, procedures must be established for introducing new platforms to existing distributed systems. These procedures must describe how access controls, security features, and audit capabilities will be implemented before operational use, and how access will be granted gradually as controls are assured. In distributed systems with diverse platforms, a risk analysis will need to be performed to ensure that the combination of network operating system, platform operating system, and security software features on each platform meet security requirements for their roles in the system. The analysis is necessary to identify and develop reconfiguration and recovery options.
19. Distributed system components must be capable of executing a controlled shutdown without impacting unrelated functions in other components. The mode (automated or manual) to perform a controlled shutdown should be based on predefined, documented criteria to ensure consistency and continuity of operations.

20. System management will need to conduct impact assessment to discover, for each node and for the network as a whole, factors that may affect the system connectivity, including:
    - The type of information traveling from node to node.
    - The levels of sensitivity or classification of each node and of the network.
    - The node and network security countermeasures in place.
    - The overall distributed system security policy.
    - The method of information transfer between nodes and the controls implemented.
    - The audit trails being created by each node and the network.

THE SYSTEMS INTEGRITY ENGINEERING METHODOLOGY

From the previous discussions on understanding the control issues and concerns associated with fully distributed and/or dispersed interoperable systems, it is clearly evident that management must take a proactive approach to designing, developing, and securing its information resources. In order to address this dynamic environment in which the system development life cycle has been shortened from weeks and months to hours and days (e.g., LINUX development), management is faced with making real-time decisions with limited information and assurances.

The model used in the development of this methodology is a highly complex global, multicorporate, multiplatform, intra- and Internetworked environment that substantiates the need for a synergistic business approach for bridging the gaps between the four key area product development support functions: system design and development, configuration management, information security, and quality assurance. These systems encompass:

- Some 3,600 personnel,
- About 1,682 large mainframes, minis, and dispersed cooperative systems,
- Five types of operating systems,
- A variety of network and communication protocols, and
- Varying geographical locations.

This approach forms an enterprise-wide discipline needed for assuring the integrity, reliability, and continuity of secure information products and services. Although the development and maintenance concepts for high-integrity systems are specifically addressed, the processes described are equally applicable to all systems, regardless of size or complexity.

Information Systems Integrity Program

Change is not easy whenever an enterprise considers reengineering its business processes. This kind of competitive business initiative typically

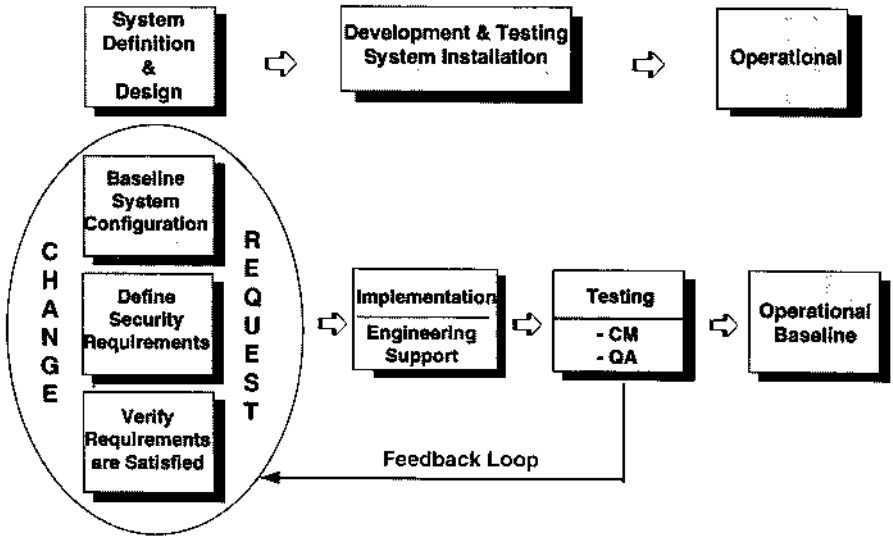


Exhibit 6. Change Process

involves redesigning and retooling value-added systems for new economies. Many of these are legacy systems which are being pulled along by new technology, making change very difficult to manage. The speed at which new emerging information technology is introduced to market has also made it difficult to maintain an information systems control architecture baseline. Continued budget constraints have become a recognized element in managing this change.

Systems Integrity Engineering Process

In today's computing world, distributed processing technologies and resources change faster than most operational platforms can be baselined. As they evolve with an ever-increasing speed, organizations are challenged with an opportunity to maintain stability for growth and strategic competitiveness. Management must consider that sensitive business systems increasingly demand higher levels of integrity in system and data availability. Within this framework, reliability, through product assurance and security assurance constructs, provides a common enterprise objective. Accordingly, the scope of an enterprise-wide product assurance partnership and management-friendly metrics must be expanded to all four functional areas as a single, logical, integrated entity with fully matrixed management (i.e., both horizontal and vertical management control). The process in which requirements for new information technology are infused into the enterprise and managed becomes the pivotal business success

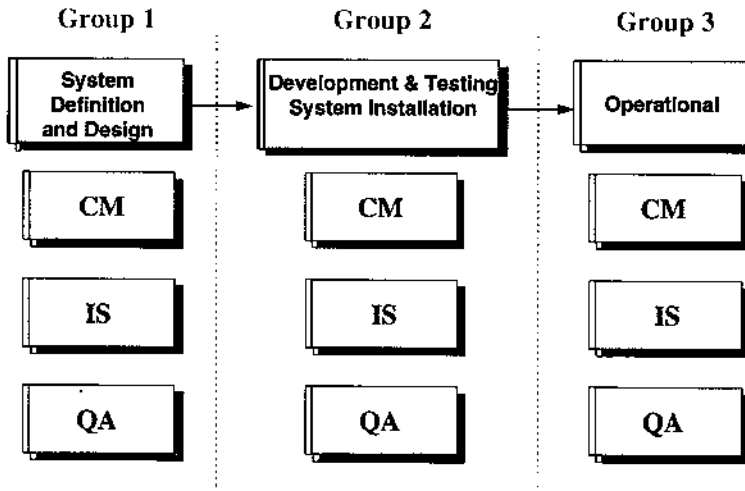


Exhibit 7. Interdependencies of Change

factor that must be defined, disseminated, and understood by the key functional support organizations.

New Alliance Partnership Model (NAPM)

In their presentation to the 18th National Information Systems Security Conference (October, 1995) on "The New Alliance: Gaining on Security Integrity Assurance", Sanchez and Evans described a new alliance partnership model developed from a four-year case study in which security, configuration management, and quality assurance functions were combined with an overall automated information systems (AIS) security engineering process. In this paper, Sanchez and Evans delineated the following.

It has become critically essential for enterprise management to understand the interdependencies and complementary pursuits that exist between the Information Systems Design and Development, the Quality Assurance (QA), Configuration Management (CM), and the Information Systems Security (IS) organizational support functions. With this knowledge, it is equally important to identify and examine a synergistic approach for realizing additional economies (cost savings/avoidances) throughout the system development life cycle with continuous improvement techniques.

Implementation of product assurance and secure information technology development is a management decision that must be judiciously exercised and integrated as part of a system control architecture. In this model, automated information systems security management is recognized as the functional point of control and authority for coordinating and guiding the

development, implementation, maintenance, and proceduralization of information security into a unique, integrated management team. The use of a security control architecture is the approved strategic methodology used to produce a composite system of security controls, requirements, and safeguards planned or implemented within an IS environment to ensure the integrity, availability, and confidentiality. This is the only approach that will allow for integration and cooperative input from the CM, AIS security engineering, and QA management groups. Each of these product assurance functional support groups must understand and embrace common corporate product assurance objectives, synergize resources, and emerge as a partnership free of corporate political strife dedicated to providing a harmonization of systems integrity, availability, and confidentiality.

The harmonization effort evolves as an enterprise-wide New Alliance Partnership Model (NAPM) in which:

- QA provides an enhanced product assurance visibility by ensuring that the intended features and requirements, including but not limited to security, are present in the delivered software. QA allows program management and the customer to follow the evolution of a capability from request through requirement and design, to a fielded product. This provides management with an enhanced capability as well as a forum for identifying and minimizing misinterpretations and omissions which may lead to vulnerabilities in a delivered system. The formal specifications required by QA increase the chance that the desired capabilities will be developed. The formal documentation of corrective actions from reviews (of specifications, designs, etc.) lessens the chance that critical issues may go undetected.
- CM provides management with the assurance that changes to an existing AIS are performed in an identifiable and controlled environment and that these changes do not adversely affect the integrity or availability properties of secure products, systems, and services. CM provides additional security assurance levels in that all additions, deletions, or changes made to a system do not compromise its integrity, availability, or confidentiality. CM is achieved through proceduralization and unbiased verification ensuring that changes to an AIS and/or all supporting documentation are updated properly, concentrating on four components: identification, change control, status accounting, and auditing.
- IS provides additional controls and protection mechanisms based upon system specifications, confidentiality objectives, legislative requirements and mandates, or perceived levels of protection. AIS security primarily addresses the concerns associated with unauthorized access to, disclosure, modification, or destruction of sensitive or

proprietary information, and denial of IT service. AIS security may be built into, or added onto, existing IT or developed IT products, systems, and services.

- Organizational management provides the empowerment and guidance for the economies of scale.

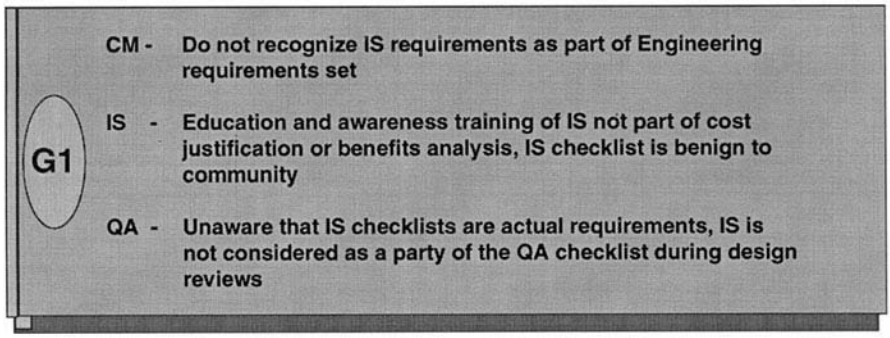


Exhibit 8. System Definition and Design Constraints

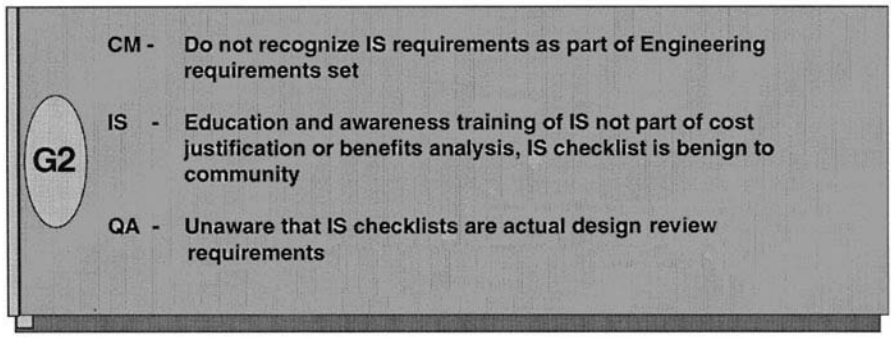


Exhibit 9. Development, Testing, and Installation Constraints

A seminal case study was presented as proof of the concept for gaining security integrity assurance. It identified the interdependencies and synergy that exist between the CM, IS security engineering, and QA functional management activities. It describes how information technology, as a principle change driver, is forcing the need for a QA, CM, and AIS security forum to evolve if the enterprise is to be successful in providing high-integrity systems.

Sanchez and Evans were able to provide the following:

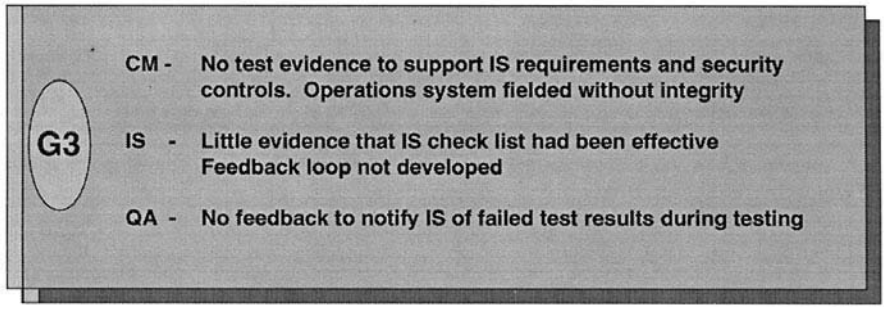


Exhibit 10. Operational Constraints

1. Change is not easy. Change has not been easy. Change will not be easy. In this case study, the members of each respective management support team have championed the process improvement initiatives and the corrective actions taken thus far. It is important to emphasize that employee empowerment of this type must be supported by top management because security integrity engineering and the implementation of an integrated product assurance and secure information technology development process such as a control architecture is a proactive management decision.
2. Information technology has been and will continue to be a major change driver that establishes a need for a functional organizational support forum dedicated to delivering high-integrity products and services. Each of the product assurance functional support organizations must understand and embrace common corporate product assurance objectives, synergize resources, and emerge as a partnership independent of corporate political strife and dedicated to harmonizing systems integrity, availability, and confidentiality.
3. The New Alliance Partnership Model (NAPM) is a viable solution that has been put to the test and proven in a highly dynamic operational environment of ever-changing distributed processing technologies. The NAPM supports the integration process and requires that direct lines of communication be bridged between key functional support organizations so as to input and feedback closure information.

Incorporating NAPM into the System Development Life Cycle

In order to fully integrate the partnership model into a System Integrity Engineering discipline it is imperative that the designers and system architects understand and embrace the requirements imposed by technology infusion and the insatiable demand for more interoperable processing capabilities and applications.

Management can no longer afford to "bury its head in the sand" and ignore threats simply because there is (1) no commercially available hardware and/or software solution(s) available; or (2) prohibitive budgetary restraints make addressing the issues improbable. The threats will not magically disappear. They must be openly and intelligently addressed. Application design or enhancements may no longer be the sole major driving force in today's interoperable development environment. Management is beginning to be more interested in systems that provide them with a high degree of confidence in protecting their information, consistency, and continuity of operation, as well as efficiency and computational effectivity.

The basic System Development Life Cycle has changed dramatically. Design and development efforts that once took months, even years, has been replaced by rapid application and joint analysis development (RAD/JAD) processes, prototyping, reuse engineering, and fourth-generation languages. These have modified the timing cycle by drastically shortening it to days and weeks, or in some cases hours and minutes.

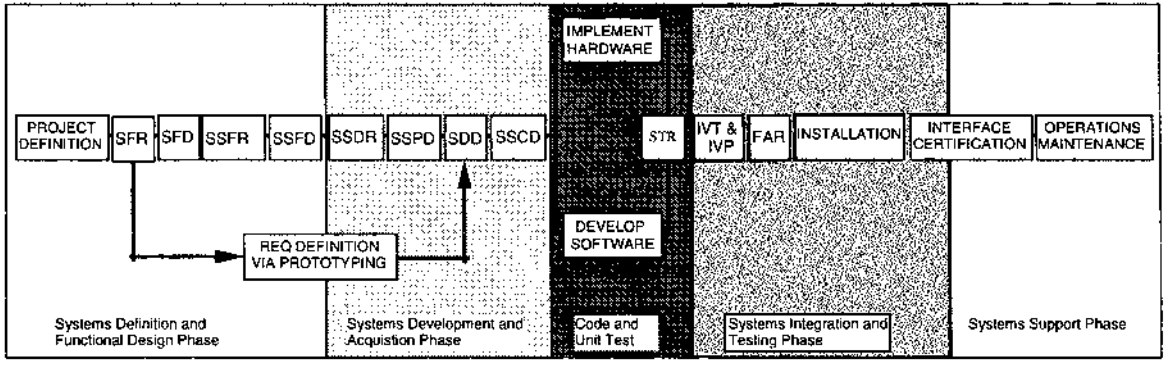
To effectively integrate a system of controls into the life cycle, designers and developers will need to consider a modified model that recognizes that in an iterative system development life cycle, security controls and protection mechanisms need to be addressed in an iterative manner as well.

Software Life Cycle as a Control Process. The basic life cycle is still comprised of a series of phases to be executed sequentially or recursively as a continual process. A set of software products to be produced during each phase is identified, including security-related analyses, documentation, and reports. The controls deployed as well as those planned during each of the life cycle phases comprises a unique control architecture for the developing software products.

It is imperative that all relevant products are developed, all reviews are held, and all follow-up actions performed within each of the life cycle phases in sequence. To provide adequate management control, it is normally necessary that the developer not be allowed to proceed unless the defined phases of development are approved, performed in their predefined order, and the developer receives authority to proceed. The controls governing the applicability of a life cycle model to development and maintenance projects must be identified, evaluated, and specified with the consideration of integrity and security-relevant controls deployment criteria.

Each of the following development life cycle approaches provides inherent integrity controls:

- The classical software development method recognizes discrete phases of development and requires that each phase of development



FAR	Final Acceptance Review	SSDR	Subsystem Detailed Requirements (Level C)
IVP	Integrity Verification Process	SSFD	Subsystem Functional Design
IVT	Independent Verification & Test	SSFR	Subsystem Fucntional Requirements (Level B)
SDD	System Detailed Design	SSPD	Subsystem Preliminary Design
SFD	System Functional Requirements (Level A)	STR	Start of Testing Review
SSCD	Subsystem Critical Design		

Exhibit 11. Example of a System Life Cycle

be complete, with the presentation of formal reviews and release of formal documentation prior to transitioning to the next phase.
- Spiral development is an iterative approach toward the classical method where the development life cycle is restarted to enable the rolling in of lessons learned into the earlier development phases.
- Rapid application development (RAD) is a method of rapidly fielding experimental and noncritical systems in order to determine user requirements or satisfy immediate needs.
- Joint analysis development (JAD) is a workshop-oriented, case-assisted method for application development within a short time frame using a small team of expert users, expert systems, expert developers, and outside technical experts, a project manager, executive sponsor, a JAD/CASE specialist, and observers.
- Cleanroom is a method for developing high-quality software with certifiable reliability. Cleanroom software development attempts to prevent errors from entering the development process at all phases. The process provides for specifiers, programmers, and testers in which a specification is prepared either formally or semiformally as notations. Programmers prepare software from the specifications. A separate team prepares tests that duplicate the statistical distribution of operational use. Programmers are not permitted to conduct tests; all testing is done by an independent test team.

Regardless of method, formal reviews and audits need to be performed to provide management and user insight into the developing system. Through the use of the review process, potential problems may be readily identified and addressed. Technical interchange meetings and peer reviews, involving technical personnel only, should be used to promote communication within the development organization and with the user community, enable the rapid identification and clarification of requirements, reduce risk, and promote the development of quality products.

Modified Interoperable Software Development Life Cycle Process. The software development life cycle (see Exhibits 12 and 13) for dispersed and distributed interoperable systems requires that prototyping be done which redefines the requirements definition, provides early identification of interfaces, and shortens the hardware and software development and acceptance phases of the life cycle when combined with real-time testing and anomaly resolution. In order to assure that appropriate controls deployments are considered and incorporated, system designers and developers will need to consider a slightly modified approach in which security-relevant safeguards and protection mechanisms are managed.

Management must be able to identify a protection strategy that addresses threat manifestations before, after, and during their occurrence(s) as a qualitative "relative timing factor" rather than as a calculated

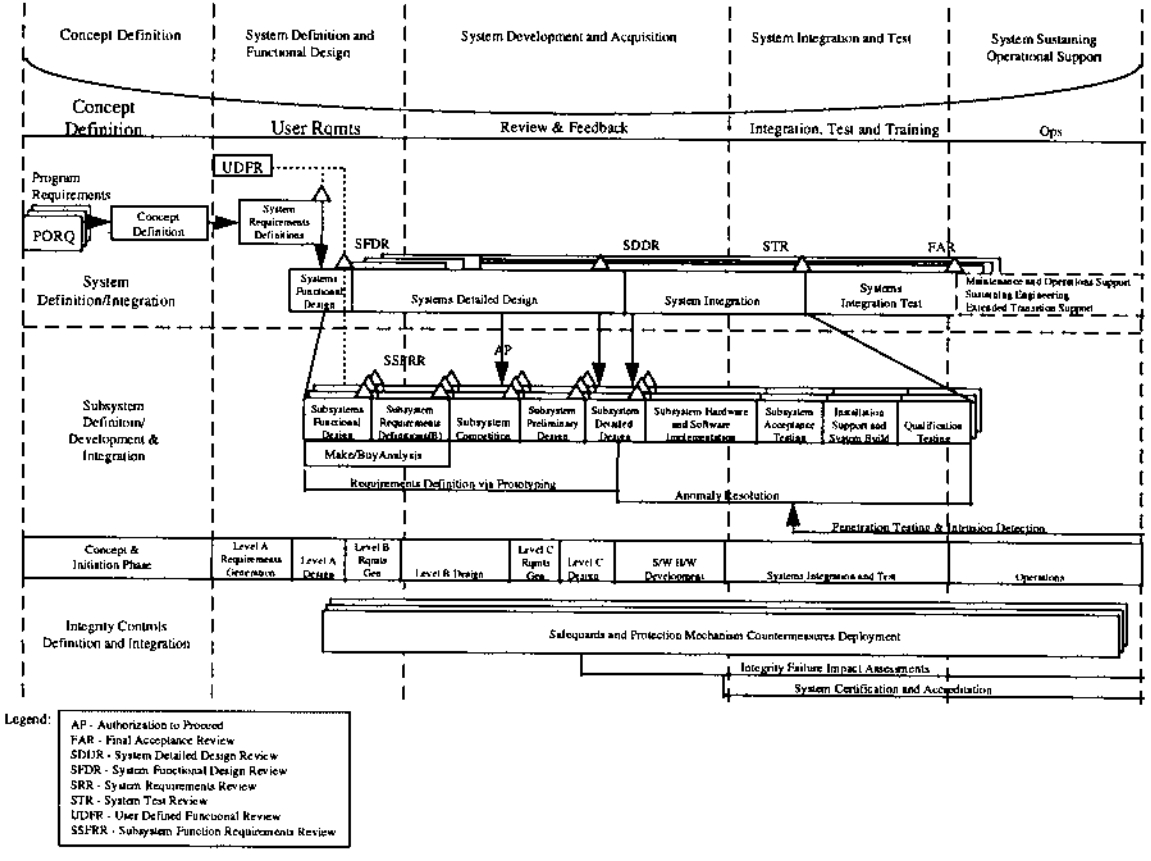


Exhibit 12. Modified System Development Life Cycle

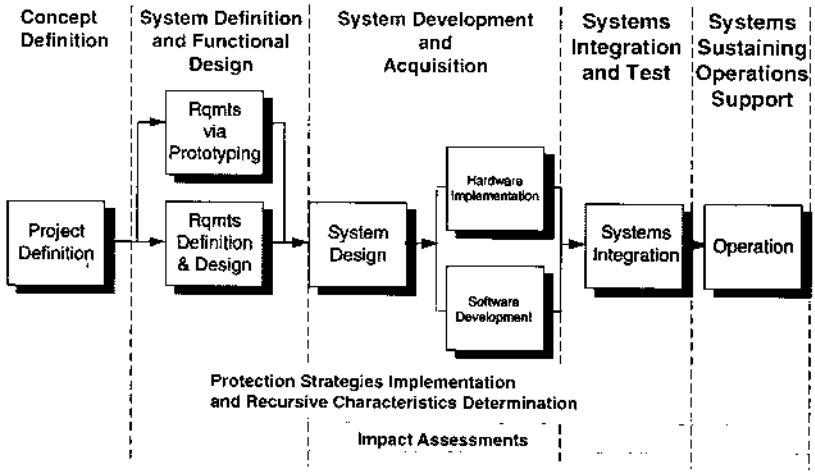


Exhibit 13. System Development Life Cycle Protection Strategies Deployments

probability of occurrence or frequency, since interoperable systems have a high probability of being exploited. For most systems an attack(s) is a foregone conclusion and simply a matter of "when" rather than " what if" or "will" a threatening event occur.

In Exhibits 13 and 14, consideration is given to the types of controls and associated safeguards and protection mechanisms deployed as countermeasures to threats. Types of controls and safeguards are generally classified as detective, preventative, and recovery controls. Since these control types may have an associated protection strategy and occur in a recursive process throughout each phase of the life cycle, then each safeguard has a unique signature depending upon each of the three types of controls and protection strategy(s) employed, as well as individualized recursive characteristics.

In Exhibit 14, the recursive characteristics and uniqueness of signature are clearly evident. Regardless of the point of origin within the PDR iteration, there is an identification (real or perceived) and a detection (D) of an exposure or risk, an associated recovery (R) strategy, followed by a preventative mechanism (P) or strategy that is for all practical purposes independent of when the threat manifestation actually occurs.

If taken in a controlled environment, prevention is normally the first of the recursive steps since there are normally control deployments based upon perceived threats rather than actual manifestations. The uniqueness of the PDR signature (i.e., 1 + 2 + 3 + … n + n+1) is attributed to the combinations of subsequent activities and protection strategies introduced into

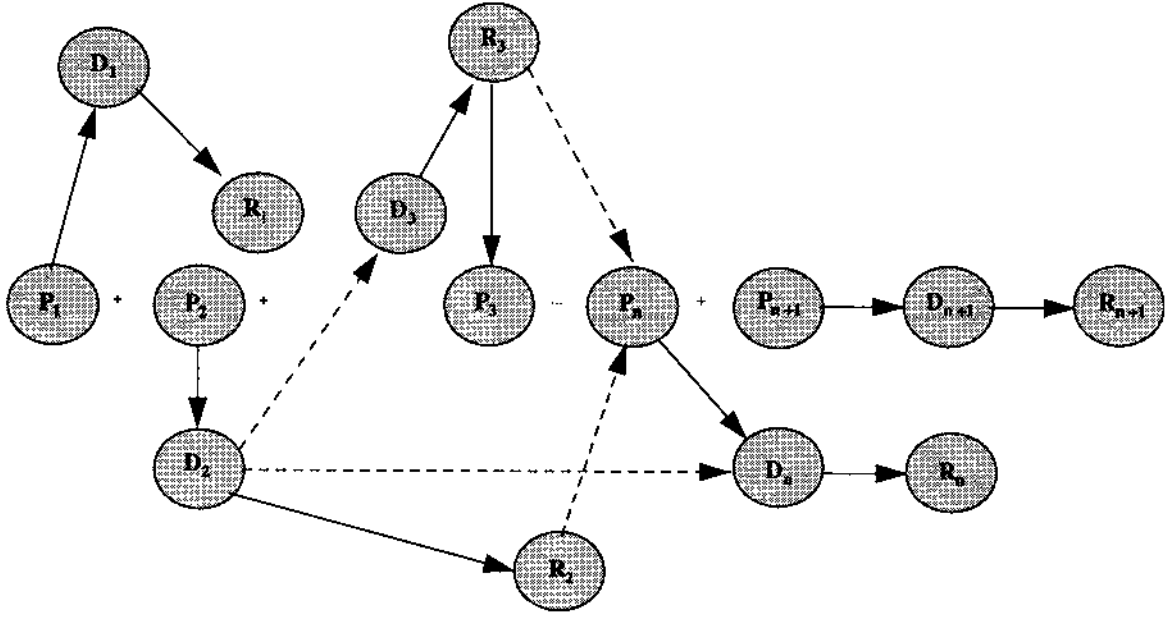


Exhibit 14. Recursive Characteristics of Protection Controls

each iteration of the process. The combination of all safeguards with respect to detection, prevention, or recovery, therefore, provides management with a process and a metric that is relatively independent of time for determining risk accountability and propensity of threat manifestation(s).

Stacey, Helsley, and Baston in their paper, "Risk-Based Management, How To: Identify Your Information Security Threats" arrived at a similar conclusion in determining threat events and their relationship to protection strategies.

They outline a structured approach for the identification of a threat population, correlating threat events and protection control strategies to security concerns. In determining when to protect a system from a threat event (before, during, or after the occurrence of a threat event), they arrived at the conclusion that once a threat event had been identified, one could assign a set of safeguards for each protection strategy (i.e., prevention, detection, and recovery) as an independent point of control.

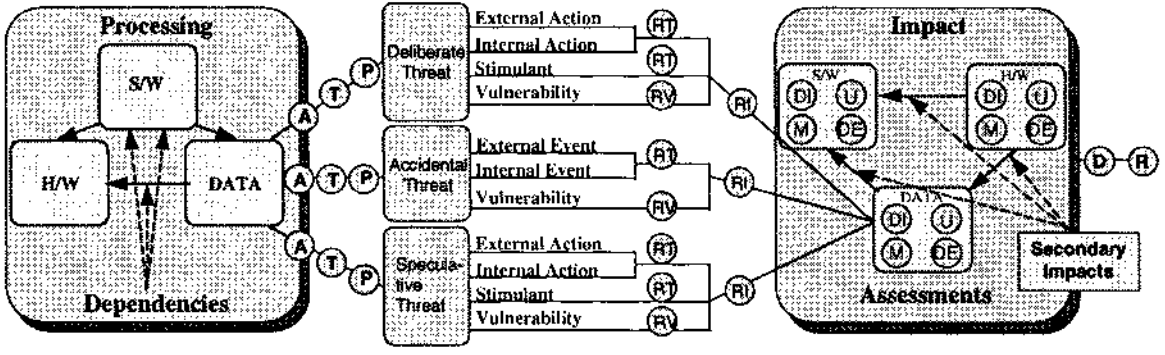
Integrity Failure Impact Assessments (IFIA)

System availability and robustness often erroneously preempt reliability and integrity concepts. In an interoperable environment comprised of a system(s), management's confidence in the integrity of the system (level of trustworthiness) is primarily based on whether the "system" is readily accessible for use and possesses the capability of being able to process information, rather than the integrity of what is produced, when it was produced, who used it (or was authorized to used it), or how was the information produced, protected, stored, transmitted, and/or disseminated.

In assessing the level of trustworthiness of a system, processing dependencies and types of controls, threat events, and impacts to its integrity, as well as the associated relationship to an enterprise's protection strategy (PDR) must be identified.

This relationship is best described as an Integrity Failure Impact Assessment (IFIA), in which deliberate and accidental threat events (including associated actions/reactions and vulnerabilities), primary and secondary impacts, processing dependencies, and protection strategies are evaluated, documented, and preserved as an enterprise-wide baseline supporting the corporate decision-making process. IFIA, which are similar in nature to reliability engineering determinations of mean-time between failure and mean-time to repair, will need to be developed based upon the enterprise's overall protection strategies.

Once IFIA have documented the frequency of occurrence and the mean-time to restore a system(s) to a known integrity state(s), management can qualitatively ascertain and maintain an acceptable level of confidence in its



Type of Control			Type of Impact
Preventative	Detective	Recovery	
(A) Avoid the Threat	(D) Detect	(R) Recover	(D) Disclosure
(T) Transfer the Threat			(M) Modification
(TR) Reduce the Threat			(U) Loss of Availability
(VR) Reduce the Vulnerability			(DE) Destruction
(RI) Reduce the Impact			
(P) Prevent			

Exhibit 15. Protection Strategies

high-integrity systems and processes based upon sound engineering concepts and practices.

MOTIVATIONAL BUSINESS VALUES AND ISSUES

The business values, issues, and management challenges that drive integrity initiatives and commitments are primarily comprised of, but are not limited to the following:

- The value of a surprise-free future.
- The value of system survivability and processing integrity.
- The value of information availability.
- The issue of the sensitivity and/or the programmatic criticality of information.
- The issue of trust.
- The issue of uncertainty.
- The issue of measurability of risk.
- The challenges in managing critical resources.
- The administrative challenge of controlling and safeguarding access to and usage of proprietary information.
- The challenge of technology infusion.

Value of a surprise-free future — If management is continually addressing unwelcome surprises, denials of services, and impacts to its processing objectives, the enterprise will experience (1) loss of credibility, (2) investment in less than optimum resource commitments and unnecessary expenditures, (3) and unproductive reactive management decisions. The optimum value is a surprise-free future which can be proactively managed. The ideal can and should be approached through substantiation of both strategic and tactical countermeasures and protection mechanisms that safeguard against those factors that contribute to the uncertainty of resources and assets These countermeasures cover a wide spectrum ranging from administrative manual procedures and processes to sophisticated engineering processes and tools that focus on disparate heteromorphic processing environments and the complexity of the domains, components, and subcomponents that comprise a corporation's overall processing program.

Value of system survivability and processing integrity — This is attained through the management of uncertainty surrounding the robustness of critical information processes and resources, their identification, quantification, assessment, and use. A system's robustness is a relational correlation of the system's components, to each component's "built in" resistance capability (including processing redundancy, logical self propagation, and accessibility to, and deployment of, additional sustaining countermeasures and protection mechanisms), to internal and external threats of misuse, abuse, espionage, or attack(s). In complex intra/Internetworked systems or systems of systems, the capability to maintain the referential

integrity of the information created, used, stored, and/or transmitted is imperative.

Value of information availability — This focuses on the demand, responsiveness, and accessibility of information resources, as needed, including preservation and recoverability following the manifestation of a disruption or denial of service.

Issue of sensitivity and/or programmatic functional criticality of information — This is determined by an enterprise-wide programmatic assessment of the values of information resources and operational performance(s). The valuation items and/or issues identified are used by management to determine the relevant consequences of both real and perceived loss of information integrity, availability, and confidentiality; and are assigned a weighting factor(s) as to their significance or perceived significance. These valuation items are imperative in determining appropriate strategic and tactical control deployments and justification of associated expenditures to meet business objectives.

Issue of trust — This is a determination resulting from the identification and assessment of where and/or how information resources are assembled, stored, and processed by human or electronic entities/agents/systems. Each process and/or associated agent normally has differing levels of privileges that may impact the integrity of the information resources. The use of trusted agents and systems to establish "webs of trust" for intra/Internetworked systems demands proactive management of uncertainty in using information resources, and is based upon the assumption that:

1. The trust level or the "need to know" and privileges of agents accessing and using information resources are assignable, verifiable, and controlled at all times.
2. Agents have certifiable skills for correctly operating interfaces to information resources.
3. The state and attributes of information environments, processing capabilities, and carriers are identifiable, accountable, and assignable at all times.
4. Systems in which uncertainties in these attributes exist have been (or are in the process of being) reduced to acceptable levels which may be independently verified.
5. Penetration testing procedures and processes will be implemented as a normal suite of tests to simulate real-world tests of the web of trust and to determine true protection limitations.

Issue of uncertainty — This is the motivational factor in which full certainty of information processing agents, systems, and information resources may not be practically achievable. Proactive minimization of

uncertainty demands accountability for risk acceptance. Acceptable levels of risk are measured in terms of those exposures that do not have corresponding safeguards to reduce or eliminate risk(s) due to weaknesses in existing or recently deployed safeguards or protection mechanism design faults, inappropriate application, or issues identified as anomolies resulting from new technology implementations.

Issue of measurability of risk — This focuses on the management of uncertainty surrounding the state of information resources. Uncertainty is identified, quantified, assessed, and is used to ascertain residual risk resulting from unavailable or improperly deployed safeguards and protection mechanisms, implementation of new technology, or speculative change (e.g., legislative or regulatory mandates, politics, etc.).

Challenges in managing critical resources — In which the management of uncertainty of impacts includes the design and implementation of:

1. Indicators that provide continuous visibility of the states of confidence.
2. Sensors and procedures that can positively verify the identity and privilege status of access to information, including verification of connectivity and interfaces.
3. Administrative and electronic controls to ensure separation of duty and assignment of privilege, and to limit unintentional or unauthorized granting and propagation of privileges.
4. Administrative and electronic mechanisms for assuring continuity of access to information, including the capability to restore systems to a known state that have been or, are perceived to be in the process of being interrupted by natural or induced disasters.

Administrative challenge of controlling and safeguarding access to and usage of proprietary information — In which an independent verification and validation process is institutionalized that attests to an acceptable status of trust in the integrity of information resources, systems, and agents.

Challenge of technology infusion — In which the management of enhancements to technology is addressed. Currently, technological enhancements of products and services is expanding at a phenomenal rate, while management methodologies, prototyping strategies, and tactical planning for their incorporation into enterprise domains are expanding at a much slower rate. Due to the dynamics and the proliferation of products and services, management is faced with a significant degree of uncertainty in deciding whether or not to use freeware, shareware, COTS products, or end-user-developed systems. Furthermore, if these are used, how will management control proprietary and/or critical information,

when should they be used, and what will be the associated long-range sustaining costs?

"EYE OF NEWT, HAIR OF DOG, BLOOD OF BAT, . . ."

In conclusion, information security is bounded only by our own prejudices and short sightedness.

In the last five years, security has changed from a discipline that was fairly isolated and unique, and easily controlled and administered, into a management dream turned into a nightmare. The Security "druids" of the 1980s, crouched over boiling cauldrons muttering strange incantations and peering into the future, have been replaced with the 1990s "techno-wennies" and "security geeks" who were let out of their closets gloomily forecasting that:

- Security can no longer be effectively added as an independent layer of protection.
- Every PC is equivalent to an international data center and should be similarly protected.
- Security in a distributed environment is a logical configuration, and cannot be physically controlled.
- Security cannot be legislated.
- Security is an operational decision, it is not part of the development life cycle and therefore, should not be addressed as a technical requirement until after a system is built and delivered.
- Once systems are opened, they can probably never be closed.
- Effective security is cost prohibitive and we can't do anything about it until a COTS product is available.

We have looked "SATAN" in the eye (1994) and "danced with the devil in the pale moonlight (1995,1996)". We are still here, the values, issues, and concerns are still here. Although we have made progress in determining what is needed, we are still ignoring the simple fact that adequate security safeguards and protection mechanisms have to be designed for, and built into our systems. We must take the initiative by accepting a synergistic approach that combines the current development and maintenance disciplines into a single Integrity Engineering discipline as the future answer to our concerns.

Introduction to UNIX Security for Security Practitioners

Jeffery J. Lowder

IN AN AGE OF INCREASINGLY SOPHISTICATED SECURITY TOOLS (e.g., firewalls, virtual private networks, intrusion detection systems, etc.), MANY PEOPLE DO NOT CONSIDER OPERATING SYSTEM SECURITY A VERY SEXY TOPIC. Indeed, given that the UNIX operating system was originally developed in 1969 and that multiple full-length books have been written on protecting UNIX machines, one might be tempted to dismiss the entire topic as "old hat." Nevertheless, operating system security is a crucial component of an overall security program. In the words of Anup Ghosh, the operating system is "the foundation for any software that runs on a machine," and this is just as true in the era of E-commerce as it was in the past. Thus, security practitioners who are even indirectly responsible for the protection of UNIX machines need to have at least a basic understanding of UNIX security. This chapter attempts to address that need by providing an overview of security services common to all flavors of UNIX; security mechanisms available in trusted UNIX are beyond the scope of this chapter (but see Exhibit 21-1).

OPERATING SYSTEM SECURITY SERVICES

Summers[7] lists the following security services that operating systems in general can provide:

1. *Identification and authentication.* A secure operating system must be able to distinguish between different users (identification); it also needs some assurance that users are who they say they are (authentication). Identification and authentication are crucial to the other operating system security services. There are typically three ways to authenticate users: something the user *knows* (e.g., a password), something the user *has* (e.g., a smart card), or something the user *is*

Exhibit 21-1. Versions of trusted or secure UNIX.

A1 (Verified Design)	No operating systems have been evaluated in class A1
B3 (Security Domains)	Wang Government Services, Inc. XTS-300 STOP 4.4.2
B2 (Structured Protection)	Trusted Information Systems, Inc. Trusted XENIX 4.0
B1 (Labeled Security Protection)	Digital Equipment Corporation ULTRIX MLS+ Version 2.1 on VAX Station 3100 Hewlett Packard Corporation HP-UX BLS Release 9.09+ Silicon Graphics Inc. Trusted IRIX/B Release 4.0.5EPL
C2 (Controlled Access Protection)	No UNIX operating systems have been evaluated in class C2
C1 (Discretionary Access Protection)	Products are no longer evaluated at this class
D1 (Minimal Protection)	No operating systems have been evaluated in class D1

Note: Various versions of UNIX have been evaluated by the U.S. Government's National Security Agency (NSA) according to the Trusted Computer System Evaluation Criteria. (By way of comparison, Microsoft Corporation's Windows NT Workstation and Windows NT Server, Version 4.0, have both been evaluated at class C2.) The above chart is taken from the NSA's Evaluated Product List.

(e.g., a retinal pattern). Passwords are by far the most common authentication method; this method is also extremely vulnerable to compromise. Passwords can be null, easily guessed, cracked, written down and then discovered, or "sniffed."

2. *Access control.* An operating system is responsible for providing logical access control through the use of subjects, objects, access rights, and access validation. A subject includes a userID, password, group memberships, privileges, etc. for each user. Object security information includes the owner, group, access restrictions, etc. Basic access rights include read, write, and execute. Finally, an operating system evaluates an access request (consisting of a subject, an object, and the requested access) according to access validation rules.
3. *Availability and integrity.* Does the system start up in a secure fashion? Does the system behave according to expectations during an attack? Is the data on the system internally consistent? Does the data correspond with the real-world entities that it represents?
4. *Audit.* An audit trail contains a chronological record of events. Audit trails can be useful as a deterrent; they are even more useful in investigating incidents (e.g., Who did it? How?). Audit trails have even been used as legal evidence in criminal trials. However, for an audit trail to be useful in any of these contexts, the operating system must record all security-relevant events, protect the confidentiality and integrity of the audit trail, and ensure that the data is available in a timely manner.

5. *Security facilities for users.* Non-privileged users need some method for granting rights to their files and changing their passwords. Privileged users need additional facilities, including the ability to lock accounts, gain access to other users' files, configure auditing options, change ownership of files, change users' memberships in groups, etc.

The following pages explore how these services are implemented in the UNIX family of operating systems.

IDENTIFICATION AND AUTHENTICATION

UNIX identifies users according to usernames and authenticates them with passwords. In many implementations of UNIX, both usernames and passwords are limited to eight characters. As a security measure, UNIX does not store passwords in plaintext. Instead, it stores the password as ciphertext, using a modified Digital Encryption Standard (DES) algorithm (crypt) for encryption. The encrypted password, along with other pertinent account information (see Exhibit 21-2), is stored in the */etc/passwd* file according to the following format:

```
username:encrypted password:UserID:GroupID:user's
          full name:home directory:login shell
```

Exhibit 21-2. Sample */etc/passwd* entries.

```
keith::1001:15:Keith Smith:/usr/keith:/bin/csh
greg:Qf@14pL1aqzqB:Greg Jones:/usr/greg/:/bin/csh
cathy:*:1003:15:Cathy Jones:/usr/cathy:/bin/csh
```

(In this example, user keith has no password, user greg has an encrypted password, and user cathy has a shadowed password.)

Unfortunately, the */etc/passwd* file is world-readable, which can place standard, "out-of-the-box" configurations of UNIX at risk for a brute-force password-guessing attack by anyone with system access. Given enough computing resources and readily available tools like Alec Muffet's **crack** utility, an attacker can eventually guess every password on the system. In light of this vulnerability, all current implementations of UNIX now provide support for so-called "shadow" passwords. The basic idea is to store the encrypted passwords in a separate file (*/etc/shadow* to be exact) that is only readable by the privileged "root" account. Also, although vanilla UNIX does not provide support for proactive password checking, add-on tools are available. Finally, password aging is not part of standard UNIX but is supported by many proprietary implementations.

UserIDs (UIDs) are typically 16-bit integers, meaning that they can have any value between 0 and 65,535. *The operating system uses UIDs, not usernames, to track users.* Thus, it is entirely possible in UNIX for two or more usernames to share the same UID. In general, it is a bad idea to give two usernames the same UID. Also, certain UIDs are reserved. (For example, any username with an UID of zero is considered root by the operating system.) Finally, UNIX requires that certain programs like **/bin/passwd** (used by users to change their passwords) and */bin/login* (executed when a user initiates a login sequence) run as root; however, users should not be able to arbitrarily gain root permissions on the system. UNIX solves this problem by allowing certain programs to run under the permissions of another UID. Such programs are called Set UserID (SUID) programs. Of course, such programs can also be risky: if attackers are able to interrupt an SUID program, they may be able to gain root access and ensure that they are able to regain such access in the future.

GroupIDs (GIDs) are also typically 16-bit integers. The GID listed in a user's entry in */etc/passwd* is that user's primary GID; however, in some versions of UNIX, a user can belong to more than one group. A complete listing of all groups, including name, GID, and members (users), can be found in the file */etc/group*.

Once a user successfully logs in, UNIX executes the global file */etc/profile* along with the *.profile* file in the user's home directory using the user's shell specified in */etc/passwd*. If the permissions on these files are not restricted properly, an attacker could modify these files and cause unauthorized commands to be executed each time the user logs in. UNIX also updates the file */usr/adm/lastlog,* which stores the date and time of the latest login for each account. This information can be obtained via the **finger** command and creates another vulnerability: systems with the **finger** command enabled may unwittingly provide attackers with useful information in planning an attack.

ACCESS CONTROL

Standard UNIX systems prevent the unauthorized use of system resources (e.g., files, memory, devices, etc.) by promoting discretionary access control. Permissions are divided into three categories: owner, group, and other. However, privileged accounts can bypass this access control. UNIX treats all system resources consistently by making no distinction between files, memory, and devices; all resources are treated as files for access control purposes.

The UNIX filesystem has a tree structure, with the top-level directory designated as **/**. Some of the second-level directories are standards. For example, */bin* contains system executables, */dev* contains devices, */usr*

contains user files, etc. Each directory contains a pointer to itself (the '.' file) and a pointer to its parent directory (the '..' file). (In the top-level directory, the '..' file points to the top-level directory.) Every file (and directory) has an owner, a group, and a set of permissions. This information can be o btained using the **ls -l** command:

```
drwxr-xr-x   1  jlowder  staff    1024   Feb 21 18:30   ./
drwxr-xr-x   2  jlowder  staff    1024   Oct 28 1996    ../
-rw-------   3  jlowder  staff    2048   Feb 21 18:31   file1
-rw-rw----   4  jlowder  staff    2048   Feb 21 18:31   file2
-rw-rw-rw-   5  jlowder  staff    2048   Feb 21 18:31   file3
-rws------   6  jlowder  staff   18495   Feb 21 18:31   file4
```

In the above example, file1 is readable and writable only by the owner; file2 is readable and writable by both the owner and members of the 'staff' group; file3 is readable and writable by everyone; and file4 is readable and writable by the owner and is a SetUID program.

Devices are displayed a bit differently. The following is the output of the command **ls -l /dev/cdrom /dev/tty02**:

```
br--------   1   root   root   1024  Oct 28 1996   /dev/cdrom
crw-------   2   root   root   1024  Oct 28 1996   /dev/tty02
```

UNIX identifies block devices (e.g., disks) with the letter 'b' and character devices (e.g., modems, printers) with the letter 'c.'

When a user or process creates a new file, the file is given default permissions. For a process-created file (e.g., a file created by a text editor), the process specifies the default permissions. For user-created files, the default permissions are specified in the startup file for the user's shell program. File owners can change the permissions (or mode) of a file by using the **chmod** (change mode) command.

UNIX operating systems treat directories as files, but as a special type of file. Directory "files" have a specified structure, consisting of filename-inode number pairs. Inode numbers refer to a given inode, a sort of record containing information about where parts of the file are stored, file permissions, ownership, group, etc. The important thing to note about the filename-inode number pairs is that *inode numbers need not be unique*. Multiple filenames can (and often do) refer to the same inode number. This is significant from a security perspective, because the **rm** command only removes the directory entry for a file, not the file itself. Thus, to remove a file, one must remove all of the links to that file.

AVAILABILITY AND INTEGRITY

One aspect of availability is whether a system restarts securely after failure. Traditional UNIX systems boot in single-user mode, usually as root. And, unfortunately, single-user mode allows literally anyone sitting at the system console to execute privileged commands. Thus, single-user mode represents a security vulnerability in traditional UNIX. Depending on the flavor of UNIX, the security administrator has one or two options for closing this hole. First, if the operating system supports it, the security practitioner should configure the system to require a password before booting in single-user mode. Second, tight physical controls should be implemented to prevent physical access to the system console.

System restarts are also relevant to system integrity. After an improper shutdown or system crash, the UNIX **fsck** command will check filesystems for inconsistencies and repair them (either automatically or with administrator interaction). Using the **fsck** command, an administrator can detect unreferenced inodes, used disk blocks listed as free blocks, etc.

Although there are many ways to supplement UNIX filesystem integrity, one method has become so popular that it deserves to be mentioned here. Developed by Gene Kim and Gene Spafford of Purdue University, Tripwire is an add-on utility that provides additional filesystem integrity by creating a signature or message digest for each file to be monitored. Tripwire allows administrators to specify what files or directories to monitor, which attributes of an object to monitor, and which message digest algorithm (e.g., MD5, SHA, etc.) to use in generating signatures. When executed, Tripwire reports on changed, added, or deleted files. Thus, not only can Tripwire detect Trojan horses, but it can also detect changes that violate organizational policy.

AUDIT

Different flavors of UNIX use different directories to hold their log files (e.g., */usr/adm, /var/adm,* or */var/log*). But wherever the directory is located, traditional UNIX records security-relevant events in the following log files:

- *lastlog:* records the last time a user logged in
- *utmp:* records accounting information used by the **who** command
- *wtmp:* records every time a user logs in or out; this information can be retrieved using the **last** command.
- *acct:* records all executed commands; this information can be obtained using the **lastcomm** command (unfortunately, there is no way to select events or users to record; thus, this log can consume an enormous amount of disk space if implemented)

Furthermore, most versions of UNIX support the following logfiles:

- *sulog:* logs all su attempts, and indicates whether they were successful
- *messages:* records a copy of all the messages sent to the console and other *syslog* messages

Additionally, most versions of UNIX provide a generic logging utility called *syslog*. Originally designed for the *sendmail* program, *syslog* accepts messages from literally any program. (This also creates an interesting audit vulnerability: any user can create false log entries.) Messages consist of the program name, facility, priority, and the log message itself; the system prepends each message with the system date, time, and name. For example:

```
Nov 7 04:02:00 alvin syslogd: restart
Nov 7 04:10:15 alvin login: ROOT LOGIN REFUSED on ttya
Nov 7 04:10:21 alvin login: ROOT LOGIN on console
```

The *syslog* facility is highly configurable; administrators specify in */etc/syslog.conf* what to log and how to log it. *syslog* recognizes multiple security states or priorities, including emerg (emergency), alert (immediate action required), crit (critical condition), err (ordinary error), warning, notice, info, and debug. Furthermore, *syslog* allows messages to be stored in (or sent to) multiple locations, including files, devices (e.g., console, printer, etc.), and even other machines. These last two options make it much more difficult for intruders to hide their tracks. (Of course, if intruders have superuser privileges, they can change the logging configuration or even stop logging altogether.)

SECURITY FACILITIES FOR USERS

Traditional UNIX supports one privileged administrative role (the "root" account). The root account can create, modify, suspend, and delete user accounts; configure auditing options; administer group memberships; add or remove filesystems; execute any program on the system; shut the system down; etc. In short, root accounts have all possible privileges. This violates both the principle of separation of duties (by not having a separate role for operators, security administrators, etc.) and the principle of complete mediation (by exempting root from access control).

Non-privileged users can change their passwords using the **passwd** command, and they can modify the permissions of their files and directories using the **chmod** program.

MISCELLANEOUS TOPICS

Finally, there are a few miscellaneous topics that pertain to UNIX security but do not neatly fall into one of the categories of operating system

security listed at the beginning of this chapter. These miscellaneous topics include *tcpwrapper* and fundamental operating system holes.

Vulnerabilities in Traditional UNIX

Many (but by no means all) UNIX security vulnerabilities result from flaws in its original design. Consider the following examples:

1. *Insecure defaults.* Traditional UNIX was designed for developers; it is shipped with insecure defaults. Out-of-the-box UNIX configurations include enabled default accounts with known default passwords. Traditional UNIX also ships with several services open by default, password shadowing not enabled, etc. Administrators should immediately disable unnecessary accounts and ports. If a default account is necessary, the administrator should change the password.
2. *Superuser and SUID attacks.* Given that UNIX does not have different privileged roles, anyone who compromises the root account has compromised the entire system. When combined with SUID programs, the combination can be disastrous. An attacker need simply "trick" the SUID program into executing an attack, either by modifying the SUID program or by supplying bogus inputs. If the SUID program runs as root, then the attack is likewise executed as root. Given this vulnerability, SUID programs should be prohibited if at all feasible; if not, the system administrator must continually monitor SUID programs to ensure they have not been tampered with.
3. *PATH and Trojan horse attacks.* When a user requests a file, the PATH environment variable specifies the directories that will be searched and the order in which they will be searched. By positioning a Trojan horse version of a command in a directory listed in the search path, such that the Trojan horse directory appears prior to the real program's directory, an attacker could get a user to execute the Trojan horse. Therefore, to avoid this vulnerability in the PATH variable, administrators can specify absolute filepaths and place the user's home directory last.
4. *Trust relationships.* UNIX allows both administrators and users to specify trusted hosts. Administrators can specify trusted hosts in the */etc/hosts.equiv* file and users in a file named *.rhosts* in their home directory. When a trust relationship exists, a user on a trusted (remote) machine can log into the local machine without entering a password. Furthermore, when the trust relationship is defined by an administrator in the */etc/hosts.equiv* file, the remote user can log into the local machine *as any user on the local system*, again without entering a password. Clearly, this is extremely risky. Even if one

trusts the users on the remote machine, there are still two significant risks. First, the trust relationships are transitive. If one trusts person A, then one implicitly trusts everyone who person A trusts. Second, if the remote machine is compromised, the local machine is at risk. For these reasons, trust relationships are extremely risky and should almost always be avoided.

TCP Wrapper

Written by Wietse Venema, *tcpwrapper* allows one to filter, monitor, and log incoming requests for various Internet services (systat, finger, ftp, telnet, rlogin, rsh, exec, tftp, talk, etc.). The utility is highly transparent; it does not require any changes to existing software. The chief advantage of *tcpwrapper* is that it provides a decent access control mechanism for network services. For example, an administrator might want to allow incoming FTP connections, but only from a specific network. *tcpwrapper* provides a convenient, consistent method for implementing this type of access control. Depending on the implementation of UNIX, *tcpwrapper* might also provide superior audit trails for the services it supports.

Login or Warning Banner

UNIX can be configured to display a "message of the day," specified in the file */etc/motd*, to all users upon login. At least part of this message should be a so-called login or warning banner, advising would-be attackers that access to system resources constitutes consent to monitoring and that unauthorized use could lead to criminal prosecution (see Exhibit 21-3).

Exhibit 21-3. Sample warning banner.

```
WARNING: THIS SYSTEM FOR AUTHORIZED USE ONLY. USE OF THIS
SYSTEM CONSTITUTES CONSENT TO MONITORING; UNAUTHORIZED USE
COULD RESULT IN CRIMINAL PROSECUTION. IF YOU DO NOT AGREE
TO THESE CONDITIONS, DO NOT LOG IN!
```

CONCLUSION

Traditional UNIX implements some of the components of operating systems security to varying extents. It has many well-known vulnerabilities; out-of-the-box configurations should not be trusted. Furthermore, add-on security tools can supplement core UNIX services. With proper configuration, a UNIX system can be reasonably protected from would-be intruders or attackers.

References

1. Anonymous, *Maximum Security*, Sams.net, New York, 1997.
2. Farrow, Rik, *UNIX System Security: How to Protect Your Data and Prevent Intruders*, Addison-Wesley, New York, 1991.
3. Garfinkel, Simson and Spafford, Gene, *Practical UNIX and Internet Security*, 2nd ed., O'Reilly & Associates, Sebastopol, CA, 1996.
4. Ghosh, Anup K., *E-commerce Security: Weak Links, Best Defenses*, John Wiley & Sons, New York, 1998.
5. Gollmann, Dieter, *Computer Security*, John Wiley & Sons, New York, 1999.
6. National Security Agency, Evaluated Products List Indexed by Rating, <URL: http://www.radium.ncsc.nil/tpep/epl/epl-by-class.html>, January 31, 2000.
7. Summers, Rita C., *Secure Computing: Threats and Safeguards*, McGraw-Hill, New York, 1997.

Microcomputer and LAN Security

Stephen Cobb

INTRODUCTION

This chapter focuses on preserving the confidentiality, integrity, and availability of information in the microcomputer and local area network (LAN) environment. We often refer to this as the desktop environment, desktop computing, or PC-based computing.

Why Desktop Computing Matters

Although mainframe computers continue to be used extensively for such tasks as large-scale batch processing and online transaction processing, for many organizations today, computer security is, in effect, desktop computer security. Networked desktop computers are the dominant computing platform of the late 1990s, from the Microsoft Windows-based computers that some airlines use to check in passengers at airports, to the stock transaction and account inquiry systems used in banking and financial institutions, from personal computer-controlled assembly lines to PC-based medical information systems.

In many of these applications the personal computer may appear to be working as a terminal access device for a larger system. But from a security perspective it is important to understand that every personal computer system is a complete computer system, capable of input, output, storage, and processing. As such, a PC poses a much more significant threat than a dumb terminal, should the PC be subverted or illegally accessed. Furthermore, with very few exceptions, none of the desktop computing devices deployed today were designed with security in mind. Add to this the enormous increase in both the depth and the breadth of computer literacy within society over the last 10 years and you have a recipe for serious security headaches.[1]

The Approach Taken

All major aspects of desktop security will be addressed in this chapter, beginning with the need to address desktop issues within the organization's information security policies. Security awareness on the part of both users and managers is stressed. The need for, and implementation of, data backup systems and regimes is outlined. Passwords and other forms of authentication for desktop users are discussed, along with the use of encryption of information on desktop machines and LANs. There is a section on malicious code. The network dimensions of desktop computing security are explored, together with the problems of remote access.

Centralized, Layered, and Design-Based Approaches

A good case can be made for saying that desktop computer security is best handled through automated background processes, preferably centrally managed on a network.[2] Desktop computer users, so the argument goes, should not be expected to worry about backups and virus scanning and access controls. These security mechanisms should be handled for them as part of the operating system.

This sounds appealing, but there are several practical reasons why an understanding of the security weaknesses of stand-alone PCs and undermanaged LANs remains critical, and why, in at least some cases, it is necessary to implement piecemeal solutions that lack the elegance and obvious efficiency of the automated, centrally managed approach:

- A lot of desktop computers are currently connected to networks that have little hope of ever being centrally managed, yet the information they handle is still important and so warrants protection.
- Many of the methods for automating and managing security will only be applicable to, or compatible with, newer hardware and software. Older systems will remain in use and will still need to be protected.[3]
- Mature tools with which to automate and centrally manage security on local area networks are only just coming to market, and many organizations are only just realizing that they need them and will have to pay for them.
- A fairly high level of security can be achieved on both current and older personal computers with the layered approach, described next.

The layered approach to desktop security maximizes existing, but underutilized, security mechanisms, plus low-cost add-ons, through policy, awareness, and training. For example, the floppy disk drive of a PC is a major security problem. Confidential and proprietary data can be copied to a floppy diskette and smuggled out.[4] Incoming diskettes may introduce pirated software, Trojan code, and viruses to the company network. Yet the BIOS in most of today's PCs allows you to tightly control use of the floppy drive, for example, disabling boot from, read from, or write to. PC security

is considerably enhanced by implementing this type of control, which is essentially free. The layered approach would extend this protection by also requiring antivirus software on the PC and putting in place a company policy governing the use of floppy disks in the office. When employees understand the threat that a serious virus outbreak or data theft poses to their jobs, most are apt to support the policy.

DESKTOP SECURITY: PROBLEMS, THREATS, ISSUES

The problems, threats, and issues of desktop security need to be placed in perspective. A common, but dangerous, mistake is to underestimate the seriousness of this aspect of information system security. A clear understanding of desktop system architecture and its security implications is required.

The Ubiquitous Micro

Historically, desktop computers have been on the fringe of information security, which has its roots in the protection of very expensive, highly centralized, multi-user information processing systems. Today, desktop computers performing distributed computing are no longer on the fringe. Failure to realize this will undermine your ability to protect any information system, big or small, for four reasons:

1. A significant percentage of mission-critical computing is now performed on personal computers deployed as LAN work stations and network file servers.[5]
2. Most large-scale computer systems are at some point connected to one or more desktop systems. Even when PC connectivity is not specifically provided to a large system, PC access may be possible, for example, via a remote maintenance line.
3. Inexpensive and widely available desktop systems now have the power to mount attacks that endanger the security of large-scale systems, such as brute force cryptanalysis, password-cracking, and denial-of-service attacks.[6]
4. Knowledge about how to use, and abuse, desktop computers is widely dispersed throughout most areas of society and most countries of the world. This is a far less homogeneous, and thus less predictable, population than previous generations of computer users.[7]
5. Such knowledge, particularly new developments in software techniques that can be abused to compromise security, is instantly accessible via the Internet.[8]

Clearly, an understanding of desktop security is more important than ever. Desktop machines are an integral part of the client–server distributed computing paradigm that dominates the late 1990s. In the vast majority of systems, the clients to which servers serve up data are microcomputers;

the primary topology by which they do this is the local area network. Furthermore, in an increasing number of systems, the servers themselves are essentially beefed-up microcomputers. This is particularly true of the Internet, which is beginning to rival leased lines and private value-added networks as the data communication channel of choice.

Desktop System Architecture

Although you may be familiar with the following definitions they are stated here because they have important security implications which are not always understood.[9] A microcomputer is a computer system in miniature, a collection of hardware and software that is small enough to fit on a desk (or into a briefcase or even a shirt pocket) but able to perform the four major functions that define a computer system: input, processing, storage, and output. Note that processing requires both a processor and random access memory (RAM). Also note that RAM is different from storage (data that are stored remains accessible after system reset or reboot, data held in RAM are typically not accessible after system reset or reboot).

Soon after microcomputers were developed, the term "personal computer" was coined to describe these self-contained computer systems. This was later shortened to "PC" although this term is often used to refer to a specific type of personal computer, that is, one based on the nonproprietary architecture developed by IBM around the Intel 8086 family of processors (including the 80286, 80386, 80486, and Pentium chips).

Today, the majority of personal computers conform to the IBM/Intel architecture, and most of these run the DOS/Microsoft Windows operating systems (a small but significant percentage still adhere to the proprietary Apple Macintosh architecture). A separate class of desktop machines are those using the UNIX operating system. Often referred to as "work stations," these UNIX machines are typically more expensive, more powerful, and confined to specialized areas such as engineering and scientific research. While the DOS and Windows 95 operating systems use an open file system, with no provision for separate user accounts on a single machine, UNIX offers tight control of file permissions and multiple accounts. UNIX machines are often used as high-performance back-room database hosts and World Wide Web servers.

Recently, a new category of machine, the network computer or NC, has been making headlines. In many ways this is simply the re-birth of the diskless PC, several models of which were unsuccessfully marketed in the late 1980s. Both the NC and the diskless PC are machines that have their own processor and random access memory and so perform local processing, but possess no local storage devices. Their operating system is a combination of a ROM-based boot process and server-based network operating system. However, whereas the diskless PC was aimed at solving

security, management, and support problems on local area networks, the NC concept has been developed in a wide area context, specifically the Internet, and in particular, the World Wide Web.

Strict categorization of desktop systems is seldom helpful. For example, IBM/Intel-based machines can run powerful versions of UNIX, such as SCO UNIX. Both BSDI UNIX and Linux run on Intel chips and are very popular as Web servers. Furthermore, Microsoft Windows NT and IBM OS/2 both offer a multi-user, multitasking alternative to UNIX, with a familiar graphical user interface (GUI). They also allow you to use a closed file system. What may be helpful is further clarification of the terms PC, work station, terminal, server, and client.

- PC: a self-contained computer system with its own processor, storage, and output devices (the screen is perhaps the most basic of output devices). Typically, it is small enough to fit on or under a desk.
- Work station: a self-contained computer system with its own processor that is also connected to a server. A work station does at least some of its own processing and may have its own storage, but may also use or rely on the server for storage.
- Terminal: a computer access device with screen and keyboard that does not have its own processing or storage capabilities.
- Server: any computer system that is providing access to its resources to another computer system, for example, a Web server provides a browser/client with access to Web pages stored on the server.
- Client: any computer system that is accessing resources made available to it by another computer system, for example, a Web browser/client accesses to Web pages stored on a Web server.

DESKTOP SECURITY POLICY AND AWARENESS

Every organization should have an information security policy. However, field experience suggests that these policies often fail to address desktop computing issues appropriately or adequately. For example, it is common for companies to have comprehensive policies for mainframe systems that address all contingencies, but only a few specific desktop policies such as antivirus procedures written in response to specific incidents such as a virus infection.

From the Top Down

Effective information security policies are created from the top down, beginning with the organization's basic commitment to information security formulated as a general policy statement. Here is a good example of a general policy statement:

1. Timely access to reliable information is vital to the continued success of Megabank.
2. Protection of Megabank's information assets and facilities is the responsibility of each and every employee and officer of Megabank.
3. The information assets and processing facilities of Megabank are the property of Megabank and may only be used for Megabank business as authorized by Megabank management.

When a general policy like this has been agreed to by top management, each employee should be required to sign, upon hiring and each year thereafter, a document consisting of the policy statement and words to this effect:

> I have read and understood the company's information security policy and agree to abide by it. I realize that serious violations of this policy are legitimate grounds for dismissal.

Once you have a general policy like this in place, you can elaborate upon particulars. In the case of desktop systems these include:

- Password policies (e.g., minimum length, storage of passwords)
- Backup duties (for individual PCs as well as the network server)
- Data classification (rating each document for sensitivity)
- Removable media handling (e.g., who can take diskettes in or out)
- Encryption (what data will be encrypted, which algorithms to use)
- Physical security (how is equipment protected against theft/tampering)
- Access policies (who is allowed to access which machines/files)

There will also need to be policies for specific systems, for example, the accounting department LAN. These can be promulgated by the staff who have responsibility for those systems provided there is oversight and sign-off by the managers of those departments and the security staff.

The Fine Print

The task of developing detailed policy is often avoided because it is seen as too daunting. It is sometimes postponed because "there is no way to predict where information technology will go next." While this is true, you need specific policies as soon as they become feasible, plus a general policy to deal with emerging areas of concern. For example, consider the fairly recent ability to browse the World Wide Web with a desktop computer attached to the company's Internet connection. It is now possible to formulate specific policy such as "employees must not use company systems to visit Web sites that contain sexually explicit material."

However, in companies where employees have, for a time at least, enjoyed unrestricted Web access, such specific policies may be resisted (as though browsing the Web on the company's dime is a right, just like selecting your own desktop design or installing your own games). But if the company has a preexisting general policy statement that asserts ownership of information

processing assets, any restrictions on how PCs may be used can immediately be vindicated and enforced because it is clearly in keeping with that policy.

On the other hand, you have to be realistic. The desktop computing environment is inherently difficult to control and so the most effective policies are those which are understood and accepted by those who must abide by them. Developing policy by consensus is clearly more effective in this environment than policy by decree. To this end, high-level policy statements which establish the company's right to control its own computers play an important psychological role.

Desktop Security Awareness

It is not enough to develop security policies for desktop systems. Users must be told what the policies are and trained to support them. The ideal situation is a self-regulating workforce so that, for example, when Fred in engineering brings to work a game on a floppy disk that his son brought home from school the night before, Mary will refuse to put it in her PC because she knows that (1) it is a violation of security policy, and (2) it exposes her PC, and thus the company LAN, to the risk of virus infection; and (3) LAN downtime and person-hours consumed by virus disinfection have a negative effect on company profitability, which in turn has a negative effect on her earnings and employment prospects.

Raising employee security awareness to this level requires a significant training effort, but it is money well spent relative to more technology-oriented solutions. In an age of universal computer literacy it would be foolish to rely solely upon high-tech security systems, since there will always be people with the skills to challenge such defenses. You can reduce the incentive to mount such challenges by eschewing policy dictation in favor of consensus-based policy making. If employees understand and thus "buy-in" to the policy, the technical defenses can be concentrated in the areas of greatest effectiveness.

Determining those areas is an ongoing process which depends upon a different type of security awareness: that which you cultivate as a security professional. It involves staying current with the latest trends in computer insecurity, for example, new virus outbreaks, newly discovered operating system vulnerabilities, and so on. You maintain this awareness by subscribing to industry publications, participating in online forums and mailing lists, attending security conferences, and networking with fellow security professionals.

PHYSICAL SECURITY: DESKTOPS AND LAPTOPS

Efforts to thwart computer equipment theft are a good illustration of the importance of security awareness. For example, do you know the total

value of desktop computer equipment that is stolen every year in North America? The answer, according to SAFEWARE, the Columbus, Ohio-based computer insurance specialist, is quite staggering: more than $1 billion. Consider some of the security implications of desktop computer theft:

- All data on a stolen hard drive that was not backed up is now lost.
- No data can be accessed in a timely manner while backups are restored to replacement equipment.
- Certain components, such a custom cables, are hard to replace if stolen.
- Most PC-based systems depend upon a very specific configuration of hardware and software which may be difficult to replicate on replacement systems.
- Unless it was encrypted, anyone who receives a stolen PC has access to the data stored on it.
- If the stolen PC is recovered it is very hard to know whether or not someone made a copy of the data that was stored on it.

Obviously, your information security policy should mandate that backups of all data be available at all times (this typically requires off-site backup storage as a defense against backup media being stolen along with the systems backed up thereon). However, even if you are in compliance with this lofty goal, backups cannot solve every security problem. If a competitor obtains copies of your trade secrets by stealing your computers, having a backup copy is not much consolation.[10]

Awareness of current trends in computer theft will not only help you plan countermeasures, but also help you refine policy and provide timely security awareness training. The first point to note is that personal computers are now a commodity, like VCRs, camcorders, and stereos. This means they can be turned into cash very quickly, making them a target for casual thieves and those supporting drug habits. Because of their higher value-to-weight ratio, notebook computers are very popular with this type of thief.

More organized felons will target notebooks at locations such as airports, where there are rich pickings. For example, a popular tactic in recent years has been for two-person teams to steal notebooks at security check points. One thief waits until a notebook-bearing bag is placed on the conveyor belt to the X-ray machine, then holds up the line going through the metal detector (not hard to do). The accomplice waiting on the other side of the check point simply picks up the bag and departs.

While desktop systems in offices are sometimes targeted by the “smash and grab for cash thief,” the more serious risk may be sophisticated criminals stealing to order. Such thieves tend to target high-end equipment like graphics work stations, large monitors, and production-quality typesetters

and color scanners. European offices seem to be particularly vulnerable due to the high demand and relative lack of resources in former Eastern bloc countries. On occasion, Scotland Yard has recovered trucks full of expensive Apple Macintosh desktop publishing equipment stolen to order and destined for Eastern Europe.

A slightly different combination of factors led to a rash of chip heists in the early 1990s. Shortages of memory chips resulted in high prices and led to several types of theft. Europe experienced a rash of thefts in which chips were removed from office systems. Employees arrived in the morning to find desktop computers torn apart (none too gracefully) and the memory chips removed. This represents a major blow to any organization (a charity for the elderly and the Automobile Association were two of the victims). No data processing can occur until the chips are replaced. Specification of chips for used equipment is no simple matter (there are many different types and many compatibility issues). Even if you can afford the high replacement cost there may be delays obtaining chips. After all, the motive for the theft was high prices caused by a shortage.

A different type of theft occurred in chip producing areas such as America's Silicon Valley and Scotland's Silicon Glen. This involved direct, and sometimes violent, attacks on chip factories and shipping facilities. However, the motivating factors were the same: memory chips are easily resold, hard to trace, and they can have a higher value-to-weight ratio than gold or platinum.

The point of these examples is that as an information systems security professional you need to be keenly aware of the current economics of both crime and computing. As this chapter is being written, memory prices are at an all-time low, reducing the incentive for chip theft, and possibly impacting your spending on countermeasures, relative to other threats. However, if prices suddenly rise again you will need to tighten security measures in this particular area.[11] Some specific microcomputer physical security measures to consider include:

1. Good site security: this not only protects against theft, but also against vandalism, unauthorized access, and media removal.
2. Case locks: these not only deter theft of internal components, but also protect BIOS-based security services, described elsewhere in this chapter.
3. Documentation: you need to keep detailed records of all your hardware and software, including serial numbers, purchase dates, invoices, and so on. These records will be invaluable if you ever have to prove loss or reclaim stolen items that have been recovered.
4. Insurance: computer equipment typically requires separate insurance or a special rider in your business insurance or office contents

policy. Note that home contents policies often exclude computers used for work.

5. Access controls and encryption: if a computer is stolen you would like to make it as difficult as possible for the person who ends up trying to use it to access the data that are stored on the system.

DESKTOP DATA BACKUP

Clearly, the single most effective technical strategy you can employ to defend the integrity and availability of computer-based data is making backup copies, often simply referred to as backup. This is standard doctrine for most information systems professionals, particularly those familiar with the mainframe environment, where backup is an integral part of computing. However, in the desktop environment, which is based on systems that have their origins in casual, even recreational use, the task of backing up is all too often neglected until it is too late.[12]

Backup Types and Devices

Most "live" data in use today are stored on hard disk drives. While the reliability of the hard disk devices found in desktop and laptop systems has steadily improved over the last decade, they are nevertheless mechanical devices quite capable of wearing out, sometimes prematurely, sometimes without warning. Furthermore, users are only human, often lacking in formal training. Sometimes they erase important files or records within files by mistake. Sometimes they delete data out of malice. Viruses and other malicious programs can destroy files. Making backup copies of all of the files that are on a hard disk is the best, and often the only, means of recovery from mechanical failure, user error, malevolent software, natural disaster, and physical theft.

Hard drives have finite storage capacity. Eventually you have to erase files from the hard disk to make way for more. You may need to keep copies of those "surplus" files, such as last year's bookkeeping ledger. These days some people use two computers, one on the desk at work, another that travels with the user or resides in the user's home. Thus we can identify at least four different types of file copying, as listed in Exhibit 23.1.

Backups=Copies of files made to defend against loss/corruption of originals
Archives=Copies of files made to relieve overcrowding on primary storage devices
Updates=Copies of files made to synchronize files between two machines
Duplicates=Copies of files made to provide other users with copies of programs or data

Exhibit 23.1. Four Different Types of File Copying

The main focus in this section is backups, but the other categories are also important. Updates that synchronize files between desktops and portable machines are a relatively recent concern and have implications for data integrity. An archive is a set of files that has been copied as an historical record. Typically these are files containing data that will not change, and immediate access to which is no longer required, such as properly aged accounting records. When the archive copy has been created the original can be erased, thus freeing up storage space. Several terms that are useful at this point are

- Primary storage — where frequently used software and data reside.
- Online storage — storage that is immediately available and randomly accessible; this includes removable media such as floppy diskettes.
- Removable media — any media that can be physically removed from the system, such as diskettes and CD-ROMs.
- Magnetic media — storage based on magnetic properties, such as hard drives, tapes, and floppies.
- Optical media — storage based on optical properties, such as CD-ROMs.
- Magneto-optical — storage based on a combination of magnetic and optical properties, like some high-capacity cartridge drives.
- Random vs. linear access — the ability to immediately access data regardless of their physical location on the media (e.g., a hard drive) as opposed to access which requires reading preceding data (e.g., a tape drive).
- Read only — the ability to read stored data but not change it.
- Write once, read many — the ability to record data in read only form and then read it multiple times (e.g., burning a CD-ROM).
- RAID — redundant array of inexpensive disks — a storage system which combines multiple disks managed as a single storage device, allowing disks to be "hot swapped," i.e., replaced without powering down or losing data.
- Jukebox — a storage system which combines multiple tapes or CD-ROM drives managed as a single storage device with automated media switching, providing large-scale storage or backup.

In the early days of personal computing the primary means of backup, software duplication, and archiving, was the floppy diskette. A floppy diskette can be described as randomly accessible removable media, with write many/read many, as well as read only capability (by physically adjusting the write-protect setting on the disk jacket you can write-protect the contents, although this is a reversible procedure, distinguishable from

Type	Capacity	Comments
Floppy diskettes	1.44 Mb	Standard equipment Low capacity, slow, cheap, tedious.
Tape drives e.g., Travan, Exabyte, DAT	400 Mb–9 Gb	Low media cost, highly automated, most widely used.
Removable cartridges e.g., Syquest, Jaz, Zip	200 Mb–4.6 Gb	High media cost, very fast, good for online systems.
CD-ROM	650 Mb	Low media cost, slow to make, convenient access.

Exhibit 23.2. Backup Options

WORM media that is physically impossible to overwrite). The floppy diskette has several benefits:

- Low cost for both drives and media
- Included as standard equipment on all machines
- Widespread compatibility between systems

Unfortunately, hard drive capacities and the complexity of both software and data have far outstripped the capacity of standard diskettes, while possible alternatives such as high-capacity cartridge drives and read/write optical media have so far failed to achieve anything like the same level of acceptance as standard equipment. The current options for backup are listed in Exhibit 23.2. Note that some of these removable media devices also work as primary storage, for active software and live data, as well as secondary or backup storage.

While constant improvements in performance, capacity, and pricing make "best buy" statements about storage devices imprudent, there are clearly some practical points that can be made. First of all, you need to match capacity and speed to need. For example, if a desktop machine uses about 600 megabytes of hard drive storage, 5 megabytes of which is updated every day, a CD-R drive might be worth considering as an alternative to tape. But tape would be better for a system that regularly stores twice as much data and updates data at a faster daily rate. For a network file server that stores several gigabytes of constantly changing data, you will probably want to use RAID for primary storage and a jukebox for constant backup.[13]

Boosting Backup

If desktop users are on a network, part of the backup problem has been solved. Any data they store on the file server will be backed up as part of normal network management (any network file server worthy of the name will have a built-in backup device, typically tape, and any network administrator worthy of the name will use it diligently). But unless the network

work stations are diskless, there will be a residual problem of local backup. It is possible to back up local work station storage through the file server, but this is not always practical (typically the work station must be on with the user logged in but not using the machine, an arrangement that has security implications). Besides, users may be keeping some data locally on removable media, such as diskettes.

What is required is a clear policy on local backup (as well as on the use of removable media). But how do you persuade users to do better in the backup department? Make it easier to do and make people want to do it. Making people want to do something is mainly a question of education. People need to be told why backups are important, and this means more than simply saying, "Because it is company policy." A positive approach is to educate, using scenarios in which backup saves the day. Users should be made aware of the variety of ways in which data can be lost or damaged. But don't dwell too long on the negative — emphasize the comfortable feeling that comes from knowing that you have current backups.

Making backup easy to do involves some decisions about hardware and software. What backup media will be used — floppy disks, tape, optical disks, cartridges? What backup software will be used? Will computers attached to a network be backed up independently or by the network? Will macros, batch files, or automated schedule programs be used to simplify the procedures? If so, who is responsible for creating and configuring these? Beyond these are questions such as how often backup should be done, what files should be backed up, and where will the backup media be stored? You should establish explicit guidelines on these matters so that users are clear about what their backup responsibilities are. Such rules and regulations can be incorporated into an education campaign. To summarize, a general improvement in backup habits is likely to occur if you:

1. Make backup a policy, not an option.
2. Make backup desirable.
3. Make backup easy.
4. Make backup mandatory.
5. Make sure users comply with backup policy.

Backup Strategy

There is no universal path to quick and easy backup. If there was, everyone would be taking it and cheerfully doing their daily backup. The user with unlimited resources has some excellent options, the most attractive probably being optical disks. But the whole culture of personal computers is shaped by economics and the inescapable fact is that most individuals and organizations do not have unlimited resources. To make effective use of time and money devoted to backup, a backup strategy should be developed. Consider what files need to be backed up, and how often the backup

should be performed. Begin by considering the type of backup that is needed.

Image Backup. Early personal computer tape drives could only perform a complete and total backup of every file on the hard disk, referred to as an image backup. This is a "warts and all" image, a track-by-track reading of the surface of the hard disk, including hidden and system files, even unused areas and cross-linked files. This caused problems when restoring data; for example, if the hard drive to which the data were being restored was not exactly the same make and model as the original. Some systems only allowed an image backup to be restored in its entirety, meaning that bad sectors were restored along with the good. But image backup has some advantages, such as speed. By treating the contents of the hard disk as a continuous stream of data bits, a lot of time that would otherwise be spent searching the disk for parts of specific files is saved. Recently, the use of image backup has been revived by more intelligent software that eliminates the shortcomings of early systems.

File-By-File. The alternative to an image backup is a file-by-file backup in which the user selects the directories and files to be backed up. The software then reads and writes each one in turn. While this may take longer than an image backup, it allows quick restoration of a single file or group of files. A file-by-file backup can also be faster than an image backup when only a small percentage of the hard disk has been used, or if the data on the hard disk are "optimized."[14] A file-by-file backup can be complete, including all of the files on the hard disk, but this is different from an image backup. In a file-by-file backup, the files are read individually rather than as a pattern on the disk.

Data Vs. Disk. When choosing the files to include in a backup, there is some logic in omitting program files because these already exist on the original program distribution disk(s). However, a fully functioning personal computer is constantly changing. Software is fine-tuned, utility programs are added, batch files and macros created, tool bars and icons are customized, and system files are tweaked for optimum performance. Recreating a system after a major crash involves a lot more than just copying back the data and reinstalling the programs. Numerous parameters, the right combinations of which were previously determined by considerable trial and error, need to be recreated. If you have no backup of configuration or user-preference files, getting the system back to normal can be quite a challenge. A good compromise is to make a complete backup at longer intervals, while backing up changing data files more frequently.

Now consider what you want to include when performing a data file backup. For example, are font files to be included? They seldom change but can take up a lot of space. You might want to omit them from a data file

backup. The same applies to spelling dictionaries and thesauri, which do not change. However, user-defined spelling supplements that are regularly updated might need to be included.

The method you use to include or exclude files from a backup operation will depend on the backup software you are using. For example, on the Macintosh, the operating system itself distinguishes between data/document files and program/application files, so backup software on the Mac often has a simple check box to include or exclude programs. Backup software on the PC often has include and exclude parameters based on file extensions. Program files can be excluded by specifying the extensions EXE and COM, plus BAT and SYS (as well as DLL on Windows systems). If you are consistent in your file naming, you might be able to group data files by specifying extensions such as DBF, XLS, DOC, and so on.

Incremental and Differential. An incremental backup involves backing up only those files that have changed since the last backup. The idea is that successive "all data files" backups are likely to include files that were already backed up. This slows down the backup process. Interim backups can be performed that only apply to files that have been added or modified since the last backup. Operating systems can do this by checking the status of files stored along with names and other directory information. Some backup software makes a distinction between incremental and differential backups; the latter is defined as all files that are new or modified since the last full backup. This differs from an incremental backup, which is all files that are new or modified since the last backup, either full or incremental.

Note that restoring from an incremental backup, as opposed to a full backup, may require more work. Several sets of media may be required, namely the previous full backup plus all incremental backups since then. On the other hand, restoring from a differential backup requires only the last full backup plus the last differential backup. However, differential backups take up more space and take longer to perform than incrementals. Basically, incrementals are better to systems that are heavily used, like file servers on a network, whereas differentials are more appropriate for single-user systems.

Backup Regimen

The timing of backups depends on how often the information on a system changes. A personal computer might operate purely as an information bank, perhaps used to look up pricing information that seldom changes — such a system only needs to be backed up when the information is updated. But a PC that records customer orders coming in as fast as they can be typed might have to be backed up at least once a day. Most systems are somewhere between these two extremes, but remember that frequency of file changes may not be a constant factor. For example, spreadsheets in the accounting department might change quite often while the annual budget

is being prepared, but remain unchanged the rest of the year. So, the backup regimen you implement will depend on how you use your computer. The three factors that need to be weighed against each other are:

- The amount of time and effort represented by changes to files.
- The amount of time and effort represented by backing up the files.
- The value of the contents of the files.

Careful consideration of work patterns is necessary to establish an appropriate backup regimen. You can combine the three levels of backup described earlier, based on three different intervals:

Interval 3	Total backup
Interval 2	Data file backup
Interval 1	Incremental data file backup

For example, you could do a total backup once a month, a total data file backup once a week, and an incremental data file backup every day. The main point is that every backup does not have to be complete or lengthy, and a schedule mixing complete and partial backups will require less time and so stand more chance of being adhered to. One important factor to bear in mind when designing your backup schedule is the ease with which the state of your data at a specific point in the past can be recreated. For example, suppose that a virus is discovered on a hard drive and many files have been infected. A process of deduction determines that the virus was probably introduced on Monday when an employee brought in a game on a floppy disk. If incremental backup is done daily with a full backup on Friday and today is Wednesday, then one option of dealing with the virus is to erase the hard disk and then restore the previous Friday's backup. Since viruses do not infect true data files you can then restore the data files from the Monday and Tuesday incremental backups.

But what if records were accidentally erased from a database on Tuesday, and this affected spreadsheets and reports created on Wednesday, yet the error was not discovered until the following Monday? You could not use the complete backup from the immediately preceding Friday to correct this problem. You would need the complete backup from the preceding Friday, plus the following Monday's incremental backup. If this sort of problem sounds challenging, that's because it is. Getting people to create backups is only part of the problem. Restoring systems and data from those backups is quite another.

Backup Handling and Storage

Consider the physical handling of the backup media. Where will it be stored? How many copies will there be? What makes a good off-site storage location? One possible media management program is to place backup copy 1 off-site (a bank, the manager's home, a different office of the same

company). Note that simply using a fireproof safe designed for important papers is not enough. Magnetic tapes give up the digital ghost at much lower temperatures than paper ignites — you want a safe that prevents internal temperature from rising above 125°F for at least 1 hour during exposure to fire at 1500°F. After a suitable interval you make backup copy 2, which is placed off-site, while backup 1 moves to on-site storage. After another interval, you reuse the backup 1 media to make backup 3, which is placed off-site while backup 2 is moved on-site. This means the off-site backup is always the most up-to-date.

For data-intensive operations, such as order processing where large amounts of data are added or altered every day, you can use a day-by-day backup schedule such as the six-way system. You begin by labeling six sets of media as Friday1, Friday2, Monday, Tuesday, Wednesday, and Thursday. On Friday afternoon, the operator goes to the backup storage cabinet and takes out the media marked Friday1. This is used to make a complete backup of the hard disk. The media is locked away over the weekend. On Monday afternoon, the operator goes to the media cabinet and gets out media marked Monday. This is used to make an incremental backup, overwriting the previous data on the media. The same thing happens on Tuesday through Thursday. Incremental backups are made each day on media marked for that day of the week.

When Friday rolls around again, the Friday2 media is used for a new complete backup. On Monday the incremental backup is made onto the Monday media, and so on, until Friday comes around again and you overwrite Friday1 with another complete backup. This system gives you a maximum archive period of two weeks. For example, on Fridays before you perform the Friday backup you have the ability to restore data from one or two Friday's ago. On any day of the week you can restore things to the way they were on same day of the previous week.

This system has several advantages. The time required for an incremental backup is generally far less than that for a full backup, making the daily routine less burdensome. Nevertheless, if restoration is required, a full set of data can be put together. If you simply use the same backup media every day, this type of recovery is not possible. A variation of this six-way routine, sometimes referred to as the father/son backup cycle, requires eight sets of media with the additional ones being called Friday3 and Friday4 so that your archive goes back a whole month.

Yet another backup cycle is the ten-way or grandfather/father/son system. This covers 12 weeks and allows you to delete data from your hard disk and retrieve it up to 3 months later. A variation of this scheme involves removing some of the complete backups from circulation at regular intervals for archive purposes, for example, once a month or once a quarter.

One advantage of this is a gradual replacement of media, which have a natural tendency to wear out from repeated use.

Give some thought to the time of day that backups are performed. It seems natural to do the backup at the end of the day, then lock the media away or take it off-site. Because some backup systems, such as tape units, allow backups to be triggered automatically, some people leave systems on overnight and have the backup performed under software control. This minimizes inconvenience to users, and leaving systems running is not considered detrimental to their health or reliability (although monitors should be turned down or off). However, even if the hardware performs reliably, there is a problem because the backup is being performed during a period of high risk.

Theft of computers, tampering with files, or disasters such as fires can progress with less chance of detection during the night. An unsupervised overnight backup operation is no protection against these threats. Indeed, if the backup media sits in the computer until a human operator arrives in the morning, it can make a nice present to someone looking to steal data. Doing backup first thing in the morning might seem like the answer, but again, an overnight attack threatens a whole day's worth of work. Besides, backup operations tend to tie up processing time and thus prevent systems from being used, which can make backing up in the morning counterproductive. One solution available to companies with an evening shift is to have them perform the backup and lock up the media before leaving. Indeed, with larger networks it will be necessary to budget staff specifically for this task.

Remote Backup Strategies

Off-site storage of backups is a strong defense against two serious threats, physical theft and natural disaster. However, some off-site storage options pose practical or tactical problems. Requiring staff to take backup media home with them imposes a considerable burden of responsibility, and requires a high degree of trust. Most banks are not set up to receive magnetic media for safe deposit outside normal banking hours. Fortunately, numerous companies now specialize in off-site storage of media, such as Arcus Data Security, DataVault, and Safesite Records Management.

Safesite's SafeNet service provides off-site storage and rotation of file server backup tapes. Outgoing tapes are placed in foam shipping trays and air-freighted overnight to secure vaults where they are bar coded and stored in a halon-protected environment that is fully temperature and humidity controlled. You pay a weekly fee for this service. Other companies operate at a local level, offering daily pickup and delivery of backup media according to standard rotation schedules. This has the added benefit of reinforcing backup regimes.

One step beyond physical off-site collection and delivery of backup media is remote off-site backup. In other words, your computers are backed up automatically, over phone lines, to a remote location, a strategy known as televaulting. This not only provides protection against theft and natural disasters at your site, it also provides insurance against errors and failures in your normal on-site backup systems. A pioneer and leading supplier of this type of service is Minneapolis-based Rimage Corporation (while the company headquarters are in Minneapolis, all its eggs are not in one basket — Rimage operates backup sites in New York and Atlanta, plus one near Los Angeles and another near San Francisco).

DEFEATING VIRUSES AND OTHER MALICIOUS CODE

One of the most persistent threats to the confidentiality, integrity, and availability of data entrusted to desktop systems, is malicious code, the most common form of which is the virus. A computer virus is self-replicating code designed to spread from system to system. Thousands of different viruses have been identified, although only a few hundred are active. This is software which can erase files, bring down networks, and waste a lot of person power and processing time. There are several types of programs, besides viruses, that can be grouped together as malicious code, or MC, although each type poses a different threat to the integrity and availability of your data.

The Malicious Code Problem

Based on numerous studies it is possible to say that malicious code has caused billions of dollars worth of damage and disruption over the last five years.[15] Malicious code has affected everything from corporate mainframes and networks to computers in homes, schools, and universities. Despite impressive advances in defensive measures, malicious programs continue to pose a major threat to information security. A key member of IBM's antivirus team, Alan Fedeli, uses the following as simple, working definitions of the three main problems for PC and LAN users:

- Virus: a program which, when executed, can add itself to another program, without permission, and in such a way that the infected program, when executed, can add itself to still other programs.
- Worm: a program which copies itself into nodes in a network, without permission.
- Trojan horse: a program which masquerades as a legitimate program, but does something other than what was expected, (as in the deceptive wooden horse used by the Greek army to achieve the fall of Troy).

Note that while viruses and worms replicate themselves, Trojan horses do not. Viruses and worms both produce copies of themselves but worms do so without using host files as carriers.

A fourth category of malicious code, the logic bomb, has historically been associated with mainframe programs but can also appear in desktop and network applications. A logic bomb can be defined as dormant code, the activation of which is triggered by a predetermined time or event. For example, a logic bomb might start erasing data files when the system clock reaches a certain date or when the application has been loaded × number of times. In practice, these various elements can be combined, so that a virus could gain access to a system via a Trojan, then plant a logic bomb, which triggers a worm.

The practical objection to viruses and worms, Trojan horses, and logic bombs, is that no programmer, however smart, can write code that will run benignly on every computer it encounters. Commercial software developers like Microsoft, which spend millions on software development and testing, cannot create such code, even when an elaborate installation program is used. The number of hardware permutations alone is staggering (with 12 alternatives in 12 categories you get 8,916,100,448,256 possible combinations). Quite simply, you cannot write benign code which can insert itself unannounced into every system without causing problems for at least some of those systems.

About Viruses

According to Dr. Peter Tippett, President of the National Computer Security Association, even if virus code does not try to cause harm, "most of the damage that viruses cause, day in and day out, relates to the simple fact that contamination by them must be cleaned up. The problem is that unless you search through all the personal computers at your site, as well as all the diskettes at your site, you can have no assurance that you have found all copies of the virus that may have actually infected only four or five PCs. Since viruses are essentially invisible the engineer must actually go looking for them on all 1000 PCs and 35,000 diskettes in an average corporate computer site. And if even a single instance of the virus is missed, then other computers will eventually be reinfected and the whole clean-up process must start again."

Further light is shed by IBM's Al Fedeli who notes that "While viruses exhibit many other characteristic behaviors, such as causing pranks, changing or deleting files, displaying messages or screen effects, hiding from detection by changing or encrypting themselves, modifying programs and spreading are the necessary and sufficient conditions for a program to be considered a virus." The very act of modifying files means that the presence of a virus causes disruption to normal operation, in addition to which the virus program can be written to carry out a specific task, like playing a tune at a certain time every day. In a mix of metaphors, such a virus task is referred to as a payload, and the event that releases or invokes it is referred

to as a trigger. This might be a date or action, such as booting up the machine. Some payloads are very nasty, such as corrupting the file allocation table (FAT) on a disk and thus rendering files inaccessible.

A lot of viruses attack operating system files, meaning that they have the potential to disrupt a wide range of users. Other viruses attack a particular application. Consider the virus that attacks dBASE data files, stored with the DBF extension. The virus reverses the order of bytes in the file as it is written to disk. The virus reverses them back to normal when the file is retrieved, making the change transparent to the casual user. However, if the file is sent to an uninfected user, or if the virus is inadvertently removed from the host system, the data are left in a scrambled state.

Before moving on to Trojan horses, it is important to point out that although some people say there are thousands of viruses to worry about, as of early 1997, only a few hundred were "in the wild." This term is reserved for viruses that have actually infected someone, somewhere. It is important to distinguish this small number of "in the wild" viruses from the much larger number of "in the zoo" viruses. We use this term to describe a virus that has never been seen in a real-world situation (believe it or not, some people who write viruses send them to antivirus researchers, which is one reason the population of the zoo far outnumbers that of the wild).[16]

The Trojan Horse

According to Rosenberger and Greenberg, "Trojan horse is a generic term describing a set of computer instructions purposely hidden inside a program. Trojan horses tell programs to do things you don't expect them to do." The original Trojan horse held enemy soldiers in its belly who thus gained entrance to the fortified city of Troy. In computer terms, a seemingly legitimate program is loaded by the user, but at some point thereafter malicious code goes to work, possibly capturing password keystrokes or erasing data.

An example appeared in 1995 when someone started distributing a file described as PKZIP 3.0, the long-awaited update of PKZIP version 2.04g, an excellent file archiving tool. Naturally, since the purpose of PKZIP is to compress and decompress files, version 2.04g was distributed as a self-extracting file. That is, it was executed as a program at the DOS prompt. PKZIP 3.0 was also made available on bulletin boards as an executable file, but it was not a self-extracting archive. Instead it was a Trojan horse that attempted to execute the DELTREE and FORMAT commands. Although clumsily written, it sometimes worked and some people lost data (one defense against such programs is to rename, remove, or relocate potentially destructive commands like FORMAT and DELTREE).

The Worm

According to virus experts Rosenberger and Greenberg, a worm is similar to a Trojan horse, but there is no "gift" involved: "If the Trojans had left that wooden horse outside the city, they wouldn't have been attacked from inside the city. Worms, on the other hand, can bypass your defenses without having to deceive you into dropping your guard." The classic example is a program designed to spread itself by exploiting bugs in a network operating software, spreading parts of itself across many different computers that are connected into a network. The parts remain in touch with, or related to, each other, thus giving rise to the term *worm*, a segmented insect. Naturally, this has a disruptive effect on the host computers, eating up empty space in memory and storage, and wasting valuable processing time.

The best-known example is the Internet worm which consumed so much memory space and processor time that eventually several thousand computers ground to a halt (the Morris/Internet worm has been exhaustively analyzed and documented on the Web). More destructive worms might erase files. Even without malicious intent, communications on the network are likely to be disrupted by any worm as it attempts to grow from one area to another. Most people agree that a worm is typified by independent growth rather than modification of existing programs. The difference between a worm and a virus might be characterized by saying a virus reproduces, while a worm grows.

The Code Bomb

One of the oldest forms of malicious programming is the creation of dormant code that is later activated or triggered by specific circumstances. Typical triggers are events such as a particular date or a certain number of system starts. Stories abound of disgruntled programmers planting logic bombs to get back at employers deemed to have been unfair. Several logic bombs have been planted in order to extort money. You have to pay up or find the malicious code and remove it. The latter option can be extremely costly when the system is a large mainframe computer.

Defenses Against MC

The layered approach to security that we advocate can provide a head start in defending against malicious code. To briefly reiterate the elements of this layered approach, they are

- Access control
  - Site — controlling who can get near the system.
  - System — controlling who can use the system.
  - File — controlling who can use specific files.

- System support
  - Power — keeping supply of power clean and constant.
  - Backup — keeping copies of files current.

The three access control items provide positive protection against infection, while the last item under System Support, backup, allows you to recover from a virus attack. However, we now add a third layer of System Support, namely Vigilance — keeping tabs on what enters or attempts to enter the system. By exercising vigilance, users and administrators alike can prevent, or at least minimize, the effects of malicious programming. To be vigilant, users need to know what they are defending against. This means:

- General training in malicious code awareness.
- Constant updating of defenses to remain effective against a threat which continues to evolve.
- An ongoing program of security checking, review, and retraining.

In the case of the most prevalent malicious code threat, viruses, vigilance means:

- Knowing what viruses are, the methods of attack they use, and what constitutes a healthy regimen of computer operation and maintenance.
- The use of hardware and/or software that prevents or warns of virus attacks (typically, software of this type needs to be updated on a regular basis in order to remain effective).
- Hardware and software buying choices might be affected, with systems and programs that are more inherently virus-free being preferred.

Staying Abreast

To be effective against malicious code you must keep abreast of the latest threats. Fortunately, this is now a lot easier than it used to be. There are a number of online sources that are sure to report new developments:

- NCSA forums on CompuServe
- NCSA pages on the Web
- Forum/Web page/BBS hosted by your antivirus vendor
- VIRUS-L news group

For the small/home office user we recommend checking in with one or more of these sources once a week. After all, it only takes a few minutes. For larger organizations we suggest that someone, probably on the support staff, be assigned the task of making a daily check.

Basic Rules

Being vigilant about the files that enter your system will go a long way towards protecting it from malicious code. If you use access controls to

extend that vigilance to the times when you are not around to oversee what is happening to your computer, you should avoid the immediate effects of malicious code attacks. To sum up the defensive measures discussed here, the following rules can be promulgated, first for the individual user, and then for the manager of users.

1. Observe site, system, and file access security procedures.
2. Always perform a backup before installing new software.
3. Only use reputable software from reputable sources.
4. Know the warning signs of a malicious program.
5. Use antivirus products to watch over your system.
6. Use an isolated machine to test software that might be suspect.

Rules for managers of users:

1. Make sure that access control and backup procedures are observed by all users.
2. Check all new software installations, floppy disks, and file transfers with an antivirus product.
3. Forbid the use of unchecked or unapproved software, floppy disks, or online connections.
4. Stay informed of latest developments in malicious programming, either through an alert service or by tasking in-house staff.
5. Keep all staff informed of latest trends in malicious code so that they know what to look for.
6. Make use of activity/operator logging systems so that you know who is using each system and what it is being used for.
7. Encourage the reporting of all operational anomalies and match these against known attacks.

Boot Sector Viruses

This type of infection hits your computer just as it loads the operating system. Most common on IBM-compatible machines, boot sector viruses can also be created for other systems (the "first" virus was an Apple II boot sector virus). Boot sectors are what get the operating system loaded into memory after you power-up the system (cold boot), or perform a hard reset (usually using a button on the front of the machine). On IBM-compatible machines, the instructions stored in the BIOS, which cannot themselves be infected by a virus since they are burned into ROM (Read Only Memory), load information from the Master Boot Sector and DOS Boot Sector into RAM, after performing the POST (Power On Self Test) and reading data, such as the time, from CMOS (which can be corrupted by viruses).

According to Virus Bulletin's description "boot sector viruses alter the code stored in either the Master Boot Sector or the DOS Boot Sector. Usually, the original contents of the boot sector are replaced by the virus

code…. Once loaded, the virus code generally loads the original boot code into memory and executes it, so that as far as the user is concerned, nothing is amiss." This might be accomplished by virus code in the boot sector that points to a different section of the disk. So the virus code is in memory and the user is none the wiser. The virus may then infect the boot sector of any floppy disk that is used in the machine's floppy disk drive, thus passing the infection on. While this is rather clever, it would seem to be an inefficient means of replicating now that so many people boot from a hard disk. If everyone cleaned their hard disk boot sector it would appear that extermination of boot sector viruses would be achievable.

Unfortunately, this overlooks the fact that there are boot sectors on ALL floppy disks, not just those that are bootable system disks. And we have all made the mistake of turning on or resetting a system with a floppy in drive A. If the floppy disk is not bootable, for example, if it is a data or program installation disk, we get the "Non-System disk or disk error. Replace and strike any key when ready" message. Alas, at that point the boot sector virus is already in memory. Indeed, that message is read onto the screen from the boot sector. Taking the floppy out and pressing "any key" will not clear the virus from memory, and besides, it may have already infected the hard disk. Note that the Macintosh uses a combination of hardware design and operating system software to spit out floppy disks when booting, thus considerably reducing the chances of this type of infection.

Even without the Mac's method of handling floppies, the solution appears quite simple: don't leave floppies in drive A, and if you do get the Non-System error message, reset the system instead of pressing "any key" when you get the message. Better still, if you have a newer BIOS that allows you to adjust the drive boot sequence, tell it to boot from C before A (this still allows you boot from a floppy if something happens to drive C). Well-known boot sector viruses include Michelangelo, Monkey.B, and perhaps the most widely occurring viruses of all time, Stoned and Form.

While at first it sounds like you could only catch a boot sector virus from a floppy disk, the threat is slightly more complex thanks to the folks who enjoy placing boot sector viruses in Trojan horse or "bait" files and then uploading them to bulletin boards. These files are designed to place the boot sector virus on your system when you execute them (ironically, these programs accomplish this task with a routine known as a "dropper," originally developed to allow the transfer of boot sector viruses between legitimate researchers and antivirus programmers).

Parasitic Viruses

More numerous than boot sector viruses but less prevalent, parasitic viruses are also referred to as file infectors, because they infect executable files. According to Virus Bulletin "they generally leave the contents of the

host program relatively unchanged, but append or prepend their code to the host, and divert execution flow so that the virus code is executed first. Once the virus code has finished its task, control is passed to the original program which, in most cases, executes normally." While such a complex operation sounds at first like it would be immediately noticeable to the user, this is often not the case since virus code is typically very compact. The temporary diversion of program flow is often indiscernible from normal operations.

Multipartite and Companion Viruses

You now know what boot sector and file infector viruses do. Put the two together and you have multipartite viruses, such as Tequila, which are capable of spreading by both methods. At the other end of the sophistication scale are companion viruses which take advantage of this simple fact about DOS: if you launch a program at the DOS prompt by entering its name, as in FORMAT, and DOS finds that there are two program files in the current directory, one called FORMAT.COM and the other called FORMAT.EXE, the COM file will be executed before the EXE file. A companion virus thus hides and spreads as a COM variant of a standard EXE file. Examples include the rare AIDS II and Clonewar viruses.

Other Types of Virus

Link viruses are a type of virus rare in the wild, despite the fact that they have considerable potential for spreading rapidly owing to the way they manipulate the directory structure of the media on which they are stored, pointing the operating system to virus code instead of legitimate programs. Academic viruses researchers and underground virus writers both spend a lot of time thinking about new ways in which viruses may be spread. This leads to many "in the zoo" or "in theory" viruses which exist more on paper than in practice. Several approaches to infection that fit into this category are source code and object code viruses. The idea behind a source code virus is to insert virus instructions into programs at the source code level, rather than through the compiled program.

A source code virus would add itself to the source code file, then get compiled into the executable file when the program code was compiled. From the complied program the virus code then seeks out further source code files to infect. This method of infection could be quite effective in some environments since most source code files have common and easily identifiable attributes, such as file extensions (like .C and .BAS). There is little evidence of such viruses on desktop machines, but widespread use of an interpreted language, like Microsoft Visual Basic, could make this an appealing path for infection.

To understand the object code virus, of which at least one example, Shifting_Objectives, has been discovered, you need to know that all of the source code for a complex program, such as Microsoft Windows or Microsoft Excel, is not compiled into one large EXE or COM file. Instead, these programs use sections of code, called objects, that are loaded into RAM and linked together only when they are needed. Programmers like to write code in the form of objects because these can be recycled very easily. For example, if treated as an object, the code required to create a dialog box can also be used in many places within a program, without the programmer having to code each dialog box individually. By infecting an object rather than an executable, the object code virus makes itself less open to normal methods of detection (for example, many antivirus strategies concentrate on protecting and monitoring executable files).

The term *kernel* is used to describe the core of the operating system. In DOS, for example, the kernel is stored in the hidden file IO.SYS. The idea behind a kernel infector, of which there are currently very few, is to operate at one level above the boot sector, but within the heart of the operating system, replacing the instructions in the real IO.SYS with its own agenda. This makes the virus more difficult to track than if it infected visible COM files such as COMMAND.COM. By loading its own code into memory ahead of the operating system the virus can achieve "stealthing" to avoid many traditional forms of virus detection.

Stealth and Polymorphism

Stealth viruses use traditional techniques for infection, such as boot sectors and executable files, but they have code which stays in memory to monitor and intercept operating system calls, thus disguising its presence. As Jonathan Wheat, one of the antivirus experts at NCSA puts it, "When the system seeks to open an infected file, the stealth virus leaps ahead, uninfects the file and allows the operating system to open it, so that all appears normal. When the operating system closes the file, the stealth virus reverses the actions, reinfecting the file. If you look at a boot sector on a disk infect by a stealth boot sector virus what you see looks normal, but it is not the real boot sector." Stealth viruses pose numerous problems for traditional antivirus products, which may even propagate the virus as they examine files when looking for infections.

The term *polymorphic* is used to describe computer viruses that mutate to escape detection by traditional antivirus software which compares suspect code to an inventory of known viruses. Polymorphic viruses can infect any type of host software. Polymorphic file viruses are most common, but polymorphic boot sector viruses have also been discovered (virus writers use a free piece of software called the Mutation Engine to transform simple

viruses into polymorphic ones, which ensures that polymorphic viruses are likely to further proliferate).

Some polymorphic viruses have a relatively limited number of variants or disguises, making them easier to identify. The Whale virus, for example, has 32 forms. Antivirus tools can detect these viruses by comparing them to an inventory of virus descriptions that allows for wildcard variations. Polymorphic viruses derived from tools such as the Mutation Engine are tougher to identify, because they can take any of four billion forms!

Macro Viruses

Viruses do not need to be written in assembly code or a higher language such as C. They can be written using any instruction set. Ask anyone who has worked with macros in programs such as 1-2-3 or Excel, WordPerfect, or Word, and you will discover that these work just like a programming language. As macros evolved from their origins in the 1970s in word processing (storing multiple keystrokes under one key) to spreadsheets in the early 1980s (enabling complex menu branches of conditional commands) they acquired a vital ingredient for virus making, automatic execution.

Of course, the purpose of automated operation was to enable the creation of easy-to-use, macro-driven applications for less-experienced users. In the mid to late 1980s this became a major activity within some organizations. Macro power increased, driven by power users of programs like 1-2-3 who worked hard to reduce complex operations, such as invoicing, to simple macro menus. Macros acquired the ability to execute operating system commands and further extended their power in the early 1990s when software designers introduced cross-application macro languages, such as WordBasic. The result is a class of computer file which appears at first to be a data file, but which may actually contain a program of macro commands.

This further blurred the distinction embodied in the oft-repeated advice that "your computer cannot be infected by a document" and "you can only be infected by programs." These statements only remain true if we carefully define documents to exclude those containing macros (and any other pseudo-language such as PostScript, which can trigger hardware events when transmitted to a printer) and define programs to include executable code in the widest sense (including ANSI codes, which could execute some unwanted actions if placed in e-mail that was displayed in text mode).

Ironically, Microsoft's domination of the software market in the mid 1990s provided the final ingredient for a "document" virus outbreak, that is, a universal, transplatform application — Microsoft Word. In late August of 1995 people learned that there was a dark side to the compatibility benefits of a *de facto* standard for word processing. A new virus came to light, capable of being spread through the exchange of Microsoft Word documents.

The virus, named Winword.Concept, replicates by adding internal macros to Word documents. If the virus is active on a system, an uninfected document can become infected simply by opening it and saving it using the "File Save As" menu option. Although Winword.Concept does not cause any intentional damage to the system, some users have reported problems when saving documents.

The macro virus becomes active when you open an infected document, doing so via Microsoft Word's "AutoOpen" macro, which executes each time you open a document. If you open an infected document with Word, the first thing the macro virus does is check the global document template, typically NORMAL.DOT, for the presence of either a macro named PayLoad or FileSaveAs. If either macro is found, the routine aborts and no infection of the global document template occurs. However, if these macros are not found, then several macros are copied to your global document template. During the course of copying the macros a small dialog box with an "OK" button appears on the screen. The dialog box simply contains the number "1" as its only text. The title bar of the dialog box indicates it is a Microsoft Word dialog box. This dialog will only be shown during the initial infection.

Once these macros are added to the global document template, they replicate by means of the virus version of "File Save" command. Consequently any document created using File Save As will contain this macro virus. An uninfected user can simply open the document and become infected. This can even happen while you are online to the World Wide Web, if you have your Web browser configured to use Word as the viewer for DOC files (the remedy is to use a viewer program such as Word Viewer, instead, as described later in this chapter). Note that the "PayLoad" macro contains the following text:

```
Sub MAIN

        REM That's enough to prove my point

End Sub
```

However, "PayLoad" is not executed at any time. Because of the flexibility of Microsoft's WordBasic macro language, almost anything could be performed here (including a file delete or other potentially damaging operating system commands). Also note that Word is available in many different languages, and in some versions the macro language commands have also been translated. This has the effect that macros written with the English version of Word will not work in, for example, the Finnish version of Word. The result is that users of such a national version of Word will not get infected by this virus. However, using an infected document in a translated version of Word will not produce any errors, and the infection will stay intact even if the document is re-saved. Under these circumstances

you should check for the presence of the virus in any case, in order not to spread infected DOC files further.

There are some preventative measures built into Word that are supposed to control automatic macros. For example, the Word for Windows manual states that if you hold down Shift while double-clicking the Word icon in Program Manager, then Word will start up with file-related "autoexecute" macros disabled. However, while this ought to inhibit the actuation of some macro viruses like WinWord.Nuclear, which relies on this feature, many users have found that it doesn't work. They also found that starting up Word with the command line WINWORD.EXE/m, which is supposed to achieve a similar effect, failed as well, as did holding down Shift while opening a document to disable any automatic macros in that file. Furthermore, many companies have invested a lot of development time in automatic Word macros to automate routine tasks. The best strategy for preventing infection is thus to scan all incoming documents. All products that achieve the NCSA's antivirus certification (listed at www.ncsa.com) are capable of spotting macro viruses.

ACCESS CONTROLS AND ENCRYPTION

Earlier it was noted that access controls and encryption are a defense against the compromise of data on stolen systems and storage media. For example, if a laptop system is stolen but the bulk of the data on the machine are stored in encrypted files, it is unlikely that the thief, or the person to whom the machine is fenced and ultimately sold, will gain access to the data.

Unfortunately, encryption is an example of security's two-edged sword. For example, the very feature that makes a notebook easier to secure physically (the small size — it can be locked away in an office drawer or a hotel-room safe) also makes it easier to run off with. Similarly, the technology that renders files inaccessible to the wrong people, encryption, can be abused to deny access to legitimate users (in the last 12 months we have received several calls from companies wanting help in retrieving their own data, encrypted by a disgruntled employee who refuses to share the password — payment is sometimes demanded, leading to the term *data ransoming*).

Nevertheless, it is better to use the digital protection schemes that are available than risk data loss or compromise. Start with the BIOS. Most laptops and desktops produced in recent years have a decent set of BIOS-based security features. For example, the trusty three-year-old Compaq Concerto on which this chapter is being written allows the user to "hot lock" with a single keystroke, preventing anyone from using the mouse or keyboard unless they can enter the correct PIN. This can be set to kick in at system startup, thus defending against a reboot attack. Beyond this, you

can disable the floppy drive, even block the ports, and all with a security program that has a Windows interface. Getting around this protection would require taking the machine apart and knowing just how to drain current from the CMOS.

Beyond BIOS-based protection you have the option of installing encryption software to scramble the contents of files so that they are useless to anyone who doesn't have the password/key. Encryption programs can operate at different levels. You can choose to encrypt just a few very valuable files on a file-by-file basis. This is simple and straightforward with something like Nortel Entrust Lite, McAfee's PC Secure, RSA's SecurPC, or Cobweb Application's KeyRing. These programs are particularly useful when you want to transmit files by e-mail, which remote users often need to do. If you routinely need to encrypt your e-mail messages, as opposed to file attachments, then PGPMail or ConnectSoft's Email Connection may be the way to go (the latter supports the S/MIME standard and requires a password before you can even run the program).

The next level of encryption is a designated area on the hard disk, in which all files stored are automatically encrypted. This is possible with programs like Utimaco's Safe Guard Easy products, which perform on-the-fly encryption. In other words, encryption and decryption are made part of the normal file save and open process. This can be more convenient in that constant entering of passwords is not required, but then again, if the master password is compromised the attacker may gain access to more data than if each file had a separate password. Program's like Symantec's Norton Your Eyes Only can actually encrypt everything on the entire hard disk, if that is what you want to do.

If you do use encryption you will need to take passwords seriously. The use of a master password, which unlocks all files you have encrypted, can simplify this, but it also increases the amount you have riding on one single password. Separate passwords for each file presents a management problem. Then there is the dilemma of easy-to-remember passwords, like your name, being easy for interlopers to guess, vs. long, obscure, and hard to crack passwords that you are tempted to write down, and thus compromise, just because they are hard to remember.

Also, there is the temptation to use the same password in different situations, which can lead to compromise. For example, it is relatively easy to crack the standard Windows 95 screen-saver password. So, you shouldn't use the same password for the screen-saver that you use for network login or sensitive file encryption (alternatively, you can use a more powerful screen-saver, such as Cobweb Application's HideThat).

Several encryption solutions attempt to go beyond passwords. For example, Fischer International offers a hardware key that fits inside a

floppy disk drive. Companies like Chrysalis and Telequip make PCMCIA cards that not only store encryption keys but also perform encryption calculations, thus mitigating some of the performance hit that encryption can impose. Encryption programs like Entrust can store passwords on floppy disks, which allows them to be kept separate from the computer where the encrypted files are stored. Keep that in your pocket when you leave your laptop behind and at least you will know that nobody can get to your files, even if they steal your machine.

DEFENDING THE LAN

The first personal computer networks were installed in the mid 1980s, allowing users to share, for purposes of efficiency, productivity and cost-saving, their storage devices, printers, and software. Naturally, these networks started out small, hence the term local area network. They were often informal, employed by a group of users who knew and trusted each other, and so people paid little attention to the security implications of this new type of computing.

Peer-to-Peer Networks

Typical of this phase of networking is the peer-to-peer network, in which each computer on the network has an equal ability to make its resources available to all the others. Examples are Appletalk, standard on the Apple Macintosh since 1984, Microsoft Windows for Workgroups, and Novell Personal NetWare. Microsoft continues to provide peer-to-peer networking in Windows 95 and Windows NT Workstation. The ease with which users of peer-to-peer networks can share files and printers is both appealing and alarming.

If you work with a small group of trusted colleagues, this approach to networking can be both convenient and efficient. But as such networks grow, systems become harder to manage, and trust is spread thinner. Access is difficult to control, because the network operating system was not designed with control in mind. All connections between a peer-to-peer network and other systems, such as the Internet or a dial-up line for a remote user are a security threat. For example, unless specific and nonobvious precautions are taken, any machine on a Windows 95 peer-to-peer network which dials out to the Internet immediately creates a path by which any other system on the Internet can access your shared resources.[17]

Server-Based Networks

Novell's main Netware product has always been a server-based network operating system and this path was followed by IBM, and later Microsoft (in the form of Microsoft LAN Manager which has evolved into Windows NT Server). Note that PCs connected to a network file server as clients act as

work stations, not terminals. In other words, they do not give up their ability to locally input, process, store, and output. Furthermore, unless they are logged onto the network, the network cannot have any effect on their security, which has serious implications. For example, when a PC has been logged off, the network operating system cannot control access to directories on its hard drive or prevent the user running locally stored applications.

Similarly, the network file server may scan both server and client directories for malicious code, but it cannot scan clients when they are not clients, that is, when they are logged off. This means that viruses can still infect machines that are part of the network. When an infected local machine later logs onto the network, it can spread the virus to the server.

While it is typical for the network file server to require that only authorized users, with valid users name and passwords, be allowed to use network resources, the network itself cannot identify users who do not log on. Theft, destruction, or corruption of data that are stored locally on a client is thus entirely possible, unless additional controls are in place. However, some interesting variations are possible when PCs are networked. For example, it is possible to configure desktop machines so that they cannot be operated unless they are logged onto the network. This can be achieved by extending the BIOS-based security described earlier (other examples of enhanced BIOS include alerting the network if the PC is logged off or disconnected).

Network Computers

If access to local storage is also blocked at the BIOS level, or removed completely, then the desktop computer becomes a truly dedicated client, useless without its properly authenticated network connection. Of course, some might argue that the machine is no longer a "personal computer," but from a security perspective the response is likely to be "so what?" In fact, today's networking technology allows the network to provide users with their own server-based storage and their own customized applications and settings, without the need for local storage. This facilitates centralized management of security tasks such as backup, authentication, and malicious code scanning.

The personal computer (PC) is thus transformed into the network computer (NC), a reincarnation of the diskless work stations that flopped in the 1980s. Back then, server-based software was far less exciting than the code you could run on stand-alone desktop machines, which were first adopted by eager do-it-yourself programmers who were people with a natural aptitude for productive use of the technology. Now that more than 50% of the workers in America have to use a computer of some kind, there is less need for each one of those computers to be personally managed and controlled.

From a security and management perspective, the NC is clearly a step forward, a cost-effective one at that. It is not unreasonable to suggest that individuals who still need or want a truly personal computer can either use their own machine at home, or use a nonnetworked system at the office. In any event, organizations should not lose sight of the fact that the "personal" computers it provides to its employees are actually the property of the organization, which is free to control the manner in which they are used, particularly when some uses such as Web surfing can increase risks to valuable data, not to mention the negative impact on productivity.

Network Security Implications

Constant improvements in hardware and software enabled LANs to grow in size and power. By the early 1990s some LANs had evolved into mission-critical information systems. The security implications increased dramatically but, even when network managers have had time to think about these implications, they have often lacked the resources and tools with which to address them. Furthermore, because many of these PC-based networks resembled the familiar paradigm of a powerful central computer supporting numerous, less powerful machines, many people assumed that the security problems could be solved in familiar ways, such as (1) give users password protected network accounts and don't let anyone log onto the network unless they can supply a valid account name and password; and (2) perform regular backups.

In practice, (2) has been easier to achieve than (1), but in a typical LAN environment (2) offers less protection than you might expect. The reason is simple. As was noted earlier, desktop computers are computers, they are not terminals. A desktop computer runs its own operating system under local control, does its own processing, has its own storage and its own input and output capabilities. Of course, you can try and make a desktop computer emulate a terminal, but unless you turn it into a terminal it will still be a computer.

Of course, there are many positive reasons for increased intercomputer communications, such as:

- Cost savings from sharing resources
- Productivity gains from faster, better communications and information sharing.

There are also potential security benefits. Any serious network operating system, or NOS, contains security features, and every NOS is more mindful of security than the popular desktop operating systems. The centralized storage of information that comes with server-based networking makes that information easier to protect, at least in terms of backup.

But these gains come with risks attached. Connecting two computers opens up a new front for the attacker who can exploit the connection, either to get at the data being transferred, or to penetrate one or more of the connected systems. Simply put, establishing a connection between two or more computers means:

- More to lose.[18]
- More ways to lose it.

The increase in potential gains from a single successful penetration of security makes the connected computer a far more promising target for the attacker. You still have to worry about in-house interlopers, both the merely curious and the seriously fraudulent, as well as disgruntled employees for whom intercomputer connections are a target for belligerence. But you also need to consider outside hackers, both amateur and professional, who live and breathe intercomputer communications.[19] The security implications of networking personal computers can be assessed as two different factors:

- The multiplication factor: normal security problems associated with an unconnected computer system are multiplied by a factor, roughly equal to the number of computer systems connected together.
- The channel factor: a new security area created by opening up channels of communications between computer systems, providing access into a computer through one port or another.

Taken together the multiplication and channel factors create the unique set of security problems normally referred to as network security. However, the term "manifold security" might better describe the situation confronting those responsible for securing personal computers which need to communicate, because, despite the existence of a substantial body of knowledge that deals with the protection of networks of large computer systems, much of it cannot be applied directly to personal computers. There are major differences in design and application. Personal computers are rarely located in secure or controlled environments. Neither personal computer hardware, nor the operating systems that control it, offer much in the way of built-in access control, particularly when it comes to connections with other hardware.

The Multiplication Factor

The security of computers that are connected has to start with individual computer security. You cannot combine a number of insecure computers into a network and create a secure system from the top down (unless you remove all local storage and processing, which in effect reduces the personal computer to a dumb terminal). While the network operating system will provide security measures, these are defeated or weakened if the

individual systems are not secure. If someone has uncontrolled use of a PC connected to a network, they have an excellent platform from which to attack the network, not to mention data that have already been transferred from the network to your PC (after all, the whole point of client/server computing is to make valuable data available on the desktop).

Even if the network is securely configured it cannot protect the PC that is not logged on. This problem is not likely to disappear any time soon, given that the default as-delivered state of most PCs continues to be unlocked and unprotected. Consider Windows 95, the first major new desktop operating system in many years. It contains plenty of hooks to which network security features can be attached, but it offers no serious stand-alone security. The point is clear: intercomputer security begins with everything in the chapter so far, from boot protection to backups, theft prevention to power conditioning, access control to virus prevention. According to the layered approach that this book advocates, each computer connected to another must be

- Protected by site, system, and file access control.
- Supported by suitable power and data backup facilities.
- Watched over by a vigilant operator/administrator.

The multiplication factor implies that protecting two computers is at least twice as difficult at protecting one. For example, a network can actually increase the damage and disruption that a virus can cause. The potential fall-out from the errors, omissions, and malicious actions of individual users is magnified when they are network users. Typically, a higher degree of user supervision is required; however, this is not always forthcoming. Users accustomed to the freedom and independence of stand-alone computing may find it irksome to submit to the rules for network users.

The Channel Factor

In previous chapters, you have seen how the layered approach to security is built up. So far, the concern has been the protection of personal computers as separate entities, vulnerable to abuse by users putting information in or taking it out via disk, screen, and keyboard. The layered approach to stand-alone security can be summarized like this:

- Access control
  - Site — controlling who can get near the system.
  - System — controlling who can use the system.
  - File — controlling who can use specific files.
- System support
  - Power — keeping supply of power clean and constant.

- Backup — keeping copies of files current.
- Vigilance — keeping tabs on what enters and leaves the system.

This arrangement needs to be expanded whenever a computer system is connected to another system. Intercomputer connection opens a channel of communication between machines. This adds a third layer, channel protection, which can be divided into three areas:

- Channel control
- Channel verification
- Channel support

Channel Control

A connection between two computers is one more way for an attacker to steal, delete, and corrupt information, or otherwise undermine normal operations. To prevent a channel of communication from becoming an avenue of attack, you need to control who can:

- Open a channel.
- Use a channel.
- Close a channel.

Clearly the first step is to ensure that proper site and system access controls are in place. The next step is to decide who needs to use a particular channel and then restrict access to authorized users. In network terms, this might be a matter of using password-controlled log-on procedures, or two-part token authentication. Password protection can be used for mainframe connections as well. Most commercial online services require an account number and password for access, and these should be closely guarded. However, system access control should be particularly tight on all personal computers equipped with modems.

Channel Verification

To be on the safe side, you should think of a channel of communication as a path through enemy territory. Whatever passes along that route runs the risk of being ambushed. Secure communications involves ongoing verification of:

- The identity of users.
- The integrity of data.
- The integrity of the channel.

Users of a communication channel should be required to identify themselves, whether the connection is a network hookup, a modem, or a mainframe link. When you are on the receiving end of intercomputer communications, that is, acting as the host for users calling in, you need to

be able to verify the claimed identity. Network nodes need to be able to verify the legitimacy of packets received.

One of the most important requirements for secure communications between computers is verification of identity. On a local area network, this might mean that each user has an ID number and a password, both of which must be entered before log-in can be completed. Of course, entry of a valid ID number/password combination does not guarantee the identity of the person using them, but the network software will tell the administrator who claims to be using the system. In small sites, a tour of the LAN can provide visual verification of these claims. In large installations, where the administrator might not be expected to put a name to every face, assistance might be provided in the form of photo-ID tags or biometric controls.

When data are being transferred via a communications channel, they are subject to possible distortion, tampering, or theft. Verifying the integrity of the channel means making sure that this does not happen. Most communications software includes some form of error checking. At a rudimentary level, this can check that the amount of data received matches the amount transmitted. More sophisticated methods confirm details of the transmission.

Verifying the integrity of the channel also means making sure nobody is listening in, or preventing the theft of anything useful if someone is. This is best accomplished by encryption. You will need to assess the likelihood of anyone attempting to intercept or overhear your communications. If the risk is high enough, then you can encrypt important communications, using a variety of devices. Some software systems encrypt all network and telephone line traffic. Hardware encryption/decryption devices can be placed at each end of a communications link. Some of these are combined with data verification systems.

Channel Support

Intercomputer communications can only be established when a large number of different parameters are properly coordinated. Once established, communications need to be maintained. This requires a high degree of reliability in communications hardware and software. The need for reliability and protection centers on those components that serve more than one user, in proportion to the number of users served. For example, in a local area network where one personal computer is acting as a file server for others, disruption or failure of the server can have far greater consequences than the breakdown of a single personal computer working on its own. Once established, channels of communication must be supported, or else those tasks that depend upon them will be jeopardized.

Business Recovery for LANs and Desktop Systems

One of the biggest challenges facing information systems professionals today is the recovery of desktop/LAN-based systems following disasters such as fires and floods (for more about the topic of business continuity planning, see Domain 8). As noted earlier in this chapter, a significant percentage of mission-critical applications are now running on desktop systems, which are inherently more complex when it comes to recovery. Unlike mainframe systems, which tend to conform to certain standards as far as equipment and code are concerned, and can thus be duplicated by a hot site with relative ease, each LAN represents a unique configuration of hardware and software.

The configuration of a particular LAN server, and the personal computer clients that it serves, may have been tweaked and fine-tuned over a long period of time. It is seldom possible to simply take the server backup tapes, load them onto a different server, and bring up the system. There are simply too many variables. There are some steps you can take to minimize these problems:

1. Carefully document the current LAN hardware and software, including all configuration settings.
2. Use "standard" equipment and configurations wherever possible.
3. Document the minimum configuration required to restore essential data and services on a replacement LAN.
4. Use server-mirroring, fault-tolerant hardware, and redundant disk arrays.

SECURE REMOTE ACCESS AND INTERNET CONNECTION

One of the most revolutionary, and largely unforeseen, implications of personal computer technology has been the emergence of the home office and the mobile worker. Invariably, users who are on the road need to call home, and so do their computers. Laptops like to link up with head office systems to update databases and download e-mail. A growing army of work-at-home telecommuters need some sort of remote access to their employer's systems. The technology with which to create these connections has been around for some time, and so has the subtle art of subverting it for nefarious purposes, or mere curiosity.

It might be hard to understand, but some people get a genuine thrill simply being "in" someone else's computer system. Remote access points are still a popular way of getting in. (Given the number of frustrating hurdles that you sometimes have to clear in order to establish a legitimate connection, it might be hard to imagine someone doing this for fun; however, at that precise moment when you finally get your own e-mail after hours of

dropped connections and redials, it is possible to sense something of the kick you get from hacking into someone else's system.)

Recent publicity about computer break-ins over the Internet has tended to overshadow hacking in through remote access points such as those provided for telecommuters, maintenance people, and field staff. However, this form of penetration is still used. Typically, it starts with a war dialer, a piece of software running on a modem-equipped PC, which automatically calls all of the phone numbers in a certain range, such as 345-0000, 345-0001 to 347-9999. The software records which numbers are answered by a modem. This gives the hacker a list of numbers worth testing for further access.

One technique that can reduce the risk of being found by such a technique is to set your modem to answer only after four or five rings — since the default operation of war dialers is geared toward speed, they may not linger that long at unanswered numbers. Of course, there are less technically sophisticated ways of getting phone numbers for computers, such as downloading lists of such numbers that are routinely shared on hacker bulletin boards, or digging through company trash for discarded phone directories.

Technically speaking you have several options for remote access. The most basic is a modem on your desktop machine which answers calls from the modem on your laptop. With "remote control" software running at both ends, the laptop user can operate the desktop machine as though seated at it. This remote control technology was popular early on in PC development since it kept to a minimum the data that needed to be sent over the phone at slow modem speeds. Later, when desktop machines were networked, the remote laptop user was able to control the desktop machine while it was logged into the network, thus giving network access.

With faster modems it became possible to log a remote caller directly into the network as a remote node. In other words, the laptop becomes a work station on the network. This is typically more convenient for the user, but it may be more expensive since the laptop needs to have its own licensed copy of the networked applications (instead of borrowing them from the desktop). However, network managers have tended to prefer remote node access because it is easier to manage, and this in turn provides security benefits. The remote machine has to prove its identity to the more demanding network server, rather than a mere desktop work station.

Recently, we have seen big strides towards consolidating remote network access, with special servers designed to run either remote node or remote control access in a tightly controlled manner. Typical methods for protecting a modem connection that is providing remote access are password protection and call-back. A simple form of the latter approach is for

the remote user to dial into the modem at the office, which then hangs up and calls the remote user back. The idea is to prevent people establishing connections from unauthorized numbers, but hackers have found that it is possible to fool the modem at the office into thinking it has dropped the connection, so that the call-back never really takes place. The addition of a password requirement at the time of call-back reduces the chances of this type of hack succeeding.

The call-back approach can be hard to scale when the number of remote users starts to grow, and the cost of long distance calls to all those users starts to add up. An alternative is to provide a toll-free number for remote users to dial into, which is answered by a remote access server. This is a combined hardware and software solution that creates a special node on the network with the ability to receive and authenticate multiple incoming calls. The connection should be authenticated by something stronger than an ordinary password, such as a one-time password generated by a smart card.

For example, modem-maker U.S. Robotics uses the SecurID system on its Total Control Enterprise Network Hub remote access server. To access the server the user enters a PIN followed by the code displayed on the SecurID card issued to that user. The code displayed on the card changes every 60 seconds, in sync with the company's ACE/Server authentication server at the office. Other options for two-factor authentication (something you know, like a PIN, plus something you have, like a token) include requiring special PCMCIA cards holding encrypted keys to be present in the remote laptop before the connection can be made.

The number of users who dial into the office is bound to increase as companies expand the use of telecommuting and virtual offices. This will continue to provide a possible channel for penetration of internal systems. But improvements in remote access servers supported by two-factor authentication systems have the potential to make such penetration increasingly difficult. Two developments that need to be watched carefully are the shift towards using the Internet for remote access to in-house databases, and public key-based digital certificates as a means of authentication.

SUMMARY

In less than two decades the microcomputer has risen from the basement workshop and the garage benchtop to become the dominant force in computer hardware. While mainframes and minicomputers continue to anchor many systems, particularly in areas such as online transaction processing, the shift towards client/server solutions based on what are, in essence, microcomputers, shows no signs of abating.

We are only just beginning to come to terms with the information security implications of this phenomenon.[20] The process starts with an understanding of the desktop computer environment. Experience has shown that you cannot simply take big-system security practices and impose them on desktop machines. We have to develop security policies and procedures that are appropriate for the desktop. We have to implement those policies and procedures by educating users about security. We might not like it, but the fact is personal computers will never be secure unless the personnel who use them also secure them.

There are alternative strategies. For example, you can emasculate the PC and make it an NC, controlled and secured by a server that is treated like a mainframe, even if it is just a beefed up PC. Whether this option will find favor, either in corporate information systems or cubicle-land, remains to be seen.

Footnotes

1. As someone you call when you get one of these headaches, I can attest to the increased frequency of the calls and the growing severity of the headaches. The opening comments in this chapter were shaped by participation in security assessments at a number of major U.S. and international corporations during the last 12 months. For a collection of recent infosec-related statistics, visit http://www.theroyfamily.com/security.html.
2. For more detailed statement of this position and its weaknesses, see *The NCSA Guide to PC and LAN Security*, McGraw-Hill, New York, 1996.
3. For example, many new PCs today have BIOS-based boot protection, but there are plenty still in use that do not.
4. Examples of this are legion, from Aldrich Ames, the CIA spy, to lists of AIDS patients made public in Florida, to company secrets valued at millions of dollars in cases brought by American Airlines and Merrill-Dow.
5. About 76% of survey respondents said they were running "mission critical" applications on local area networks. Ernst & Young survey of 1,271 technology and business executives, January, 1995.
6. For example, a modest 486 and a modem is all it takes to mount a very effective denial of service attack on a Web site, mail gateway, or even an Internet Service Provider such as the New York provider, PANIX, which was disrupted for more than a week in 1996.
7. "After 1998, the widespread availability of inexpensive disruptive technology and the broadening base of home computer users will put threat capabilities into the hands of a wider, less-privileged class, dramatically increasing the risk for intermediate-size organizations (0.8 probability)." Gartner Group.
8. For example, instructions for mounting the type of attack suffered by PANIX were posted on the Internet and recently an easy-to-use Windows attack program was released.
9. For example, it is relatively easy to configure a dumb terminal so that the screen is the only output device which is ideal for transitory lookup access to confidential data, such as medical records. But it is relatively difficult to lobotomize a PC so that it cannot retain or redirect whatever data it receives. I still meet mainframe-oriented systems people who have not yet grasped this distinction.
10. "Someone broke into the offices of Interactive Television Technologies, Inc. in Amherst, New York, and stole three computers containing the plans, schematics, diagrams and specifications for proprietary Internet access technology still in development but conservatively valued at $250 million." Reuters, 1996.
11. For example, case locks, building locks, increase surveillance.

12. A few years ago a manufacturer of data backup tapes, 3M Corp., did a survey about backup regimes and found that, of those respondents who regularly performed backups, some 80 percent only started to do so *after* they had lost data through lack of backup.
13. A tape jukebox can cycle through multiple tapes and backup RAID data that is mirrored and not being accessed.
14. The term "optimized" refers to organizing data on the disk so that files are stored in contiguous sectors, in logical order for the most efficient retrieval. The term "defragmented" is used to describe the process of rearranging files so that they are stored in contiguous sectors.
15. One of the most comprehensive studies is the one performed by NCSA, available at their Web site, www.ncsa.com.
16. A list of current "in the wild" viruses can be found at www.ncsa.com/virus/wildlist.html. The list is maintained independently for the computing community by Joe Wells, with the help of over 40 volunteers around the world.
17. For a test, point your Web browser to www.omna.com/yes/mwc/info, a page that tells you how your Windows 95 machine is configured.
18. A 1993 study by Infonetics Research of San Jose, California found that when companies experienced losses due to LAN outages, the average amount per company, including lost revenues and productivity, was $7.5 million.
19. Remember that hacker Kevin Mitnik's first arrest was for stealing manuals from a Pacific Bell switching station — that was in 1981, when he was 17.
20. See footnote 7.

Reflections on Database Integrity

William Hugh Murray

THIS CHAPTER DISCUSSES THE CONCEPT OF DATABASE INTEGRITY. It contrasts this concept to those of data integrity and database management system integrity. The purpose of the discussion is to arrive at a set of recommendations for the owners and operators of such databases on how to preserve that integrity.

CONCEPTS AND DESCRIPTIONS

This section sets forth some definitions and concepts that describe and bound the issue of database integrity.

Integrity

Integrity is the property of being whole, complete, and unimpaired; free from interference or contamination; unbroken; in agreement with requirements or expectation.

Data can be said to have integrity when it is internally consistent (e.g., the books are in balance) and when it describes what it intends (e.g., the books accurately reflect the performance and condition of the business). A system can be said to have integrity when it performs according to a complete specification most of the time, fails in a predictable manner, presents sufficient evidence of its failure to permit timely and effective corrective action, and permits orderly recovery.

Database

For purposes of this discussion, a database can be defined as a monolithic collection of related or interdependent data elements. Alternatively, it is a monolithic collection of information represented in coded data elements and specific relationships between those data elements. A database is usually intended to be shared across users, uses, or applications.

The abstraction of *database* is relatively novel, no older than the modern computer. Until the appearance of database management software for the microcomputer, perhaps a decade ago, it was esoteric. Analogous collections of data, such as the books of account for a business, existed before the computer. The term can properly be applied to most of the data that is usually recorded on such media as ledger cards or 3×5 cards. However, it is usually reserved for the most formal, rigorous, and systematic of such collections.

Information in a database can be explicitly represented in the form of coded data elements; employee name is a common example. However, there is other information in the database in the form of associations, both explicit and implicit, between the data elements.

Relationships are special kinds of associations between the data elements. For example, the various fields in an employee database record are related logically in much the same way as they are related on a piece of paper. The meaning and identity of each field is determined, in part, by this context. This information is at least as important as that in the data elements themselves.

The relationships can be expressed in the data itself (relational), in the arrangement or order of the elements within the database (structured), or in meta-data, data about the data, that explicitly describes or encodes the relationships (e.g., indexed or object oriented). While databases can be characterized by how the relationships are primarily expressed, in practice, all databases use a combination of these mechanisms. For example, in those databases known as *relational,* some relationships are expressed in the structure (i.e., tables and views), some in the data (i.e., references to other tables), and some in meta-data, the names of the columns.

Database Integrity

A database can be said to have integrity when it preserves the information in the data, that is, when both the data and the relationships are maintained. Database integrity is about the integrity of the records. The integrity of the database is separate from, and can be contrasted to that of the data, on the one hand, and of the database management system on the other.

Database Management System

For our purposes, a database management system is a generalized, abstract, and automated mechanism for creating, maintaining, storing, preserving, and presenting a database to, and on behalf of, applications.

Database managers are often characterized by the name of the mechanism on which they primarily rely to describe the relationships among the

data elements. Thus, database managers in which the relationship between two data elements is normally implied in the data itself, for example, the content of a data element (two employee records have the same department number), or the ordering of the data (employee A precedes B in the sort order of the name field) can be called *relational database managers*. Those in which the relationship is implied by how the two elements are physically stored, (for example, all employees in the same department are stored together, or employee A is always stored before B) can be referred to as *structured database managers*.

Relational Integrity

Relational integrity is the aspect of database integrity that deals with the preservation of the special relationships between the data elements.

Referential integrity is an example and a special case of relational integrity A reference is a relationship in which a value in one record points to another record, usually of another record type. For our purposes, it is an example and illustration of what it might mean to say that a database has integrity to the extent that relationships are preserved.

Consider the case of an employee record with a department number in it that refers to a department record. If the department number in the employee record is *N*, then referential integrity requires that there be a department record for department *N*. It would prohibit the creation of an employee record with a department number for which there was no corresponding department record, the deletion of department record *N* as long as any employee record pointed to it, and more than one department record *N* for the employee record to point to.

It should be noted that this kind of integrity is optional. That is, the condition could exist, coincidentally or accidentally, without any declaration, commitment, or enforcement. Likewise, it can be implemented and enforced either by using applications or the database management system. As a rule, it is preferable to have it implemented in the database management system so that the mechanism can be shared across applications and so that one using application need not rely on another.

METHODS

This chapter section discusses some of the methods for implementing database managers and preserving the integrity of the database.

Localization

By definition, a database is a monolith. That is, all of its elements and all of its relationships are essential to its identity. If any element or relationship is lost or broken, then the identity and the integrity are destroyed.

Of course, this is separate from the physical database manager, which might contain two or more independent databases. However, all other things being equal, keeping the elements of the database together helps preserve its integrity. Therefore, most database managers strive to keep the database together.

Single Owning Process

An important form of localization is the single owning process. Because a database is a monolith, there must be a single process that can see all of it, manage it, and have responsibility for its integrity. This owning process is usually the database manager. An implication is that a database manager is usually a single process.

Redundancy

To make the database more reliable than the media and devices on which it is stored, most database managers apply some kind of redundant data. The data is recorded in more than the minimum number of bits otherwise required to express it.

Dynamic Error Detection and Correction

Often, redundancy takes the form of error detection and correction codes. The data is recorded in codes that make the alteration of a bit obvious and its timely and automatic correction possible. One such code is *parity*, in which an additional bit is added to each frame of 7 or 8 bits to make the frame conform to some arbitrary rule such as *odd* or *even.* A variance from the rule signals the alteration of a bit. Some codes are so powerful as to permit the automatic detection and correction of multiple bit errors. These codes can be implemented in both the storage device (i.e., below the line) or in the database manager (above the line between software and hardware only mechanisms).

Duplication

Redundancy can be carried as far as one or more complete copies of the database or its elements. Such copies can be either inside or outside the database manager. Because relationships are usually best known to the database manager, they are best preserved using the duplication facilities that are provided by it.

Mirroring

One form of duplication is mirroring, in which two synchronized copies of the data are maintained. Mirroring is done internal to a mechanism; the copy is not visible from outside it. For example, a file manager can mirror files. It will apply changes to both copies, satisfy requests from either, but

conceal the existence of the second copy to processes outside itself. Mirroring can be done on the same device or on a different one. When done on a single device, mirroring protects against a media failure or a limited failure of the device (e.g., a bad track). When done across devices, it protects against a general device failure.

Backup

Backup copies of the database are made independent of the database manager. Among other losses, these copies are specifically intended to protect against damage that might occur to the data if the manager should fail or become corrupt.

Such copies can be prepared automatically by the database manager, or by using utilities or other program processes that are independent of the mechanism itself. Of course, although intended to protect against database manager failures, the use of an independent backup system may itself be a threat to the integrity of the database. It is difficult for an independent system to know and enforce the rules that the database manager itself enforces.

Checkpoints and Journals

A checkpoint is a special case of a backup copy. It is taken at a particular point in time. For example, the initial state of the database, even if empty, is a checkpoint. Checkpoints are used in conjunction with a journal or log of all update activity subsequent to the checkpoint to reconstruct the database. This mechanism preserves both integrity and currency.

Reconstruction

Such secondary copies can be employed to reconstruct the database, even from massive failures. However, this means that, at least under some circumstances, the integrity of the database will depend on the integrity of these copies.

Compartmentation

To compartmentalize is to place things into segregated compartments. The intent is to contain the effects of what happens in one compartment in such a way as to limit the impact on other compartments. For example, one might run multiple small database managers, in preference to a single large one, so as to limit the impact of a failure.

Segregation and Independence

Database management systems often implement segregation and independence of sub-processes to preserve integrity. For example, they may

isolate the process that does an update, from that which checks to see that it was done correctly, from the one that attempts corrective action. The purpose is to minimize the chances that the same fault will affect all three.

Encapsulation

The database manager can be viewed as a package, container, or capsule, one role of which is to protect the database from any outside interference or contamination. Encapsulation can be either physical or logical. For a database manager, physical encapsulation might be provided by placing it in a separate computer. Logical encapsulation might be provided by placing it in an isolated and protected process within an environment provided by a shared computer and its operating system. Logical encapsulation may also be provided, in part, and in static conditions, by the use of secret codes.

Most database management systems provide some encapsulation of the databases they contain. Object-oriented database management systems do so, by definition, explicitly and globally. Increasingly, one sees database managers themselves being encapsulated in their own hardware.

Hiding

Capsules hide or conceal their contents so that they cannot be seen or addressed from the outside. While this does not make the database safer from destruction, it does protect it from unauthorized disclosure and from malicious, but covert, change. Hiding can be implemented in many ways; the most common are by means of process-to-process isolation, data typing and type managers, and by the use of secret codes.

Binding

Binding is used to resolve and fix, for example, a data characteristic or reference, so as to resist later change. In computer science, one speaks of early and late binding. For example, in some programming, symbolic names are bound, that is, resolved so as to resist later change at compile time, while in others the same characteristic may not be bound until execution time.

Many structured database management systems can bind relationships in the database at programming time or at load time. This tends to improve both the integrity and performance at the expense of loss of flexibility and increased maintenance cost. Relational database managers also employ binding of table existence at creation time.

Binding applies only within the environment in which it takes place. If data or databases are removed from the database manager, then characteristics are no longer bound or reliable.

Atomic Update

Atomic update means that any change to the database takes place completely or not at all. There are no partial updates. This includes both data elements and relationships. Most database managers implement this by maintaining the ability to "roll back" any partial updates that they are unable to complete.

Locking

One potential threat to the integrity of a database results from concurrent use by two or more processes. For example, where two users make changes to a database, there is some potential that the second change will overwrite the first. Database management systems are expected to provide mechanisms, such as locking, that resist such problems.

Locking is a mechanism that database managers employ to ensure that partially updated elements and relationships are not used. It involves marking the element as "in use" or "asking for the lock" for all elements involved in an update. The mechanism will not permit a second use of an element that is in use and will not begin an update until is can obtain the locks for all elements involved. However, locking is ordinarily a logical, rather than physical mechanism. It is usually just a bit or flag that is set by locking or unlocking.

Locking may come in several levels of transparency and granularity. Ideally, locking would be automatic and transparent to all users or using processes. However, this might have unnecessary performance impact. For example, for maximum transparency, a database management system might restrict access from application B to any data that A is looking at, on the assumption that A might elect to update it. Thus, B will see a performance penalty even if he does not care about potential updates.

Performance might also require that B's access be limited to only the smallest element that A might update. B should not be restricted from an entire table simply because A is interested in a single row of the table. Thus, maximum performance requires that both A and B declare their intent.

Access Control

Access control is a mechanism provided by the database management system to enable the owners and managers of the database to control which users or using processes can alter the database, its elements, or

its relationships. These controls are most likely to be included in database management systems intended for use by multiple users. It is an integrity mechanism in that it reduces the size of the population that can alter the database to the intended population. It can also be used to enforce dual controls intended to resist errors and malice.

Privileged Controls

Most database management systems, particularly those that provide access controls, provide what can be referred to as privileged controls. These controls are intended for use by the managers of the system. They are intended for use to exercise ultimate control, particularly to remedy unusual situations. Two unusual situations are of particular interest. The first is to override the access controls. This capability may be necessary to avoid a deadlock situation. The second is the use of such privilege to repair the database itself. In the early days of structured databases, such controls were frequently used to "repair broken chains."

It should be noted that such privilege includes the ability to contaminate or interfere with the database.

Reconciliation

Reconciliation refers to an act or process that brings the database into harmony or consistency; that is, the act or process of checking the database against expectation and correcting for variances. Normally, database management systems perform this kind of checking on a routine, automatic, frequent, and repetitive, if not quite continuous, basis. For example, after making a WRITE request to another process (e.g., the file system), the database manager can make an immediate inspection to satisfy itself that the request completed correctly. The routine and automatic nature of this activity, among other things, distinguishes it from recovery. Another is that it relies almost exclusively on internal resources.

Recovery

Recovery is the integrity mechanism of last resort, the one that is used when the database is broken beyond the ability of any other mechanism to repair it. It is usually externally invoked and relies on external resources such as backup copies of the data. While it must bring the database back to a state of integrity, it may do so at the expense of currency or even lost data.

CONCLUSIONS

Database integrity is essential. If one cannot rely on the data, it is useless. Integrity is easier to preserve than to recreate. No single tool or

mechanism is sufficient unto itself. Database management systems will employ a variety of tools, and owners and managers will compensate for the inherent limitations of the database managers by employing tools that are completely external to it.

At least four things are necessary to preserve the integrity of a database:

1. One must preserve both the data elements and the relationships among them.
2. One must understand and exploit the mechanisms provided by the database management systems.
3. One must not compromise any of these mechanisms, either in the way one uses them or external to them.
4. One must understand the limitations of the database management system and compensate for them.

A simple copy of the data elements may not preserve the information contained in the relationships. For example, if a structured database contains information about the relationships in the physical location of the data within the device, then a copy of the data can preserve the relationships only if it is on an identical device.

Because all database management systems employ a combination of mechanisms to implement relationships and because most of these mechanism are concealed, management or operational procedures that bypass the database management system are suspect. On the other hand, if there are no measures taken to preserve integrity that are independent of the database management system, then a failure of the mechanism can destroy the database.

It should be noted that the most robust database managers so encapsulate the database that they cannot be bypassed. Any attempt to do so will result, at best, in the distortion of the database, and, at worst, in the destruction of the database and the database management system. Most of these systems will also provide one or more built-in mechanisms for creating external representations of the database.

One final issue is that of scale. Most databases are relatively small when compared to the systems and devices on which they reside. However, many of the most important databases are very large and span tens or even hundreds of devices. In such databases, information about relationships can span many devices. The integrity of the database requires the preservation of the devices and their relationship to each other.

On the other hand, it is common in these databases to create external copies by backing up the devices rather than the database or even the files. Such backups are device and device field dependent. While they provide adequate protection against the failure of one or two devices,

recovery from the destruction of the entire environment might require the complete replication of the environment. Timeliness may require that this be done in days or even hours. Thus, in exactly the databases in which it may be most urgent to have device-independent backups, it may be least likely to have them.

RECOMMENDATIONS

This chapter section sets forth recommendations for preserving the integrity of databases. These include some recommendations for using the database management system and some for compensating for its limitations.

1. Choose a database manager whose characteristics, features, and properties are sufficiently robust for the intended application and environment. Consider the size of the database and its importance to the enterprise.
2. Use the database management system according to directions. Note and respect all limitations.
3. Place the database and its manager in a robust environment.
4. Provide adequate resources (e.g., mirror files, devices, and control units) as indicated by the application and environment.
5. Prefer monolithic databases for integrity. Use distributed database managers only to the extent justified by major differences in performance.
6. For integrity, prefer a one-to-one relationship between a database, a database management system, and a processor. Share only to the extent indicated by major economies of scale. Keep in mind that today's computer systems can be more readily scaled to their applications. Large-scale sharing no longer offers the economies that it used to.
7. Prefer relational and object-oriented databases for integrity. Prefer structured databases for performance.
8. Applications and users should check those behaviors of the database manager that they rely on.
9. Limit access to the database and to elements within it to the minimum number of known users and processes consistent with the application.
10. Apply access controls in such a way as to involve multiple people in sensitive updates to the database.
11. Involve multiple people in the use of privileged or potent controls.
12. Keep multiple backup copies and generations of the data, including checkpoints and journals of update activity.
13. Prefer device-independent backups, particularly for databases that span multiple devices.

14. For device independence, prefer to make backups with services provided by the database manager. Use independent mechanisms for performance.
15. Prefer to make backups with services provided by the database manager for preservation of relationships. Prefer backups made by other means for independence and to protect against failure in the mechanism.
16. To protect external copies of the database, involve multiple people in their custody.
17. Check integrity after recovery and before use. Remember that even normal use of a corrupt database may spread the damage and that using bad data may result in serious damage to the enterprise.

121

Firewalls, Ten Percent of the Solution: A Security Architecture Primer

Chris Hare, CISSP, CISA

A solid security infrastructure consists of many components that, through proper application, can reduce the risk of information loss to the enterprise. This article examines the components of an information security architecture and why all the technology is required in today's enterprise.

A principal responsibility of the management team in any organization is the protection of enterprise assets. First and foremost, the organization must commit to securing and protecting its intellectual property. This intellectual property provides the organization's competitive advantage. When an enterprise loses that competitive advantage, it loses its reason for being an enterprise.

Second, management must make decisions about what its intellectual property is, who it wants to protect this property from, and why. These decisions form the basis for a series of security policies to fulfill the organization's information protection needs.

However, writing the policies is only part of the solution. In addition to developing the technical capability of implementing these policies, the organization must remain committed to these policies, and include regular security audits and other enforcement components into its operating plan. This is similar to installing a smoke alarm: if you do not check the batteries, how will you know it will work when you need it?

There are many reasons why a corporation should be interested in developing a security architecture including:

- Telecommunications fraud
- Internet hacking
- Viruses and malicious code
- War dialing and modem hacking
- Need for enhanced communications
- Globalization
- Cyber-terrorism
- Corporate espionage
- E-commerce and transaction-based Web sites

Telecommunications fraud and Internet and modem hacking are still at the top of the list for external methods of attacking an organization. Sources of attack are becoming more sophisticated and know no geographical limits. Consequently, global attacks are more predominant due to the increased growth in Internet connectivity and usage.

With business growth has come the need for enhanced communications. No longer is remote dial-up sufficient. Employees want and need high-speed Internet access, and other forms of services to get their jobs done, including videoconferencing, multimedia services, and voice conferencing. Complicating the problem

is that many corporate networks span the globe, and provide a highly feature-rich, highly connected environment for both their employees and for hackers.

The changes in network requirements and services has meant that corporations are more dependent on technologies that are easily intercepted, such as e-mail, audio conferencing, videoconferencing, cellular phones, remote access, and telecommuting. Employees want to access their e-mail and corporate resources through wireless devices, including their computers, cell phones, and personal digital assistants such as the PalmPilot and Research in Motion (RIM) BlackBerry.

With the Information Age, more and more of the corporation's knowledge and intellectual capital are being stored electronically. Information technology is even reported as an asset on the corporation's financial statements. Without the established and developed intellectual capital, which is often the distinguishing factor between competitors, the competitive advantage may be lost.

Unfortunately, the legal mechanisms are having difficulty dealing with this transnational problem, which affects the effectiveness and value of the legislation — expertise of law enforcement, investigators, and prosecutors alike. This legal ineffectiveness means that companies must be more diligent at protecting themselves because these legal deficiencies limit effective protection.

Add to this legal problem the often limited training and education investment made to maintain corporate security and investigative personnel in the legal and information technology areas. Frequently, the ability of the hacker far surpasses the ability of the investigator.

Considering the knowledge and operational advantages that a technology infrastructure provides, the answer is this: the corporation requires a security infrastructure because the business needs one.

Over the past 15 years, industry has experienced significant changes in the business environment. Organizations of all sizes are establishing and building new markets. Globalization has meant expanding corporate and public networks and computing facilities to support marketing, sales, and support staff. It addition to the geographical and time barriers, enterprises are continually faced with cultural, legal, language, and ethical issues never before considered.

In this time frame, we have also seen a drive toward electronic exchange of information with suppliers and customers, with E-commerce and transaction-based Web sites being the growth leaders in this area.

This very competitive environment has forced the enterprise to seek efficiencies to drive down product costs. The result of this activity has been to outsource noncore activities, legacy systems, consolidation of workforces, and a reduction in nonessential programs.

The mobile user community reflects the desire to get closer to our customer for improved responsiveness (e.g., automated sales force). In addition, legislation and the high cost of real estate have played a role in providing employees with the ability to work from home.

The result of these trends is that information is no longer controlled within the confines of the data center, thereby making it easier to get access to, and less likely that this access would be noticed.

Where Are the Risks?

The fact is that firewalls provide the perimeter security needed by today's organizations. However, left on their own, they provide little more than false assurance that the enterprise is protected. Indeed, many organizations believe the existence of a firewall at their perimeter is sufficient protection. It is not!

The number of risks in today's environment grows daily. There have been recent documented instances in which members of some of these areas, such as outsourced consultants, have demonstrated that they are more a risk than some organizations are prepared to handle. For example, *Information Week* has reported cases where outsourced consultants have injected viruses into the corporate network. A few of the many risks in today's environment include:

- Inter-enterprise networking with business partners and customers
- Outsourcing
- Development partners
- Globalization
- Open systems
- Access to business information
- Research and development activities

- Industrial and economic espionage
- Labor unrest
- Hacking
- Malicious code
- Inadvertent release or destruction of information
- Fraud

These are but a few of the risks to the enterprise the security architecture must contend with. Once the organization recognizes that the risk comes from both internal and external sources, the corporation can exert its forces into the development of technologies to protect its intellectual property.

As one legitimate user community after another have been added to the network, it is necessary to identify who can see what and provide a method of doing it. Most enterprises have taken measures to address many of the external exposures, such as hacking and inadvertent leaks, but the internal exposures, such as industrial or economic espionage, are far more complex to deal with. If a competitor really wants to obtain valuable information, it is easier and far more effective to plant someone in the organization or engage a business partner who knows where the information can be found.

Consider this: the FBI estimates that 1 out of every 700 employees is actively working against the company.

Establishing the Security Architecture

The architecture of the security infrastructure must be aligned with the enterprise security policy. If there is no security policy, there can be no security infrastructure. As security professionals, we can lead the best technologists to build the best and most secure infrastructure; however, if it fails to meet the business goals and objectives, we have failed. We are, after all, here to serve the interests of the enterprise — not the other way around.

The security architecture and resulting technology implementation must, at the very least, meet the following objectives:

- It must not impede the flow of authorized information or adversely affect user productivity.
- It must protect information at the point of entry into the enterprise.
- It must protect the information throughout its useful life.
- It must enforce common processes and practices throughout the enterprise.
- It must be modular to allow new technologies to replace existing ones with as little impact as possible.

Enterprises and their employees often see security as a business impediment. Consequently, they are circumvented in due course. For security measures to work effectively, they must be built into operating procedures and practices in such a way that they do not represent an "extra effort." From personal experience, this author has seen people spend up to ten times the effort and expense to avoid implementing security.

The moment the security infrastructure and technology are seen, *or perceived*, to impact information flow, system functionality, or efficiencies, they will be questioned and there will be those that will seek ways to avoid the process in the interest of saving time or effort. Consequently, the infrastructure must be effective, yet virtually transparent to the user.

Once data has entered the system, it must be assumed that it may be input to one or more processes. It is becoming impractical to control the use of all data elements at the system layer; therefore, any data that is considered sensitive, or can only be "seen" by a particular user community, must be appropriately protected at the point of entry to the network or system and, most importantly, wherever it is subsequently transferred. This involves the integration of security controls at all levels of the environment: the network, the system, the database, and the application.

A centralized security administration system facilitates numerous benefits, both in terms of efficiency and consistency. Perhaps the most significant advantage is knowing who has access to what and if, for whatever reason, access privileges are to be withdrawn, that can be accomplished for all systems expeditiously.

Quite clearly, it is not economically feasible to rewrite existing applications or replace existing systems. Therefore, an important aspect of the security architecture must be the ability to accommodate the existing infrastructure. Along the same lines of thinking, the size of existing systems and the population using them precludes a one-time deployment plan. A modular approach is an operational necessity.

The infrastructure resulting from the architecture must also provide specific services and meet additional objectives, including:

- Access controls
- Authorization
- Information classification
- Data integrity

Achieving these goals is not only desirable, it is possible with the technology that exists today. It is highly desirable to have one global user authentication and authorization system or process, a single encryption tool, and digital signature methodology that can be used consistently across the enterprise for all applications. Authenticating the user does not necessarily address the authorization criteria; it may prove that you are who you say you are but does not dictate what information can be accessed and what can be done with it.

Given the inter-enterprise electronic information exchange trend, one can no longer be certain that the data entering the corporate systems is properly protected and stored at the points of creation. Data that is submitted from unsecured areas represents a number of problems, primarily related to integrity, the potential for information to be modified (e.g., the possibility of the terminal device being "spoofed," collecting data, modifying it, and retransmitting it as if from the original device), and confidentiality (e.g., "shoulder surfing").

Unfortunately, one cannot ignore the impact of government in our infrastructure. In some way or another, domestic and foreign policies regarding what one can and cannot use do have an effect. Consider one of the major issues today being the use of encryption. The United States limits the export of encryption to a key length, whereas other governments (e.g., France) have strict rules regarding the use of encryption and when they require a copy of the encryption key.

In addition, governments also impose import and export restrictions on corporations to control the movement of technology to and from foreign countries. These import/export regulations are often difficult to deal with due to the generalities in the language the government uses, but they cannot be ignored. Doing so may result in the corporation not being able to trade with some countries, or lose its ability to operate.

An Infrastructure Model

The security infrastructure must be concerned with all aspects of the information, and the technology used to create and access it, including:

- Physical security for the enterprise and security devices
- Monitoring tools
- Public network connectivity
- Perimeter access controls
- Enterprise WAN and LAN
- Operating systems
- Applications
- Databases
- Data

This also does not discount the need for proper policies and an awareness program as discussed earlier. The protection objects listed above, if viewed in a reverse order (Exhibit 121.1), provides an outside in view to protecting the data.

What this model also does is incorporate the elements of physical security and awareness, including user training, which are often overlooked. Without the user community understanding what is expected from them in the security model, it will be difficult — if not impossible — to maintain.

The remainder of this article focuses on the technology components and how to bring them together in a sample architecture model.

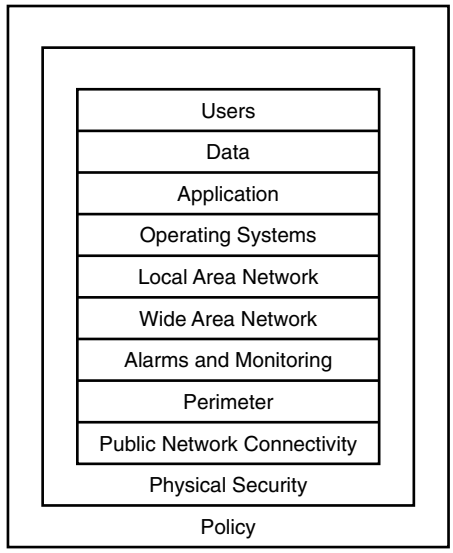


EXHIBIT 121.1 The Infrastructure Model.

Establishing the Perimeter

The 1980s brought the development of the microcomputer, and despite its cost, many enterprises that were mainframe oriented could now push the work throughout the enterprise on these lower-cost devices. Decentralization of the computing infrastructure brought several benefits and, consequently, several challenges.

As connectivity to the Internet increased, a new security model was developed. This consisted of a "moat," where the installation of a firewall provided protection against unauthorized access. Many organizations then, as today, took the approach that information contained within the network was available for any authorized employee to access. However, this open approach meant that the enterprise was dependent upon other technology such as network encryption devices to protect the information and infrastructure.

The consequence many organizations have witnessed with this model is that few internal applications and services made any attempt to operate in a secure fashion. As the number of external organizations connected to the enterprise network increases, the likelihood of the loss of intellectual property also increases.

With the knowledge that the corporate network and intellectual property were at risk, it was evident that a new infrastructure was required to address the external access and internal information security requirements.

Security professionals around the globe have embarked on new technology and combinations. Consequently, it is not uncommon for the network perimeter to include:

- Screening or filter routers
- Firewalls
- Protected external networks
- Intrusion detection systems

When assembled, the perimeter access point resembles the diagram in Exhibit 121.2.

The role of the screening or filter router between the external network and the firewall is to limit the types of traffic allowed through, thereby reducing the quantity of network traffic visible to the firewall. This establishes the first line of defense. The firewall can then respond more effectively to the traffic that is allowed through by the filter router. This first filter router performs the ingress traffic filtering, meaning it limits the traffic inbound to your network based on the filter rules.

Traditionally, enterprises have placed their external systems such as Web and FTP servers outside their firewall, which is typically known as the DMZ (demilitarized zone). However, placing the systems in this

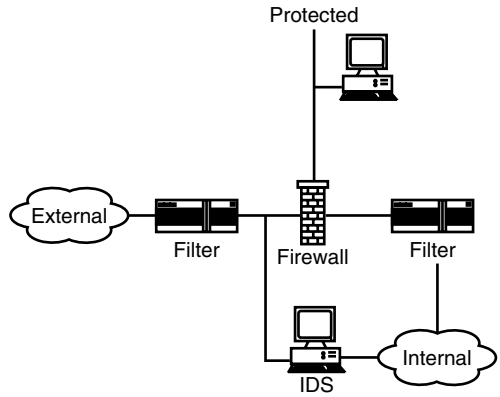


EXHIBIT 121.2 Perimeter Access Point.

manner exposes them to attack from the external network. An improved approach is to add additional networks to the firewall for these external systems. Doing so creates a protected network, commonly known as a service network or screened subnet.

The filters on the external filter router should be written to allow external connections to systems in the protected network, but only on the allowed service ports. For example, if there is a Web server in the protected network, the filter router can be designed to send all external connection requests to the Web server to only the Web server. This prevents any connections into the internal network due to an error on the firewall.

Note: The over use of filters on routers can impact the overall performance of the device, increasing the time it takes to move a packet from one network to another. For example, adding a single rule: <any IP address> to <any IP address> adds ten percent to the processing load on the router CPU. Consequently, router filter rules, although recommended, must be carefully engineered to not impede network performance.

The firewall is used to create the screened or protected subnet. A screened subnet allows traffic from the external network into the screened subnet, but not directly into the corporate network. Additionally, firewall rules are also used to further limit the types of traffic allowed into the screened subnet, or into the internal network.

Should a system in the protected network require access into the internal network, the firewall provides the rules to do so, and limits the protocols or services available into the internal network.

The second filter router between the firewall and the internal network is used to limit outbound traffic to the external network. This is particularly important to prevent network auto-discovery systems such as HP Openview from trying to use its auto-discovery features to map the entire Internet. This filter router can also allow other traffic that the enterprise does not want sent out to the Internet to be blocked. This is egress filtering, or using the router to limit the traffic types being sent to the external network. Some enterprises combine both filters on one router, which is acceptable depending on the ultimate architecture implemented.

The final component is an intrusion detection system (IDS) to identify connection attempts or other unauthorized events and information. Additionally, content filtering systems can be used to scan for undesirable content in various protocols such as Web and e-mail. Many vendors offer solutions for both, including those that can prevent the distribution of specific types of attachments in e-mail messages. E-mail attachment scanning should also be implemented in the enterprise to prevent the distribution of attachments such as malicious code within the enterprise.

The Network Layer

The network layer addresses connectivity between one user, or system, and another for the purposes of information exchange. In this context, information may be in the form of data, image, or sound and may be transmitted using copper, fiber, or wireless technologies. This layer will include specific measures to address intra- and inter-enterprise information containment controls, the use of private or public services, protocols, etc.

Almost all enterprises will have some level of connectivity with a public data network, be it the Internet or other value-added networks. The security professional must not forget to examine all network access points and connectivity with the external network points and determine what level of protection is needed. At very least, a screening router must be used. However, in some cases, external legislation determines what network access control devices are used and where they must be located.

The enterprise wide area network (WAN) is used to provide communications between offices and enterprise sites. Few enterprises actually maintain the WAN using a leased line approach due to the sheer cost of the service and associated management. Typically, WAN services are utilized through public ATM or Frame Relay networks. Although these are operated and managed by the public telecommunications providers, the connectivity is private due to the nature of the ATM and Frame Relay services.

Finally, the local area network (LAN) used within each office provides network connectivity to each desktop and workstation within the enterprise. Each office or LAN can be used to segregate users and departments through security domains (see Exhibit 121.3).

In this case, the security professional works with the network engineering teams to provide the best location for firewalls and other network access devices such as additional filter routers. Utilizing this approach can prevent sensitive traffic from traveling throughout the network and only be visible to the users who require it. Additionally, if the information in the security domain requires it, network and host-based IDSs should be used to track and investigate events in this domain.

Finally, the security professional should recommend the use of a switched network if a shared media such as coaxial or twisted-pair media is used. Traditional shared media networks allow any system on the network to see all network traffic. This makes it very easy for a sniffer to be placed on the network and packets collected, including password and sensitive application data. Use of a switched network makes it much more difficult, although not impossible.

Other controls should be used in the design of the LAN. If the enterprise is using DHCP, any person who connects to the LAN and obtains an IP address can gain access to the enterprise network. For large enterprises, it is unrealistic to attempt to implement MAC level controls due to the size of the network. However, public areas such as lobbies and conference rooms should be set up in one of the following manners:

- No live network jacks
- DHCP on a separate subnet and security domain
- Filtered traffic

The intent of these controls is to prevent a computer in a conference room from being able to participate fully on the network, and only offer limited services. In this context, security domains can be configured to specifically prevent access to other parts of the network or specific systems based on the source IP address.

Other LAN-based controls for network analysis and reporting, such as Nicksun Probe and NetVCR, provide network diagnostics, investigation, and forensics information. However, on large, busy networks, these provide an additional challenge, that being the disk space to store the information for later analysis.

Each of the foregoing layers provides the capability to monitor activities within that layer. Monitoring systems will be capable of collecting information from one or more layers, which will trigger alarm mechanisms when certain undesirable operational or security criteria are met. The alarm and monitoring tools layer will include such things as event logging, system usage, exception reporting, and clock synchronization.

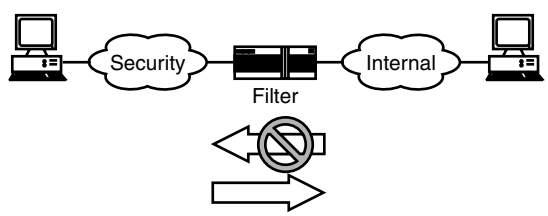


EXHIBIT 121.3 Local Area Network with Security Domains.

Physical Security

Physical security pertains to all practices, procedures, and measures relating to the operating environment, the movement of people, equipment or goods, building access, wiring, system hardware, etc. Physical security elements are used to ensure that the corporate assets are not subjected to unwarranted security risks. Items addressed at this layer include secure areas, security of equipment off-premises, movement of equipment, and secure disposal of equipment.

The physical security of the following network access control devices, is paramount to ensuring the ongoing protection of the network and enterprise data:

- Firewall
- IDS
- Filter routers
- Hubs
- Switches
- Cabling and
- Security systems

Should these systems not be adequately protected, a device could be installed and no one would notice. Physical security controls for these devices should include locked cabinets and cable conduits, to name only two.

System Controls

Beyond the network are the systems and applications that users use on a daily basis to fulfill enterprise business objectives. The protection of the operating system, the application proper, and the data are just as important as the network.

Fundamentally, information security is in the hands of the users. Regardless of the measures that may be implemented, carelessness on the part of individuals involved in the preparation, consolidation, processing, recording, or movement of information can compromise any or all security measures. This layer then looks at the human-related processes, procedures, and knowledge related to developing a secure environment, such as user training, information security training and awareness, and security policies and procedures.

Access to the environment must be controlled through a coordinated access control program, as discussed later in this article. Access control provides the control mechanisms to limit access to systems, applications, data, or services to authorized people or systems. It includes, for example, identification of the user, their authorization, and security practices and procedures. Examples of items that would be included in access control systems include identification and authentication methods, privilege management, and user registration. One could argue that privilege management is part of authorization; however, it should be closely coupled to the authentication system.

The operating system controls provide the functionality for applications to be executed and management of system peripheral units, including connectivity to network facilities. A heterogeneous computing environment cannot be considered homogeneous from a security perspective because each manufacturer has addressed the various security issues in a different manner. However, within your architecture, the security professional should establish consistent operating system baselines and configurations to maintain the overall environment.

Just as the security professional will likely install a network-based intrusion detection system, so too should host-based systems be considered for the enterprise's critical systems and data. Adding the host-based element provides the security professional with the ability to monitor for specific events on the system itself that may not be monitored by or captured through a network-based intrusion detection system.

The data aspect of the architecture addresses the measures taken to ensure data origination authenticity, integrity, availability, non-repudiation, and confidentiality. This layer will address such things as database management, data movement and storage, backup and recovery, and encryption. Depending on the applications in use, a lot of data is moved between applications. These data transfers, or interfaces, must be developed appropriately to ensure that there is little possibility for data compromise or loss while in transit.

The application and services layer addresses the controls required to ensure the proper management of information processing, including inputs and outputs, and the provision of published information exchange services.

Establishing the Program

The security architecture must not only include the elements discussed so far, but also extend into all areas to provide an infrastructure providing protection from the perimeter to the data. This is accomplished by linking security application and components in a tightly integrated structure to implement a security control infrastructure (see Exhibit 121.4).

The security control infrastructure includes security tools and processes that sit between the application and the network. The security control infrastructure augments or, ideally, replaces some of the control features in the applications — mostly user authentication. This means that the application does not maintain its own view of authentication, but relies on the security control infrastructure to perform the authentication. The result is that the user can authenticate once, and let the security control infrastructure take over. This allows for the eventual implementation of a single sign-on capability.

A centralized tool for the management of individual user and process privileges is required to enable the security control infrastructure to achieve this goal. The centralized user management services interact with the control infrastructure to determine what the user is allowed to do. Control infrastructure and other services within it depend on the existence of an enterprisewide privilege database containing the access and application rights for every user.

The result is a security infrastructure that has the ability to deliver encryption, strong authentication, and a corporate directory with the ability to add single sign-on and advanced privilege management in the future.

The Corporate Directory

The corporate directory, which is a component of the security control infrastructure, contains elements such as:

- Employee number, name, department, and other contact information
- Organizational information such as the employee's manager and reporting structure
- Systems assigned to the employee
- User account data
- E-mail addresses
- Authorized application access
- Application privileges
- Authentication information, including method, passwords, and access history
- Encryption keys

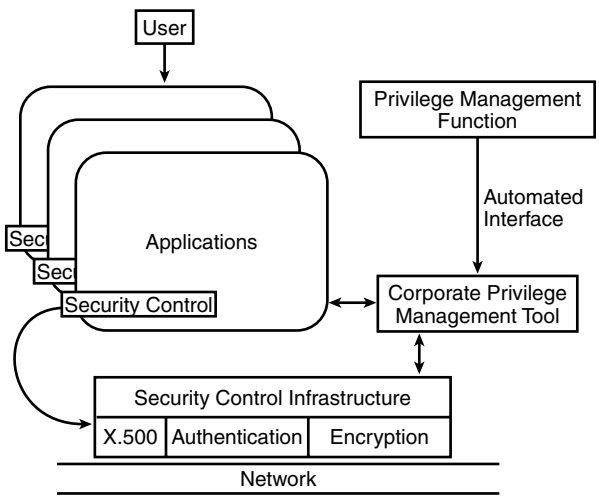


EXHIBIT 121.4 Security Control Infrastructure.

All of this information is managed through the enterprise user and privilege management system to provide authentication information for network, system, and application access on a per-user basis (see Exhibit 121.5).

With the wide array of directory products available today, most enterprises will not have to develop their own technology, but are best served using X.500 directory services as they provide Lightweight Directory Access Protocol (LDAP) services that can be used by many of today's operating systems, including Windows 2000.

The enterprise directory can be used to provide the necessary details for environments that cannot access the directory directly, such as NIS and non-LDAP-ready Kerberos implementations.

Using the enterprise privilege management applications, a new user can be added in a few minutes, with all the necessary services configured. New applications and services can be added at any time. Should an employee no longer require access to specific applications or application privileges, the same tool can be used to remove them from the enterprise directory, and subsequently the application itself.

A major challenge for many enterprises is removing user access when that user's employment ends. The enterprise directory removes this problem because the information can be removed or invalidated within the directory, thereby preventing the possibility of the employee's access remaining active and exposing the company beyond the user's final day of work.

Authentication Systems

There are many different identification and authentication systems available, including passwords, secure tokens, biometrics, and Kerberos to name a few (see Exhibit 121.6). The enterprise must ultimately decide what authentication method makes sense for its own business needs, and may require multiple systems for different information types within the enterprise. However, the common thread is that in today's environment, the simple password is just not good enough anymore.

When a user authenticates to a system or application, his credentials are validated against the enterprise directory, which then makes the decision to allow or deny the user's access request. The directory can also provide authorization information to the requesting application, thereby limiting the access rights for that user. Using this methodology, the exact authentication method is irrelevant and could be changed at any time. For example, using a password today could be replaced with a secure token, biometrics, or Kerberos at any time, and multiple authentication technologies can easily co-exist within the enterprise.

However, one must bear in mind that user authentication is only one aspect. A second aspect concerns authentication of the information. This is achieved through the use of a digital signature, which provides the authentication and integrity of the original message.

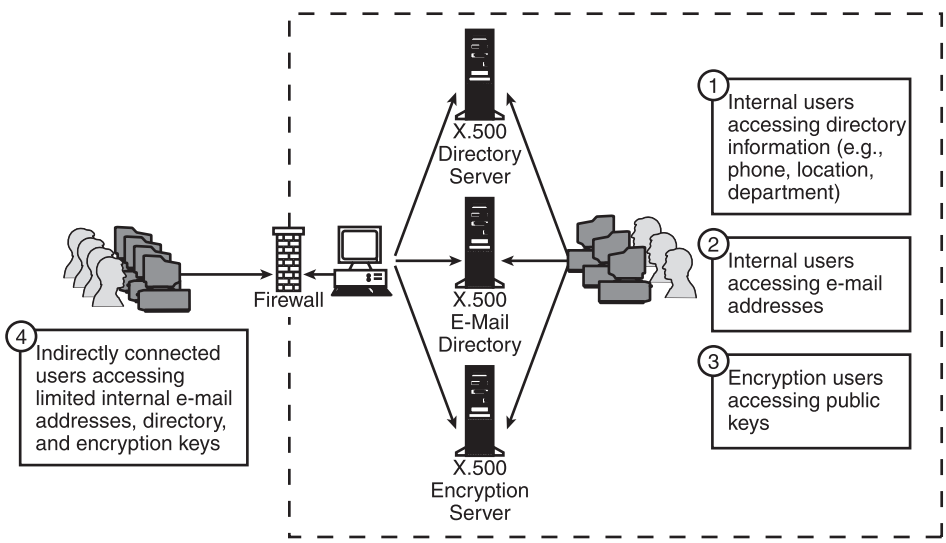


EXHIBIT 121.5 Authentication Information for Network, System, and Application Access.

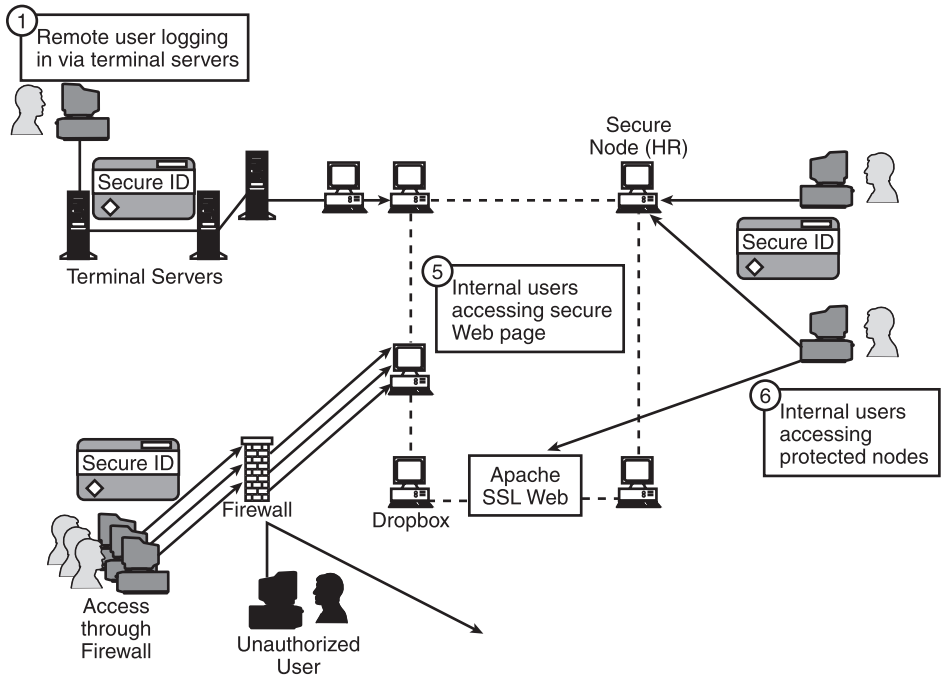


EXHIBIT 121.6 Authentication Systems.

It is important to remember that no authentication method is perfect. As security professionals, we can only work to establish even greater levels of trust to the authenticating users.

Encryption Services

Encryption is currently the only way to ensure the confidentiality of electronic information. In today's business environment, the protection of enterprise and strategic information has become a necessity. Consequently, the infrastructure requirements include encryption and digital signatures (see Exhibit 121.7).

Encryption of files before sending them over the Internet is essential, given the amount of business and intellectual property stolen over the Internet each year. The infrastructure must provide for key management, as well as the ability to handle keys of varying size. For example, global companies may require key management abilities for multiple key sizes.

Encryption of enterprise information may be required within applications. However, without a common application-based encryption method, this is difficult to achieve. Through the use of virtual private network (VPN) technologies, however, one can construct a VPN within the enterprise network for the protection of specific information, regardless of the underlying network technologies. Virtual private networking is also a critical service when sessions are carried over insecure networks such as the Internet.

In addition, the mobile user community must be able to protect the integrity and confidentiality of its data in the event a computer is stolen. This level of protection is accomplished with more than encryption, such as disk and system locking tools.

Customer and Business Partner Access

The use of the security infrastructure allows for the creation of secure environments for information exchange. One such example is the customer access network (see Exhibit 121.8) or those entry points where nonenterprise employees such as customers and suppliers can access the enterprise network and specific resources. In our global community, the number of networks being connected every day continues to grow. However, connecting one's corporate network to "theirs" also exposes one to all of the other networks "they" are connected to.

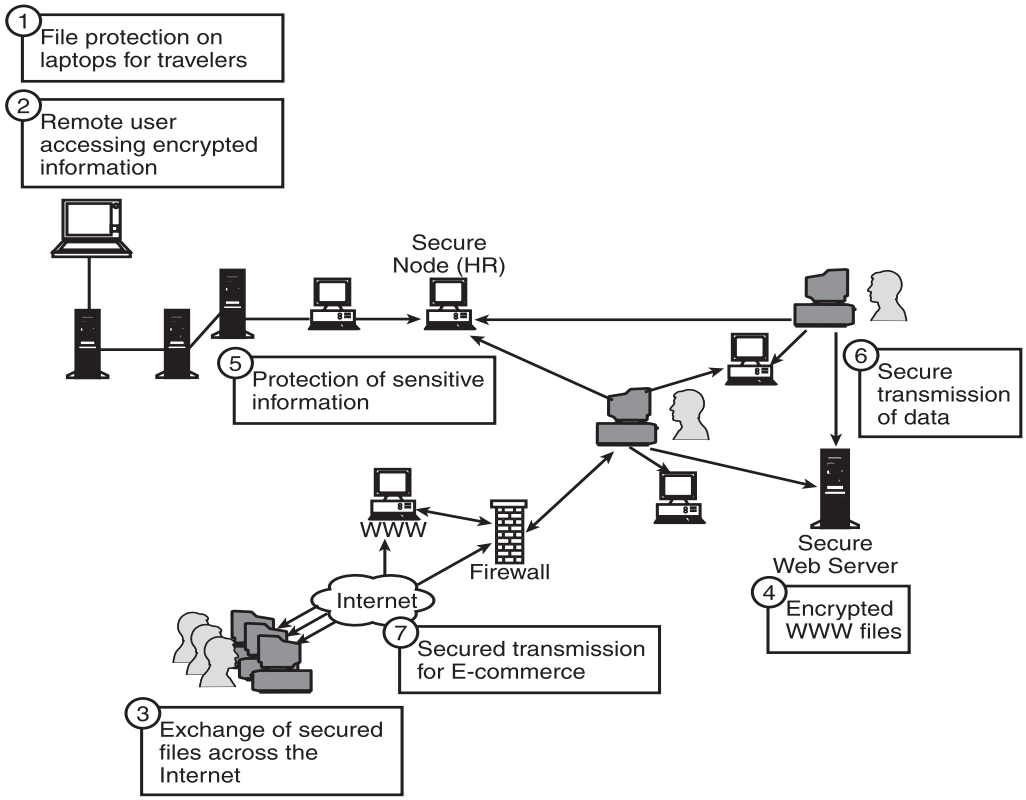


EXHIBIT 121.7 Encryption Services.

Through the deployment of customer access networks, the ability to provide connectivity with security is achieved.

The customer access network is connected to the customer network and to one's corporate network, configured to prevent access between connected partners, and includes a firewall between it and the corporate network. In fact, the customer may also want a firewall between its network and the access point.

With VPN technologies, the customer access network may not be extremely complicated, but does result in a VPN endpoint and specific rules within the VPN device for restricting the protocol types and destinations that the customer is permitted to access.

The rules associated with the individual customer should be stored in the enterprise directory to allow easy setup and removal of the VPN access rules and keys. The real purpose behind the customer access network is not only to build a bridge between the two networks, but also to build a secure bridge.

Conclusion

This article focused on the technologies and concepts behind a security infrastructure. There are other elements that ideally should be part of the security infrastructure, including:

- Desktop and server anti-virus solutions
- Web and e-mail content filtering
- Anti-spam devices

At the same time, however, one's infrastructure must be designed at the conceptual level using the business processes and needs, and not be driven by the available technology. The adage that "the business must drive

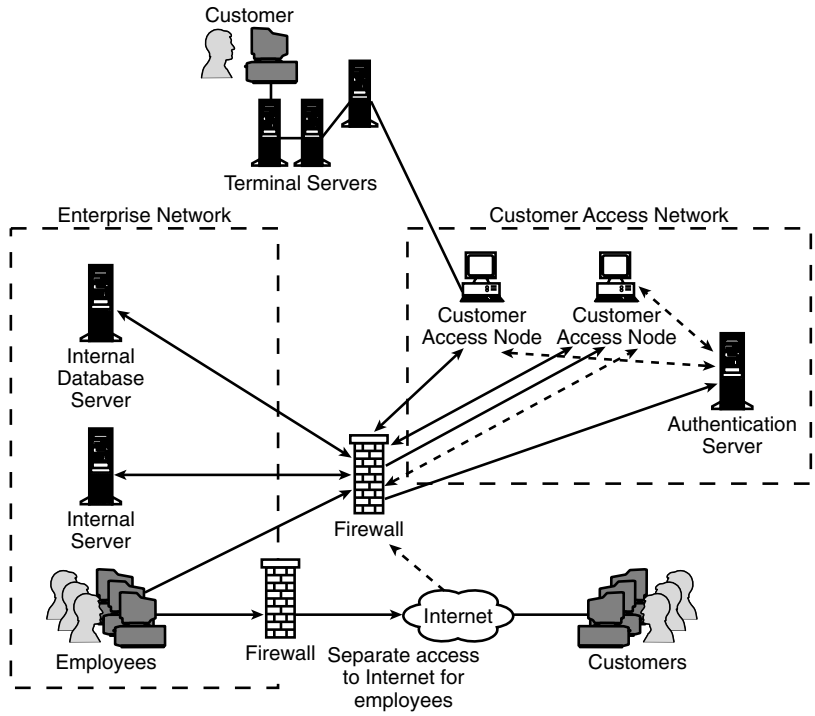


EXHIBIT 121.8 Customer Access Network.

the technology" is especially true. Many security and IT professionals forget that their jobs are dependent upon the viability and success of the enterprise — they exist to serve the enterprise, and not the other way around!

Many infrastructure designers are seduced by the latest and greatest technology. This can have dire consequences for the enterprise due to unreliable code or hardware. Additionally, one nevers knows when one has something that works because one is constantly changing it. To make matters worse, because the users will not know what the "flavor of the week" is, they will simply refuse to use it.

Through the development of a security infrastructure that is global in basis and supported by the management structure, the following benefits are realized:

- The ability to encourage developers to include security in the early stages of their new products or business processes
- The risk and costs associated with new ventures or business partners are reduced an order of magnitude from reactive processes
- Centralized planning and operations with an infrastructure responsive to meeting business needs
- Allow business application developers to deliver stronger controls over stored intellectual capital
- The risks associated with loss of confidentiality are minimized
- A strengthening of security capabilities within the installed backbone applications (e.g., e-mail, servers, Web)
- The privacy and integrity associated with the corporation's intellectual capital are increased
- The risks and costs associated with security failures are reduced

In short, we have created a security infrastructure, which protects the enterprise assets, is manageable, and is a business enabler.

Above all this, the infrastructure must allow the network users, developers, and administrators to contribute to the corporation's security by allowing them to "do the right thing."

122

The Reality of Virtual Computing

Chris Hare, CISSP, CISA

A major issue in many computing environments is accessing the desktop or console display of a different graphical-based system than the one you are using. If you are in a homogeneous environment, meaning you want to access a Microsoft Windows system from a Windows system, you can use applications such as Timbuktu, pcAnywhere, or RemotelyPossible.

In today's virtual enterprise, many people have a requirement to share their desktops or allow others to view or manipulate it. Many desktop-sharing programs exist aside from those mentioned, including Microsoft NetMeeting and online conferencing tools built into various applications.

The same is true for UNIX systems, which typically use the X Windows display system as the graphical user interface. It is a simple matter of running the X Windows client on the remote system and displaying it on the local system.

However, if you must access a dissimilar system (e.g., a Windows system from a UNIX system) the options are limited. It is difficult to find an application under UNIX that allows a user to view an online presentation from a Windows system using Microsoft PowerPoint. This is where Virtual Network Computing, or VNC, from AT&T's United Kingdom Research labs, enters the picture.

This chapter discusses what VNC is, how it can be used, and the security considerations surrounding VNC. The information presented does get fairly technical in a few places to illustrate the protocol, programming techniques, and weaknesses in the authentication scheme. However, the corresponding explanations should address the issues for the less technical reader.

What Is VNC?

The Virtual Network Computing system, or VNC, was developed at the AT&T Research Laboratories in the United Kingdom. VNC is a very simple graphical display protocol allowing connections from heterogeneous or homogeneous computer systems.

VNC consists of a server and a viewer, as illustrated in Exhibit 122.1. The server accepts connection requests to display its local display on the viewer.

The VNC services are based on what is called a *remote framebuffer* or RFB. The framebuffer protocol simply allows a server to update the framebuffer or graphical display device on the remote viewer. With total independence from the graphical device driver, it is possible to represent the local display from the server on the client or viewer. The portability of the design means the VNC server should function on almost any hardware platform, operating system, windowing system, and application.

Support for VNC is currently available for a number of platforms, including:

- Servers:
  - UNIX (X Window system)
  - Microsoft Windows

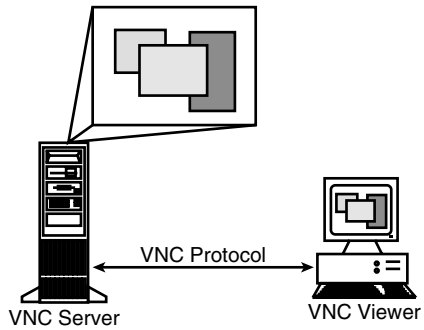


EXHIBIT 122.1 The VNC components.

  - Macintosh
- Viewers:
  - UNIX (X Window System)
  - Microsoft Windows
  - Macintosh
  - Java
  - Microsoft Windows CE

VNC is described as a thin client protocol, making very few requirements on the viewer. In this manner, the client can run on the widest range of hardware. There are a number of factors distinguishing VNC from other remote display systems, including:

- VNC is stateless, meaning you can terminate the session and reconnect from another system and continue right where you left off. When you connect to a remote system using an application such as a PC X Server and the PC crashes or is restarted, the X Window system applications running terminate. Using VNC, the applications remain available after the reboot.
- The viewer is a thin client and has a very small memory footprint.
- VMC is platform independent, allowing a desktop on one system to be displayed on any other type of system, including Java-capable Web browsers.
- It can be shared, allowing multiple users the ability to view and share a single desktop at the same time. This can be useful when needing to perform presentations over the network.
- And, best of all, VNC is free and distributed under the standard GNU General Public License (GPL).

These are some of the benefits available with VNC. However, despite the clever implementation to share massive amounts of video data, there are a few weaknesses, as presented in this chapter.

How It Works

Accessing the VNC server is done using the VNC client and specifying the IP address or node name of the target VNC server as shown in Exhibit 122.2.

The window shown in Exhibit 122.2 requests the node name or IP address for the remote VNC server. It is also possible to add a port number with the address. The VNC server has a password to protect unauthorized access to the server. After providing the target host name or IP address, the user is prompted for the password to access the server, as seen in Exhibit 122.3.

The Microsoft Windows VNC viewer does not display the password when the user enters it, as shown in Exhibit 122.4. However, the VNC client included in Linux systems does not hide the password when the user enters it. This is an issue because it exposes the password for the server to public view. However, because there is no user-level authentication, one could say there is no problem. Just in case you missed it, *there is no user-level authentication*. This is discussed again later in this chapter in the section entitled "Access Control."

EXHIBIT 122.2 The X Windows VNC client.

EXHIBIT 122.3 Entering the VNC server password.

EXHIBIT 122.4 The UNIX VNC client displays the password.

The VNC client prompts for the password after the connection is initiated with the server and requests authentication using a challenge–response scheme. The challenge–response system used is described in the section entitled "Access Control."

Once the authentication is successful, the client and server then exchange a series of messages to negotiate the desktop size, pixel format, and the encoding schemes. To complete the initial connection setup, the client requests a full update for the entire screen and the session commences. Because the client is stateless, either the server or the client can close the connection with no impact to either the client or server.

Actually, this chapter was written logged into a Linux system and using VNC to access a Microsoft Windows system that used VNC to access Microsoft Word. When using VNC on the UNIX- or Linux-based client, the user sees the Windows desktop as illustrated in Exhibit 122.5.

The opposite is also true — a Windows user can access the Linux system and see the UNIX or Linux desktop as well as use the features and functionality offered by the UNIX platform (see Exhibit 122.6). However, VNC is not limited to these platforms, as mentioned earlier and demonstrated later.

However, this may not be exactly what the Linux user was expecting. The VNC sessions run as additional displays on the X server, which on RedHat Linux systems default to the TWM Window Manager. This can be changed; however, that is outside the topic area of this chapter.

EXHIBIT 122.5 The Windows desktop from Linux.

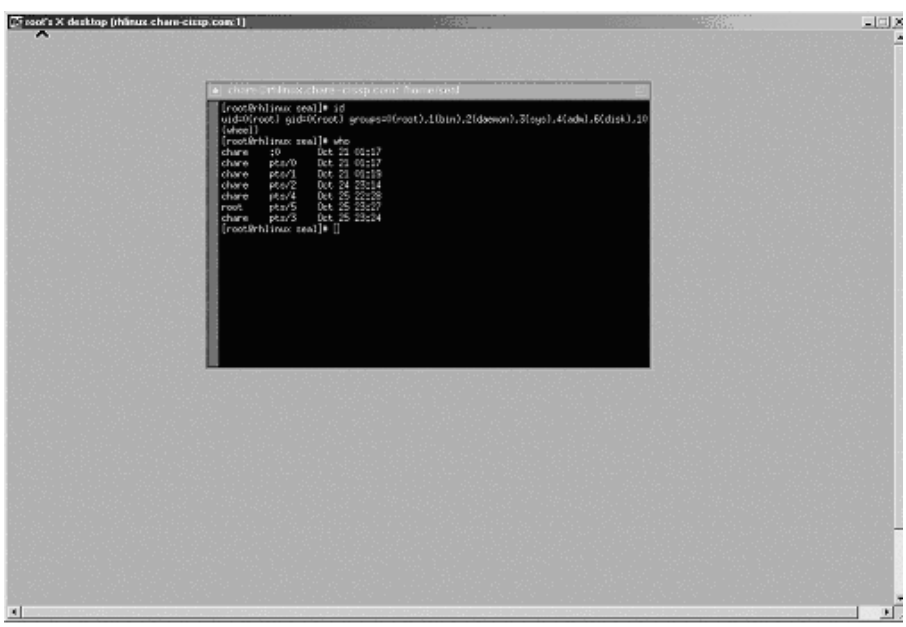


EXHIBIT 122.6 The TWM Window Manager from Windows.

Network Communication

All network communication requires the use of a network port. VNC is a connection-based TCP/IP application requiring the use of network ports. The VNC server listens on two ports. The values of these ports depend on the access method and the display number.

The VNC server listens on port 5900 plus the display number. WinVNC for Microsoft Windows defaults to display zero, so the port is 5900. The same is true for the Java-based HTTP port, listening at port 5800 plus the display number. This small and restrictive Web server is discussed more in the section entitled "VNC and the Web."

If there are multiple VNC servers running on the same system, they will have different port numbers because their display number is different, as illustrated in Exhibit 122.7.

There is a VNC server executed for each user who wishes to have one. Because there is no user authentication in the VNC server, the authentication is essentially port based. This means user chare is running a VNC server, which is set up on display 1 and therefore port 5901. Because the VNC server is running at user chare, anyone who learns or guesses the password for the VNC server can access chare's VNC server and have all of chare's privileges.

Looking back to Exhibit 122.6, the session running on the Linux system belonged to root as shown here:

```
[chare@rhlinux chare]$ ps -ef | grep vnc
root20368    10 23:21 pts/100:00:00 Xvnc :
   1 -desktop X -httpd/usr/s
chare20476204360 23:25 pts/300:00:00 grep vnc
[chare@rhlinux chare]$
```

In this scenario, any user who knows the password for the VNC server on display 1, which is port 5901, can become root with no additional password required. Because of this access control model, good-quality passwords must be used to control access to the VNC server; and they must be kept absolutely secret.

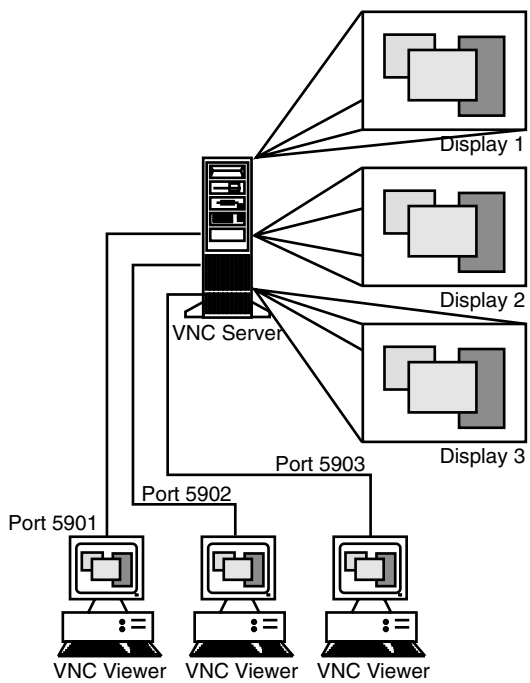


EXHIBIT 122.7 Multiple VNC servers.

As mentioned previously, the VNC server also runs a small Web server to support access through the Java client. The Web server listens on port 58xx, where xx is the display number for the server. The HTTP port on the Web server is only used to establish the initial HTTP connection and download the applet. Once the applet is running in the browser, the connection uses port 59xx. The section entitled "VNC and the Web" describes using the VNC Java client.

There is a third mode, where the client listens for a connection from the server rather than connecting to a server. When this configuration is selected, the client listens on port 5500 for the incoming connection from the server.

Access Control

As mentioned previously, the client and server exchange a series of messages during the initial connection setup. These protocol messages consist of:

- ProtocolVersion
- Authentication
- ClientInitialization
- ServerInitialization

Once the ServerInitialization stage is completed, the client can send additional messages when it requires and receive data from the server.

The protocol version number defines what level of support both the client and server have. It is expected that some level of backward compatibility is available because the version reported should be the latest version the client or server supports. When starting the VNC viewer on a Linux system, the protocol version is printed on the display (standard out) if not directed to a file.

Using a tool such as tcpdump, we can see the protocol version passed from the client to the server (shown in bold text):

```
22:39:42.215633 eth0 < alpha.5900 > rhlinux.chare-cissp.com.1643:
P 1:13(12) ack 1 win 17520 <nop,nop,timestamp 37973 47351119>
                         4500 0040 77f0 0000 8006 4172 c0a8 0002
                         c0a8 0003 170c 066b 38e9 536b 7f27 64fd
                         8018 4470 ab7c 0000 0101 080a 0000 9455
                         02d2 854f 5246 4220 3030 332e 3030 330a

                         E^@ ^@ @ w.. ^@^@ ..^F A r.... ^@^B
                         .... ^@^C ^W^L ^F k 8.. S k ^¿ ' d..
                         ..^X D p  .. | ^@^@ ^A^A ^H^J ^@^@.. U
                        ^B.. .. O RFB 003. 003^J
```

and then again from the server to the client:

```
22:39:42.215633 eth0 > rhlinux.chare-cissp.com.1643
> alpha.5900: P 1:13(12) ack 13 win 5840 <nop,nop,time
stamp47351119 37973> (DF)
 4500 0040 e1b5 4000 4006 d7ac c0a8 0003
 c0a8 0002 066b 170c 7f27 64fd 38e9 5377
 8018 16d0 d910 0000 0101 080a 02d2 854f
 0000 9455 5246 4220 3030 332e 3030 330a
 E^@  ^@ @ .... @^@  @^F  .... .... ^@^C
 .... ^@^B ^F k ^W^L ^¿ ' d..  8..  S w
 ..^X ^V.. ..^P ^@^@ ^A^A ^H^J ^B.. .. O
 ^@^@ .. U RF  B    0 0  3.   0 0  3^J
```

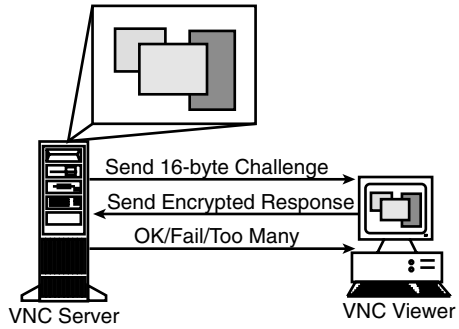


EXHIBIT 122.8 The VNC authentication challenge–response.

With the protocol version established, the client attempts to authenticate to the server. The password prompt shown in Exhibit 122.3 is displayed on the client, where the user enters the password.

There are three possible authentication messages in the VNC protocol:

1. *Connection Failed.* The connection cannot be established for some reason. If this occurs, a message indicating the reason the connection could not be established is provided.
2. *No Authentication.* No authentication is needed. This is not a desirable option.
3. *VNC Authentication.* Use VNC authentication.

The VNC authentication challenge–response is illustrated in Exhibit 122.8.

The VNC authentication protocol uses a challenge–response method with a 16-byte (128-bit) challenge sent from the server to the client. The challenge is sent from the server to the client in the clear. The challenge is random, based on the current time when the connection request is made. The following packet has the challenge highlighted in bold.

```
14:36:08.908961 < alpha.5900 > rhlinux.chare-cissp.com.
2058: P 17:33(16) ack 13 win 17508 <nop,nop,timestamp
800090 8590888>
              4500 0044 aa58 0000 8006 0f06 c0a8 0002
              c0a8 0003 170c 080a ae2b 8b87 f94c 0e34
              8018 4464 1599 0000 0101 080a 000c 355a
              0083 1628 0456 b197 31f3 ad69 a513 151b
              195d 8620

              E^@ ^@ D .. X ^@^@ ..^F ^O^F .... ^@^B
              .... ^@^C ^W^L ^H^J .. + .... .. L ^N 4
              ..^X D d ^U.. ^@^@ ^A^A ^H^J ^@^L 5 Z
              ^@.. ^V ( ^D V .... 1.. .. I ..^S ^U^[
          ^Y] ..
```

The client then encrypts the 16-byte challenge using Data Encryption Standard (DES) symmetric cryptography with the user-supplied password as the key. The VNC DES implementation is based upon a public domain version of Triple-DES, with the double and triple length support removed. This means VNC is only capable of using standard DES for encrypting the response to the challenge. Again, the following packet has the response highlighted in bold.

```
14:36:11.188961 < rhlinux.chare-cissp.com.2058 >
alpha.5900: P 13:29(16) ack 33 win 5840
<nop,nop,timestamp 8591116 800090> (DF)
              4500 0044 180a 4000 4006 a154 c0a8 0003
              c0a8 0002 080a 170c f94c 0e34 ae2b 8b97
```

```
8018 16d0 facd 0000 0101 080a 0083 170c
000c 355a 7843 ba35 ff28 95ee 1493 caa7
0410 8b86

E^@  ^@ D ^X^J @^@  @^F .. T .... ^@^C
.... ^@^B ^H^J ^W^L .. L ^N 4 .. + ....
..^X ^V.. .... ^@^@ ^A^A ^H^J ^@.. ^W^L
^@^L 5 Z  x C .. 5 .. ( .... ^T.. ....
^D^P....
```

The server receives the response and, if the password on the server is the same, the server can decrypt the response and find the value issued as the challenge. As discussed in the section "Weaknesses in the VNC Authentication System" later in this chapter, the approach used here is vulnerable to a man-in-the-middle attack, or a cryptographic attack to find the key, which is the password for the server.

Once the server receives the response, it informs the client if the authentication was successful by providing an *OK*, *Failed*, or *Too Many* response. After five authentication failures, the server responds with *Too Many* and does not allow immediate reconnection by the same client.

The *ClientInitialization* and *ServerInitialization* messages allow the client and server to negotiate the color depth, screen size, and other parameters affecting the display of the framebuffer.

As mentioned in the "Network Communication" section, the VNC server runs on UNIX as the user who started it. Consequently, there are no additional access controls in the VNC server. If the password is not known to anyone, it is safe. Yes and no. Because the password is used as the key for the DES-encrypted response, the password is never sent across the network in the clear. However, as we will see later in the chapter, the challenge–response method is susceptible to a man-in-the-middle attack.

The VNC Server Password

The server password is stored in a password file on the UNIX file system in the ~/.vnc directory. The password is always stored using the same 64-bit key, meaning the password file should be protected using the local file system permissions. Failure to protect the file exposes the password, because the key is consistent across all VNC servers.

The password protection system is the same on the other supported server platforms; however, the location of the password is different.

The VNC source code provides the consistent key:

```
/*
•We use a fixed key to store passwords, because we assume
•that our local file system is secure but nonetheless
•don't want to store passwords as plaintext.
*/

unsigned char fixedkey[8] = {23,82,107,6,35,78,88,7};
```

This fixed key is used as input to the DES functions to encrypt the password; however, the password must be unencrypted at some point to verify authentication.

The VNC server creates the ~/.vnc directory using the standard default file permissions as defined with the UNIX system's umask. On most systems, the default umask is 022, making the the ~/.vnc directory accessible to users other than the owner. However, the password file is explicitly set to force read/write permissions only for the file owner; so the chance of an attacker discovering the password is minimized unless the user changes the permissions on the file, or the attacker has gained elevated user or system privileges.

If the password file is readable to unauthorized users, the server password is exposed because the key is consistent and publicly available. However, the attacker does not require too much information, because the functions to encrypt and decrypt the password in the file are included in the VNC source code. With the knowledge of the VNC default password key and access to the VNC server password file, an attacker can obtain the password using 20 lines of C language source code.

A sample C program, here called attack.c, can be used to decrypt the VNC server password should the password file be visible:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <vncauth.h>
#include <d3des.h>
main( argc, argv)
      int argc;
      char **argv;
{
   char *passwd;
   if (argc < = 1)
      {
      printf ("specify the location and name of a VNC
         password file\n");
      exit(1);
      }
   /* we might have a file */
   passwd = vncDecryptPasswdFromFile(argv[1]);
   printf ("passowrd file is%s\n," argv[1]);
   printf ("password is%s\n," passwd);
   exit(0);
}
```

Note: Do not use this program for malicious purposes. It is provided for education and discussion purposes only.

Running the attack.c program with the location and name of a VNC password file displays the password:

```
[chare@rhlinux libvncauth]$./attack $HOME/.vnc/passwd
passowrd file is/home/chare/.vnc/passwd
password is holycow
```

The attacker can now gain access to the VNC server. Note, however, this scenario assumes the attacker already has access to the UNIX system.

For the Microsoft Windows WinVNC, the configuration is slightly different. Although the methods to protect the password are the same, WinVNC uses the Windows registry to store the server's configuration information, including passwords. The WinVNC registry entries are found at:

- *Local machine-specific settings:*
  HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\
- *Local default user settings:*
  HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\Default
- *Local per-user settings:*
  HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\<*username*>
- *Global per-user settings:*
  HKEY_CURRENT_USER\Software\ORL\WinVNC3

The WinVNC server password will be found in the local default user settings area, unless a specific user defines his own server. The password is stored as an individual registry key value as shown in Exhibit 122.9.

Consequently, access to the registry should be as controlled as possible to prevent unauthorized access to the password.

The password stored in the Windows registry uses the same encryption scheme to protect it as on the UNIX system. However, looking at the password shown in Exhibit 122.9, we see the value:

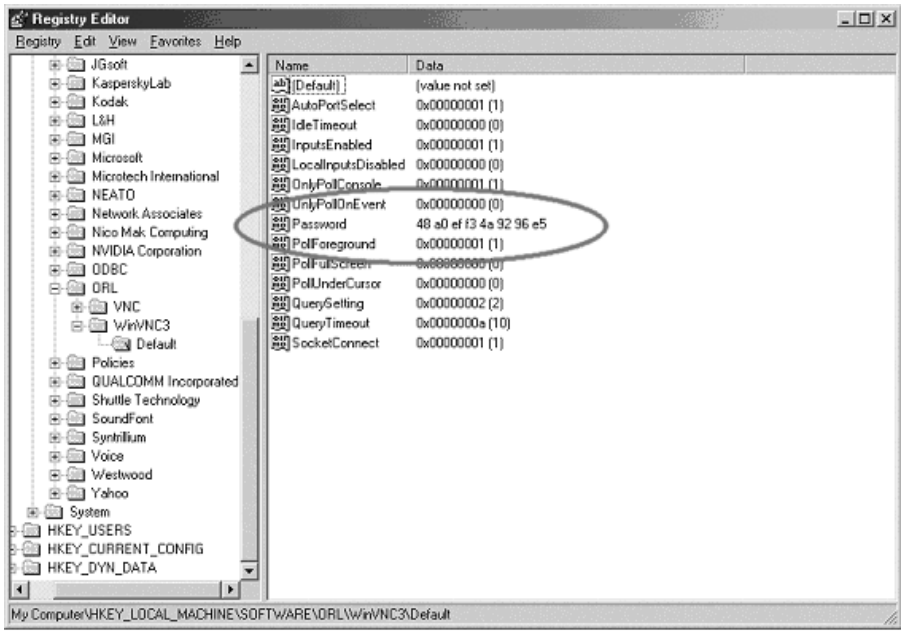


EXHIBIT 122.9 WinVNC Windows registry values.

```
48 a0 ef f3 4a 92 96 e5
```

and the value stored on UNIX is:

```
a0 48 f3 ef 92 4a e5 96
```

Comparing these values, we see that the byte ordering is different. However, knowing that the ordering is different, we can use a program to create a binary file on UNIX with the values from the Windows system and then use the attack.c program above to determine the actual password. Notice that because the password values shown in this example are the same, and the encryption used to hide the passwords is the same, the passwords are the same.

Additionally, the VNC password is limited to eight characters. Even if the user enters a longer password, it is truncated to eight. Assuming a good- quality password with 63 potential characters in each position, this represents only 63^8 possible passwords. Even with this fairly large number, the discussion thus far has demonstrated the weaknesses in the authentication method.

Running a VNC Server under UNIX

The VNC server running on a UNIX system uses the X Window System to interact with the X-based applications on UNIX. The applications are not aware there is no physical screen attached to the system. Starting a new VNC server is done by executing the command:

```
vncserver
```

on the UNIX host. Because the vncserver program is actually written in Perl, most common problems with starting vncserver are associated with the Perl installation or directory structures.

Any user on the UNIX host can start a copy of the VNC server. Because there is no user authentication built into the VNC server or protocol, running a separate server for each user is the only method of providing limited access. Each vncserver has its own password and port assignment, as presented earlier in the chapter.

The first time a user runs the VNC server, he is prompted to enter a password for the VNC server. Each VNC server started by the same user will have the same password. This occurs because the UNIX implemen-

tation of VNC creates a directory called .vnc in the user's home directory. The .vnc directory contains the log files, PID files, password, and X startup files. Should the user wish to change the password for the VNC servers, he can do so using the `vncpasswd` command.

VNC Display Names

Typically the main display for a workstation using the X Window System is display 0 (zero). This means on a system named *ace*, the primary display is ace:0. A UNIX system can run as many VNC servers as the users desire, with the display number incrementing for each one. Therefore, the first VNC server is display ace:1, the second ace:2, etc. Individual applications can be executed and, using the DISPLAY environment variable defined, send their output to the display corresponding to the desired VNC server.

For example, sending the output of an xterm to the second VNC server on display ace:2 is accomplished using the command:

```
xterm -display ace:2 &
```

Normally, the `vncserver` command chooses the first available display number and informs the user what that display is; however, the display number can be specified on the command line to override the calculated default:

```
vncserver :2
```

No visible changes occur when a new VNC server is started, because only a viewer connected to that display can actually see the resulting output from that server. Each time a connection is made to the VNC server, information on the connection is logged to the corresponding server log file found in the $HOME/.vnc directory of the user executing the server. The clog file contents are discussed in the "Logging" section of this chapter.

VNC as a Service

Instead of running individual VNC servers, there are extensions available to provide support for VNC under the Internet Super-Daemon, inetd and xinetd. More information on this configuration is available from the AT&T Laboratories Web site.

VNC and Microsoft Windows

The VNC server is also available for Microsoft Windows, providing an alternative to other commercial solutions and integration between heterogeneous operating systems and platforms. The VNC server under Windows is run as a separate application or a service. Unlike the UNIX implementation, the Windows VNC server can only display the existing desktop of the PC console to the user. This is a limitation of Microsoft Windows, and not WinVNC. WinVNC does not make the Windows system a multi-user environment: if more than one user connects to the Windows system at the same time, they will all see the same desktop.

Running WinVNC as a service is the preferred mode of operation because it allows a user to log on to the Windows system, perform his work, and then log off again.

When running WinVNC, an icon as illustrated in Exhibit 122.10 is displayed. When a connection is made, the icon changes color to indicate there is an active connection.

The WinVNC properties dialog shown in Exhibit 122.11 allows the WinVNC user to change the configuration of WinVNC. All the options are fully discussed in the WinVNC documentation.

With WinVNC running as a service, a user can connect from a remote system even when no user is logged on at the console. Changing the properties for WinVNC when it is running as a service has the effect of changing the service configuration, also known as the default properties, rather than the individual user properties. However, running a nonservice mode WinVNC means a user must have logged in on the console and started WinVNC for it to work correctly. Exhibit 122.12 illustrates accessing WinVNC from a Linux system while in service mode.

Aside from the specific differences for configuring the WinVNC server, the password storage and protocol-level operations are the same, regardless of the platform. Because there can be only one WinVNC server running at a time, connections to the server are on ports 5900 for the VNC viewer and 5800 for the Java viewer.

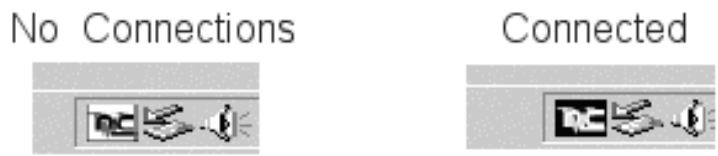


EXHIBIT 122.10 WinVNC system tray icons.

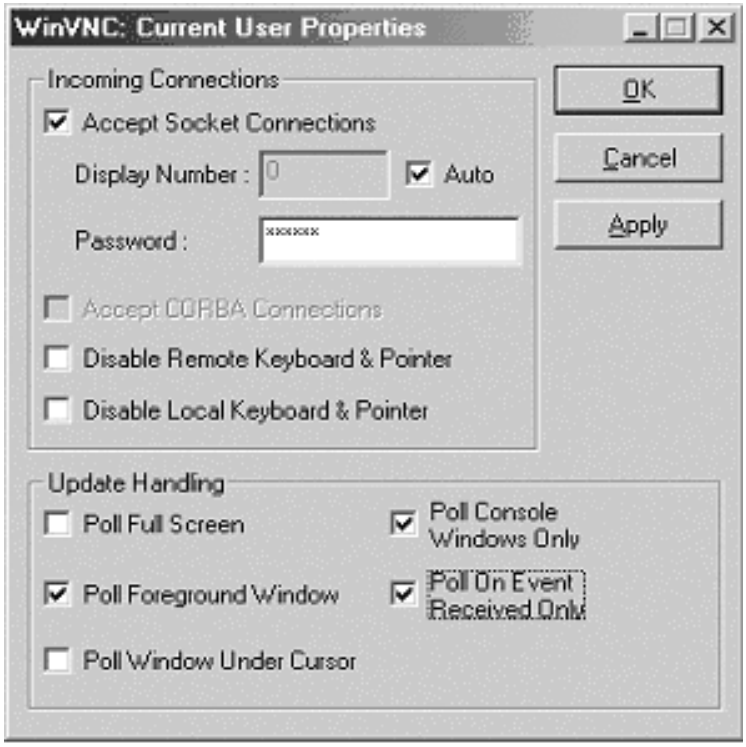


EXHIBIT 122.11 The WinVNC Properties dialog.

VNC and the Web

As mentioned previously, each VNC server listens not only on the VNC server port but also on a second port to support Web connections using a Java applet and a Web browser. This is necessary to support Java because a Java applet can only make a connection back to the machine from which it was served.

Connecting to the VNC server using a Java-capable Web browser to:

```
http://ace:5802/
```

loads the Java applet and presents the log-in screen where the password is entered. Once the password is provided, the access controls explained earlier prevail. Once the applet has connected to the VNC server port, the user sees a display resembling that shown in Exhibit 122.13.

With the Java applet, the applications displayed through the Web browser can be manipulated as if they were displayed directly through the VNC client or on the main display of the workstation.

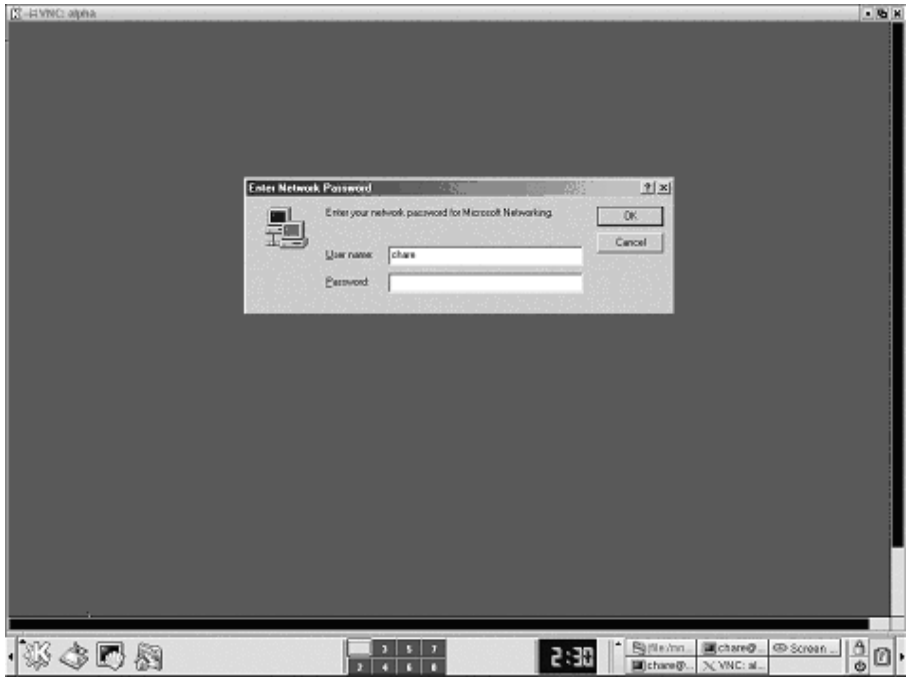


EXHIBIT 122.12 Accessing WinVNC in service mode.

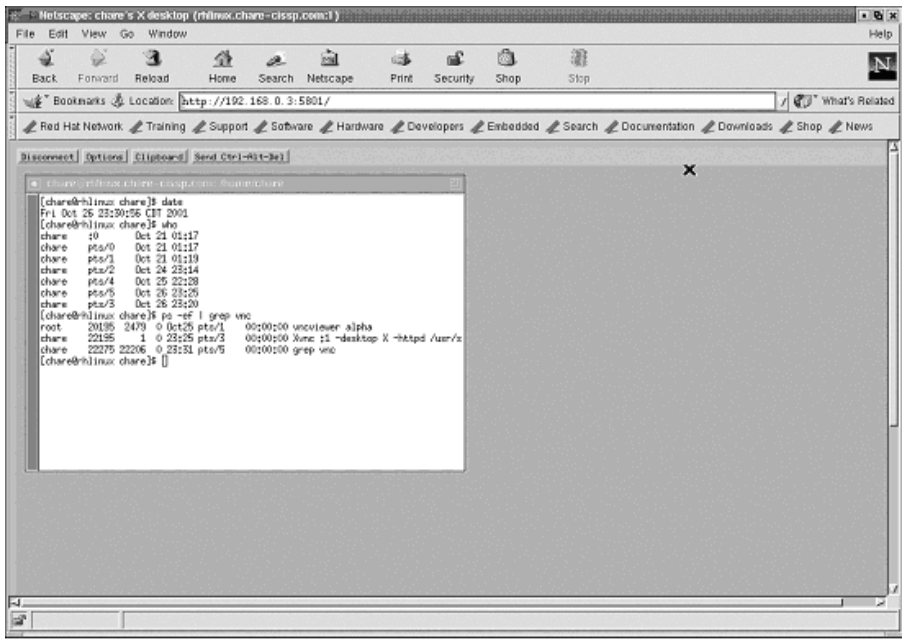


EXHIBIT 122.13 A VNC connection using a Java-capable Web browser.

Logging

As with any network-based application, connection and access logs provide valuable information regarding the operation of the service. The log files from the VNC server provide similar information for debugging or later analysis. A sample log file resembles the following. The first part of the log always provides information on the VNC server, including the listing ports, the client name, display, and the URL.

```
26/10/01 23:25:47 Xvnc version 3.3.3r2
26/10/01 23:25:47 Copyright © AT&T Laboratories Cambridge.
26/10/01 23:25:47 All Rights Reserved.
26/10/01 23:25:47 See http://www.uk.research.att.com/
   vnc for information on VNC
26/10/01 23:25:47 Desktop name 'X' (rhlinux.chare-cissp.com:1)
26/10/01 23:25:47 Protocol version supported 3.3
26/10/01 23:25:47 Listening for VNC connections on TCP port 5901
26/10/01 23:25:47 Listening for HTTP connections on TCP port 5801
26/10/01 23:25:47  URL http://rhlinux.chare-cissp.com:5801
```

The following sample log entry shows a connection received on the VNC server. We know the connection came in through the HTTPD server from the log entry. Notice that there is no information regarding the user who is accessing the system — only the IP address of the connecting system.

```
26/10/01 23:28:54 httpd: get '' for 192.168.0.2
26/10/01 23:28:54 httpd: defaulting to 'index.vnc'
26/10/01 23:28:56 httpd: get 'vncviewer.jar' for 192.168.0.2
26/10/01 23:29:03 Got connection from client 192.168.0.2
26/10/01 23:29:03 Protocol version 3.3
26/10/01 23:29:03 Using hextile encoding for client 192.168.0.2
26/10/01 23:29:03 Pixel format for client 192.168.0.2:
26/10/01 23:29:03  8 bpp, depth 8
26/10/01 23:29:03  true colour: max r 7 g 7 b 3, shift r 0 g 3 b 6
26/10/01 23:29:03 no translation needed
26/10/01 23:29:21 Client 192.168.0.2 gone
26/10/01 23:29:21 Statistics:
26/10/01 23:29:21  key events received 12, pointer events 82
26/10/01 23:29:21  framebuffer updates 80, rectangles 304, bytes 48528
26/10/01 23:29:21    hextile rectangles 304, bytes 48528
26/10/01 23:29:21  raw bytes equivalent 866242, compression ratio
 17.850354
```

The log file contains information regarding the connection with the client, including the color translations. Once the connection is terminated, the statistics from the connection are logged for later analysis, if required.

Because there is no authentication information logged, the value of the log details for a security analysis are limited to knowing when and from where a connection was made to the server. Because many organizations use DHCP for automatic IP address assignment and IP addresses may be spoofed, the actual value of knowing the IP address is reduced.

Weaknesses in the VNC Authentication System

We have seen thus far several issues that will have the security professional concerned. However, these can be alleviated as discussed later in the chapter. There are two primary concerns with the authentication. The first is the man-in-the-middle attack, and the second is a cryptographic attack to uncover the password.

The Random Challenge

The random challenge is generated using the rand(3) function in the C programming language to generate random numbers. The random number generator is initialized using the system clock and the current system time. However, the 16-byte challenge is created by successive calls to the random number generator, decreasing the level of randomness on each call. (Each call returns 1 byte or 8 bits of data.)

This makes the challenge predictable and increases the chance an attacker could establish a session by storing all captured responses and their associated challenges. Keeping track of each challenge–response pair can be difficult and, as discussed later, not necessary.

The Man-in-the-Middle Attack

For the purposes of this illustration, we will make use of numerous graphics to facilitate understanding this attack method. The server is system S, the client is C, and the attacker, or man in the middle, is A. (This discussion ignores the possibility the network connection may be across a switched network, or that there are ways of defeating the additional security provided by the switched network technology.)

The attacker A initiates a connection to the server, as seen in Exhibit 122.14. The attacker connects, and the two systems negotiate the protocols supported and what will be used. The attacker observes this by sniffing packets on the network.

We know both the users at the client and server share the DES key, which is the password. The attacker does not know the key. The password is used for the DES encryption in the challenge–response.

The server then generates the 16-byte random challenge and transmits it to the attacker, as seen in Exhibit 122.15. Now the attacker has a session established with the server, pending authorization.

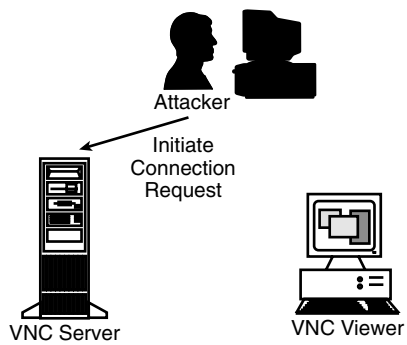


EXHIBIT 122.14 Attacker opens connection to VNC server.

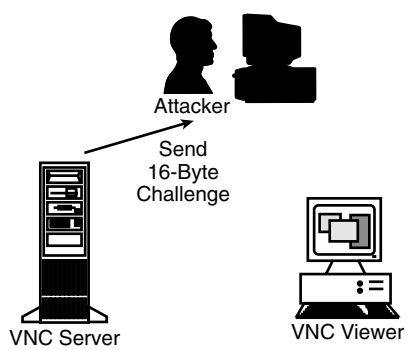


EXHIBIT 122.15 Server sends challenge to attacker.

At this point, the attacker simply waits, watching the network for a connection request to the same server from a legitimate client. This is possible as there is no timeout in the authentication protocol; consequently, the connection will wait until it is completed.

When the legitimate client attempts a connection, the server and client negotiate their protocol settings, and the server sends the challenge to the client as illustrated in Exhibit 122.16. The attacker captures the authentication request and changes the challenge to match the one provided to him by the server.

Once the attacker has modified the challenge, he forges the source address and retransmits it to the legitimate client. As shown in Exhibit 122.17, the client then receives the challenge, encrypts it with the key, and transmits the response to the server.

The server receives two responses: one from the attacker and one from the legitimate client. However, because the attacker replaced the challenge sent to the client with his own challenge, the response sent by the client to server does not match the challenge. Consequently, the connection request from the legitimate client is refused.

However, the response sent does match the challenge sent by the server to the attacker; and when the response received from the attacker matches the calculated response on the server, the connection is granted. The attacker has gained unauthorized access to the VNC server.

Cryptographic Attacks

Because the plaintext challenge and the encrypted response can both be retrieved from the network, it is possible to launch a cryptographic attack to determine the key used, which is the server's password. This is easily done through a brute-force or known plaintext attack.

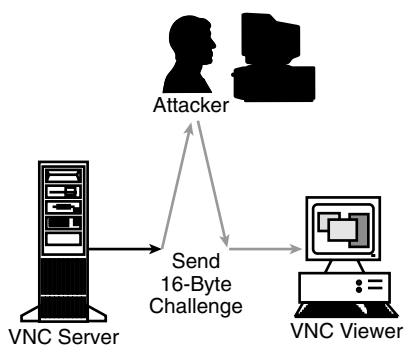


EXHIBIT 122.16 Attacker captures and replaces challenge.

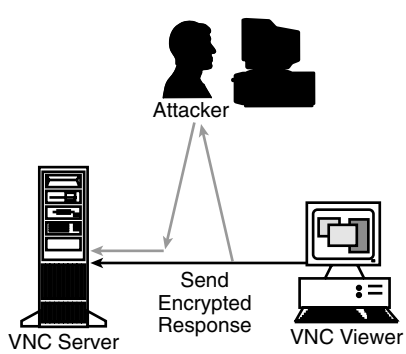


EXHIBIT 122.17 Attacker and client send encrypted response.

A brute-force attack is the most effective, albeit time-consuming method of attack. Both linear cryptanalysis, developed by Lester Mitsui, and differential cryptanalysis, developed by Biham and Shamir, are considered the two strongest analytic (shortcut) methods for breaking modern ciphers; and even these have been shown as not very practical, even against Single-DES.

The known plaintext attack is the most advantageous method because a sample of ciphertext (the response) is available as well as a sample of the plaintext (the challenge). Publicly available software such as *crack* could be modified to try a dictionary and brute-force attack by repeatedly encrypting the challenge until a match for the response is found. The nature of achieving the attack is beyond the scope of this chapter.

Finding VNC Servers

The fastest method of finding VNC servers in an enterprise network is to scan for them on the network devices. For example, the popular nmap scanner can be configured to scan only the ports in the VNC range to locate the systems running it.

```
[root@rhlinux chare]# nmap -p "5500,5800-5999" 192.168.0.1-5
Starting nmap V. 2.54BETA29 (www.insecure.org/nmap/)
All 201 scanned ports on gateway (192.168.0.1) are: filtered
Interesting ports on alpha (192.168.0.2):
(The 199 ports scanned but not shown below are in state: closed)
Port       State   Service
5800/tcp   open    vnc
5900/tcp   open    vnc

Interesting ports on rhlinux.chare-cissp.com (192.168.0.3):
(The 199 ports scanned but not shown below are in state: closed)
Port       State   Service
5801/tcp   open    vnc
5901/tcp   open    vnc-1

Nmap run completed — 5 IP addresses (3 hosts up) scanned in 31 seconds
[root@rhlinux chare]#
```

There are other tools available to find and list the VNC servers on the network; however, nmap is fast and will identify not only if VNC is available on the system at the default ports but also all VNC servers on that system.

Improving Security through Encapsulation

To this point we have seen several areas of concern with the VNC environment:

- There is no user-level authentication for the VNC server.
- The challenge–response system is vulnerable to man-in-the-middle and cryptographic attacks.
- There is no data confidentiality built into the client and server.

Running a VNC server provides the connecting user with the ability to access the entire environment at the privilege level for the user running the server. For example, assuming root starts the first VNC server on a UNIX system, the server listens on port 5901. Any connections to this port where the remote user knows the server password result in a session with root privileges.

We have seen how it could be possible to launch a man-in-the-middle or cryptographic attack against the authentication method used in VNC. Additionally, once the authentication is completed, all the session data is unencrypted and could, in theory, be captured, replayed, and watched by malicious users. However, because VNC uses a simple TCP/IP connection, it is much easier to add encryption support with Secure Sockets Layer (SSL) or Secure Shell (SSH) than, say, a telnet, rlogin, or X Window session.

Secure Shell (SSH) is likely the more obvious choice for most users, given there are clients for most operating systems. SSH encrypts all the data sent through the tunnel and supports port redirection; thus, it can be easily

supported with VNC. Furthermore, although VNC uses a very efficient protocol for carrying the display data, additional benefits can be achieved at slower network link speeds because SSH can also compress the data.

There are a variety of SSH clients and servers available for UNIX, although if you need an SSH server for Windows, your options are very limited and may result in the use of a commercial implementation. However, SSH clients for Windows and the Apple Macintosh are freely available. Additionally, Mindbright Technology offers a modified Java viewer supporting SSL.

Because UNIX is commonly the system of choice for operating a server, this discussion focuses on configuring VNC with SSH using a UNIX-based system. Similar concepts are applicable for Windows-based servers, once you have resolved the SSH server issue. However, installing and configuring the base SSH components are not discussed in this chapter.

Aside from the obvious benefits of using SSH to protect the data while traveling across the insecure network, SSH can compress the data as well. This is significant if the connection between the user and the server is slow, such as a PPP link. Performance gains are also visible on faster networks, because the compression can make up for the time it takes to encrypt and decrypt the packets on both ends.

A number of extensions are available to VNC, including support for connections through the Internet superserver inetd or xinetd. These extensions mean additional controls can be implemented using the TCP Wrapper library. For example, the VNC X Window server, Xvnc, has been compiled with direct support for TCP Wrappers.

More information on configuring SSH, inetd, and TCP Wrappers is available on the VNC Web site listed in the "References" section of this chapter.

Summary

The concept of thin client computing will continue to grow and develop to push more and more processing to centralized systems. Consequently, applications such as VNC will be with the enterprise for some time. However, the thin client application is intended to be small, lightweight, and easy to develop and transport. The benefits are obvious — smaller footprint on the client hardware and network, including support for many more devices including handheld PCs and cell phones, to name a few.

However, the thin client model has a price; and in this case it is security. Although VNC has virtually no security features in the protocol, other add-on services such as SSH, VNC, and TCP Wrapper, or VNC and xinetd provide extensions to the basic VNC services to provide access control lists limited by the allowable network addresses and data confidentiality and integrity.

Using VNC within an SSH tunnel can provide a small, lightweight, and secured method of access to that system 1000 miles away from your office. For enterprise or private networks, there are many advantages to using VNC because the protocol is smaller and more lightweight than distributing the X Window system on Microsoft Windows, and it has good response time even over a slower TCP/IP connection link. Despite the security considerations mentioned in this chapter, there are solutions to address them; so you need not totally eliminate the use of VNC in your organization.

References

1. CORE SDI advisory: weak authentication in AT&T's VNC, http://www.uk.research.att.com/ vnc/archives/2001–01/0530.html.
2. VNC Computing Home Page, http://www.uk.research.att.com/vnc/index.html.
3. VNC Protocol Description, http://www.uk.research.att.com/vnc/rfbproto.pdf.
4. VNC Protocol Header, http://www.uk.research.att.com/vnc/rfbprotoheader.pdf.
5. VNC Source Code, http://www.uk.research.att.com/vnc/download.html.

123

Overcoming Wireless LAN Security Vulnerabilities

Gilbert Held

The IEEE 802.11b specification represents one of three wireless LAN standards developed by the Institute of Electrical and Electronic Engineers. The original standard, which was the 802.11 specification, defined wireless LANs using infrared, Frequency Hopping Spread Spectrum (FHSS), and Direct Sequence Spread Spectrum (DSSS) communications at data rates of 1 and 2 Mbps. The relatively low operating rate associated with the original IEEE 802.11 standard precluded its widespread adoption.

The IEEE 802.11b standard is actually an annex to the 802.11 standard. This annex specifies the use of DSSS communications to provide operating rates of 1, 2, 5.5, and 11 Mbps.

A third IEEE wireless LAN standard, IEEE 802.11a, represents another annex to the original standard. Although 802.11- and 802.11b-compatible equipment operate in the 2.4-GHz unlicensed frequency band, to obtain additional bandwidth to support higher data rates resulted in the 802.11a standard using the 5-GHz frequency band. Although 802.11a equipment can transfer data at rates up to 54 Mbps, because higher frequencies attenuate more rapidly than lower frequencies, approximately four times the number of access points are required to service a given geographic area than if 802.11b equipment is used. Due to this, as well as the fact that 802.11b equipment reached the market prior to 802.11a devices, the vast majority of wireless LANs are based on the use of 802.11b compatible equipment.

Security

Under all three IEEE 802.11 specifications, security is handled in a similar manner. The three mechanisms that affect wireless LAN security under the troika of 802.11 specifications include the specification of the network name, authentication, and encryption.

Network Name

To understand the role of the network name requires a small diversion to discuss a few wireless LAN network terms. Each device in a wireless LAN is referred to as a station, to include both clients and access points. Client stations can communicate directly with one another, referred to as *ad hoc* networking. Client stations can also communicate with other clients, both wireless and wired, through the services of an access point. The latter type of networking is referred to as infrastructure networking.

In an infrastructure networking environment, the group of wireless stations to include the access point form what is referred to as a basic service set (BSS). The basic service set is identified by a name. That name, which is formally referred to as the service set identifier (SSID), is also referred to as the network name.

One can view the network name as a password. Each access point normally is manufactured with a set network name that can be changed. To be able to access an access point, a client station must be configured

with the same network name as that configured on the access point. Unfortunately, there are three key reasons why the network name is almost valueless as a password. First, most vendors use a well-known default setting that can be easily learned by surfing to the vendor's Web site and accessing the online manual for their access point. For example, Netgear uses the network name "Wireless." Second, access points periodically transmit beacon frames that define their presence and operational characteristics to include their network name. Thus, the use of a wireless protocol analyzer, such as WildPackets' Airopeek or Sniffer Technologies' Wireless Sniffer could be used to record beacon frames as a mechanism to learn the network name.

A third problem associated with the use of the network name as a password for access to an access point is the fact that there are two client settings that can be used to override most access point network name settings. The configuration of a client station to a network name of "ANY" or its setting to a blank can normally override the setting of a network name or an access point.

Exhibit 123.1 illustrates an example of the use of SMC Networks' EZ Connect Wireless LAN Configuration Utility program to set the SSID to a value of "ANY." Once this action was accomplished, this author was able to access a Netgear wireless router/access point whose SSID was by default set to a value of "Wireless." Thus, the use of the SSID or network name as a password to control access to a wireless LAN needs to be considered as a facility easily compromised, as well as one that offers very limited potential.

Authentication

A second security mechanism included within all three IEEE wireless LAN specifications is authentication. Authentication represents the process of verifying the identity of a wireless station. Under the IEEE 802.11 standard to include the two addenda, authentication can be either open or shared key. Open authentication in effect means that the identity of a station is not checked. The second method of authentication, which is referred to as shared key, assumes that when encryption is used, each station that has the correct key and is operating in a secure mode represents a valid user. Unfortunately, as soon noted, shared key authentication is vulnerable because the Wired Equivalent Privacy (WEP) key can be learned by snooping on the radio frequency.

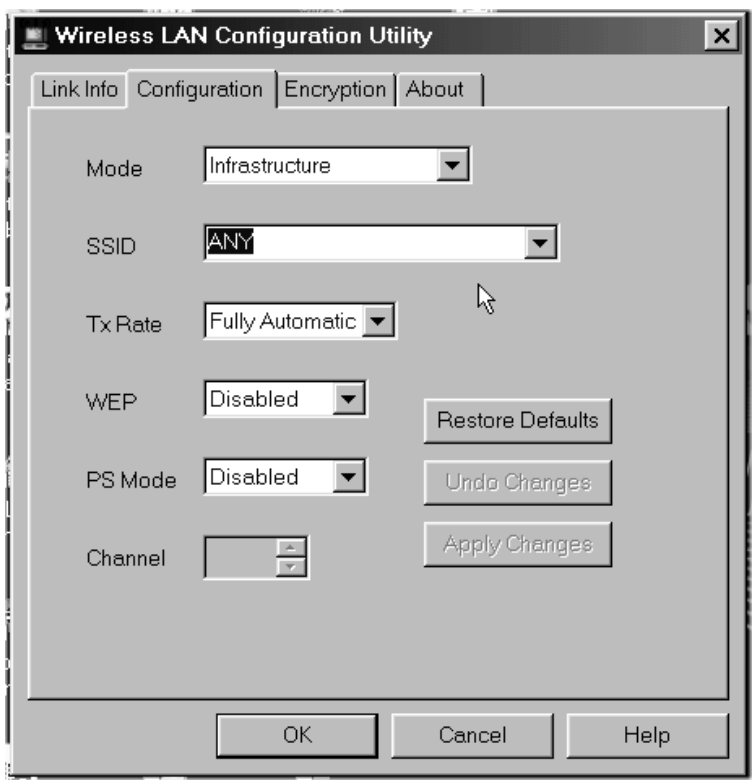


EXHIBIT 123.1 Setting the value of the SSID or network name to "ANY".

Encryption

The third security mechanism associated with IEEE 802.11 networks is encryption. The encryption used under the 802.11 series of specifications is referred to as WEP. The initial goal of WEP is reflected by its name. That is, its use is designed to provide a level of privacy equivalent to that occurring when a person uses a wired LAN. Thus, some of the vulnerabilities uncovered concerning WEP should not be shocking because the goal of WEP is not to bulletproof a network. Instead, it is to simply make over-the-air transmission difficult for a third party to understand. However, as we will note, there are several problems associated with the use of WEP that make it relatively easy for a third party to determine the composition of network traffic flowing on a network.

Exhibit 123.2 illustrates the drop-down list of the WEP field of SMC Networks' Wireless Configuration Utility program. Note that, by default, WEP is disabled; and unless you alter the configuration on your client stations and access points, any third party within transmission range could use a wireless LAN protocol analyzer to easily record all network activity. In fact, during the year 2001, several articles appeared in *The New York Times* and *The Wall Street Journal* concerning the travel of two men in a van from one parking lot to another in Silicon Valley. Using a directional antenna focused at each building from a parking lot and a notebook computer running a wireless protocol analyzer program, these men were able to easily read most network traffic because most networks were set up with WEP disabled.

Although enabling WEP makes it more difficult to decipher traffic, the manner by which WEP encryption occurs has several shortcomings. Returning to Exhibit 123.2, note that the two WEP settings are shown as "64 Bit" and "128 Bit." Although the use of 64- and 128-bit encryption keys may appear to represent a significant barrier to decryption, the manner by which WEP encryption occurs creates several vulnerabilities. An explanation follows.

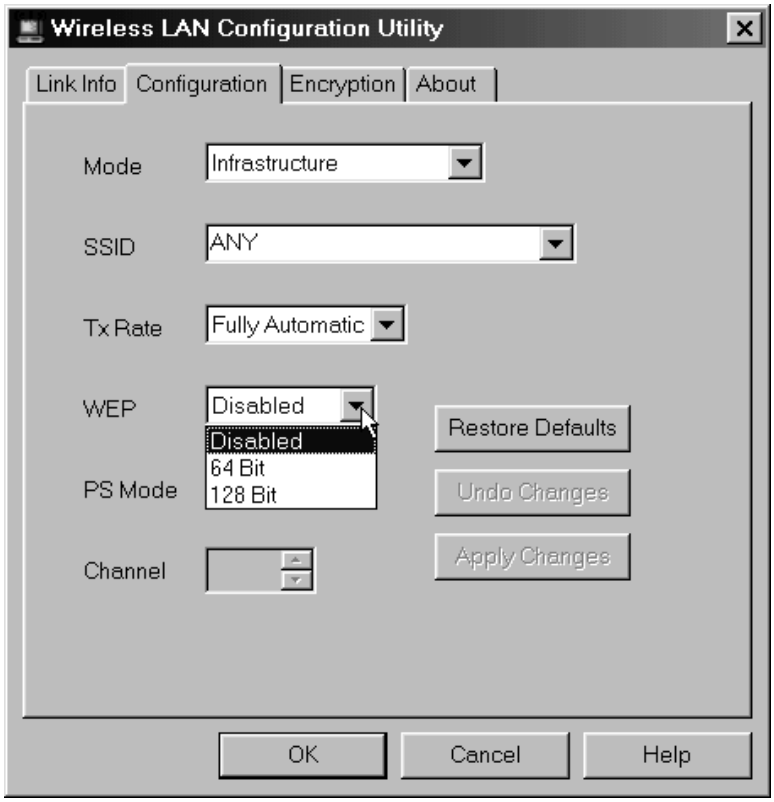


EXHIBIT 123.2 WEP settings.

EXHIBIT 123.3 Creating a WEP encryption key.

WEP encryption occurs via the creation of a key that is used to generate a pseudo-random binary string that is modulo-2 added to plaintext to create ciphertext. The algorithm that uses the WEP key is a stream cipher, meaning it uses the key to create an infinite pseudo-random binary string.

Exhibit 123.3 illustrates the use of SMC Networks' Wireless LAN Configuration Utility program to create a WEP key. SMC Networks simplifies the entry of a WEP key by allowing the user to enter a passphrase. Other vendors may allow the entry of hex characters or alphanumeric characters. Regardless of the manner by which a WEP key is entered, the total key length consists of two elements: an initialization vector (IV) that is 24 bits in length and the entered WEP key. Because the IV is part of the key, this means that a user constructing a 64-bit WEP key actually specifies 40 bits in the form of a passphrase or 10 hex digits, or 104 bits in the form of a passphrase or 26 hex digits for a 128-bit WEP key.

Because wireless LAN transmissions can easily be reflected off surfaces and moving objects, multiple signals can flow to a receiver. Referred to as multipath transmission, the receiver needs to select the best transmission and ignore the other signals. As one might expect, this can be a difficult task, resulting in a transmission error rate considerably higher than that encountered on wired LANs. Due to this higher error rate, it would not be practical to use a WEP key by itself to create a stream cipher that continues for infinity. This is because a single bit received in error would adversely affect the decryption of subsequent data.

Recognizing this fact, the IV is used along with the digits of the WEP key to produce a new WEP key on a frame-by-frame basis. Although this is a technically sound action, unfortunately the 24-bit length of the IV used in conjunction with a 64- or 104-bit fixed length WEP key causes several vulnerabilities. First, the IV is transmitted in the clear, allowing anyone with appropriate equipment to record its composition along with the encrypted frame data. Because the IV is only 24 bits in length, it will periodically repeat. Thus, capturing two or more of the same IVs and the encrypted text makes it possible to perform a frequency analysis of the encrypted text that can be used as a mechanism to decipher the captured data. For example, assume one has captured several frames that had the same IV. Because "e" is the most common letter used in the English language followed by the letter "t," one would begin a frequency analysis by searching for the most common

letter in the encrypted frames. If the letter "x" was found to be the most frequent, there would be a high probability that the plaintext letter "e" was encrypted as the letter "x." Thus, the IV represents a serious weakness that compromises encryption.

During mid-2001, researchers at Rice University and AT&T Laboratories discovered that by monitoring approximately five hours of wireless LAN traffic, it became possible to determine the WEP key through a series of mathematical manipulations, regardless of whether a 64-bit or 128-bit key was used. This research was used by several software developers to produce programs such as Airsnort, which enables a person to determine the WEP key in use and to become a participant on a wireless LAN. Thus, the weakness of the WEP key results in shared key authentication being compromised as a mechanism to validate the identity of wireless station operators. Given an appreciation for the vulnerabilities associated with wireless LAN security, one can now focus on the tools and techniques that can be used to minimize or eliminate such vulnerabilities.

MAC Address Checking

One of the first methods used to overcome the vulnerabilities associated with the use of the network name or SSID, as well as shared key authentication, was MAC address checking. Under MAC address checking, the LAN manager programs the MAC address of each client station into an access point. The access point only allows authorized MAC addresses occurring in the source address field of frames to use its facilities.

Although the use of MAC address checking provides a significant degree of improvement over the use of a network name for accessing the facilities of an access point, by itself it does nothing to alter the previously mentioned WEP vulnerabilities. To attack the vulnerability of WEP, several wireless LAN equipment vendors introduced the use of dynamic WEP keys.

Dynamic WEP Keys

Because WEP becomes vulnerable by a third party accumulating a significant amount of traffic that flows over the air using the same key, it becomes possible to enhance security by dynamically changing the WEP key. Several vendors have recently introduced dynamic WEP key capabilities as a mechanism to enhance wireless security. Under a dynamic key capability, a LAN administrator, depending on the product used, may be able to configure equipment to either exchange WEP keys on a frame-by-fame basis or at predefined intervals. The end result of this action is to limit the capability of a third party to monitor a sufficient amount of traffic that can be used to either perform a frequency analysis of encrypted data or to determine the WEP key in use. Although dynamic WEP keys eliminate the vulnerability of a continued WEP key utilization, readers should note that each vendor supporting this technology does so on a proprietary basis. This means that if one anticipates using products from multiple vendors, one may have to forego the use of dynamic WEP keys unless the vendors selected have cross-licensed their technology to provide compatibility between products. Having an appreciation for the manner by which dynamic WEP keys can enhance encryption security, this discussion of methods to minimize wireless security vulnerabilities concludes with a brief discussion of the emerging IEEE 802.1x standard.

The IEEE 802.1x Standard

The IEEE 802.1x standard is being developed to control access both to wired and wireless LANs. Although the standard was not officially completed during early 2002, Microsoft added support for the technology in its Windows XP operating system released in October 2001.

Under the 802.1x standard, a wireless client station attempting to access a wired infrastructure via an access point will be challenged by the access point to identify itself. The client will then transmit its identification to the access point. The access point will forward the challenge response to an authentication server located on the wired network. Upon authentication, the server will inform the access point that the wireless client can access the network, resulting in the access point allowing frames generated by the client to flow onto the wired network.

Although the 802.1x standard can be used to enhance authentication, by itself it does not enhance encryption. Thus, one must consider the use of dynamic WEP keys as well as proprietary MAC address checking or an 802.1x authentication method to fully address wireless LAN security vulnerabilities.

Additional Reading

Held, G., "Wireless Application Directions," *Data Communications Management* (April/May 2002).
Lee, D.S., "Wireless Internet Security," *Data Communications Management* (April/May 2002).

124

Formulating an Enterprise Information Security Architecture

Mollie E. Krehnke, CISSP, IAM and David C. Krehnke, CISSP, CISM, IAM

Introduction

Ours is a connected world, and a dependent world. The condition and livelihood of any organization is dependent on the integrity, availability, and confidentiality of information obtained from or protected from other sources. Today, organizations are at greater risk and their security stance against malicious actors, in the form of individuals, criminal cartels, terrorists, or nation-states, will affect the well-being of many persons, other companies, and perhaps the nation. These organizations often depend upon cyberspace — hundreds of millions of interconnected computers, servers, routers, switches, and fiber-optic cables that allow our critical infrastructures to work.[1]

Threat Opportunities Abound

Individuals and organizations with malicious intent will use any means to disrupt business processes; obtain the data the information systems create, maintain, and transmit; and acquire the power that the information systems and associated networks possess for other unauthorized acts. Malicious actors have the intent (political, economic, national security), the tools (widely available), and the targets (many and well-known vulnerabilities). Malicious actors also have the time and the financial resources necessary to implement attacks. These attacks can have serious consequences, such as disruption of critical operations, causing loss of revenue and intellectual property, or loss of life. Such attacks could use any available cyber resources, including computers located in homes or small businesses to initiate attacks on critical infrastructure organizations — exploiting weaknesses, disrupting communications, hindering defensive or offensive responses, or delaying emergency responders.

Vulnerabilities result from weaknesses in technology and improper implementation and oversight of technological products.[2] The majority of vulnerabilities can be mitigated through good security practices, although such practices must go beyond mere installation, and include proper training, operation, regular patching, and virus updates. The vulnerabilities within an organization can be used to mount an attack against that organization or against other organizations.

Responding to an Increasing Threat

The cyberspace vulnerabilities must be addressed at an individual level and an organizational level. "Each American who depends on cyberspace must secure the part that they own or for which they are responsible."[3]

Likewise, each organization must establish and maintain an effective enterprise information security architecture that contributes to its own security, its employees, customers, business partners — and that of the nation.

The effective deployment of security for an enterprise is dependent on the business functions of the enterprise. To gain business commitment, the security functions determined to be necessary must support the business functions of the organization and provide "added value." The provision of added value in the form of enterprise information security is dependent upon many factors: accurate identification of business functions; configuration and management of the existing and planned resources (e.g., networks and technologies); business and security infrastructures; enterprise business processes; people (employees, business partners, and vendors); physical security of facilities, equipment, and remote sites; and associated security or security-supporting policies and processes. The mere presence of certain security mechanisms will not guarantee an acceptable level of risk for the enterprise. Therefore, an enterprise information security architecture must be defined, installed, monitored, assessed, and upgraded on a periodic basis to ensure that the security architecture is appropriate for the enterprise. The major key to successful implementation of security is the commitment of upper management.

Architectural Design Concepts

Association of Business Functions to Security Services

To add value to an organization's business functions, those functions must be understood. A business will have documentation that presents an overview of those functions. Certain individuals will be good resources as well, and should be delighted to discuss security from an added-value standpoint. Business unit managers who oversee specific lines of business (business domains) and subject matter experts can support the documentation of business functions and provide the business perspective to the sequencing of automated and nonautomated processes to address the business mission. The business functions to be addressed also have to be viewed in light of capital planning, enterprise engineering, and program management.

Exhibit 124.1 presents an approach for enterprise architecture development. If such an architecture exists for the enterprise, then the creation of a security architecture has a firm foundation. Business functions and associated business processes, data and data flows, applications and associated functionality, present technology architecture, business locations, business partners and vendors, and strategic goals to support the business mission may already exist — in some form.

The three-to-five-year target enterprise architecture is a good resource for determining future goals of the organization that will have to be addressed from a security standpoint. Any goals beyond that timeframe will not be as useful for the establishment of an effective information security architecture — technology, customer focus, and external requirements are key drivers in this architecture and they are not easily defined beyond that time with any accuracy.

Association of Enterprise Architecture to Information Security Architecture

The target enterprise architecture can provide answers to the following questions that will be invaluable to the enterprise security architecture initiative:

- What are the strategic business objectives of the organization?
- What information is needed to support the business?
- What applications are needed to provide information?
- What technology is needed to support the applications?
- What is the needed level of interoperability between the data sources and the users of the data?
- What information technology is needed to support the enterprise's technical objective?
- What systems are going to be replaced in the near term? In the long term? What systems are going to be migrated to the new enterprise architecture?
- What risks are associated with the current sequencing plan?
- What alternatives are currently available if funding or resources are delayed?
- What are the budgetary and territorial concerns?

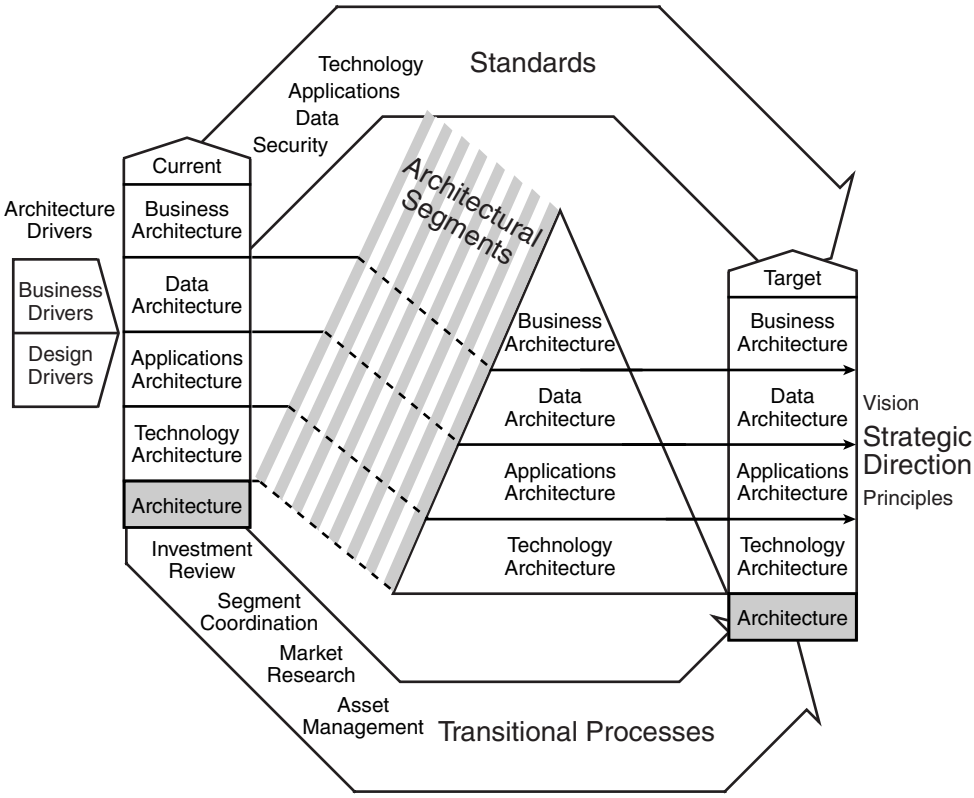


EXHIBIT 124.1 Structure of the Federal Enterprise Architecture Framework. (*Source:* A Practical Guide to Federal Enterprise Architecture, Chief Information Officer Council, Version 1.0, February 2001, Figure 6, Structure of the FEAF Components.)

The enterprise architecture can be managed as "a program that facilitates systematic agency [business] change by continuously aligning technology investments and projects with agency mission needs."[4] There are going to be areas in which the enterprise architecture information, such as data information and flows, can move directly into an enterprise information security architecture as factors in establishing processes and functionality. There will be others, such as the identification of the business areas or information needs with the greatest potential payoff for the enterprise, which will have to be tempered with other security considerations. Although an organization certainly wants to address these high payoff areas in terms of information availability, integrity, and confidentiality, there may be other less "visible" areas that have higher areas of risk that will also have to be appropriately addressed in order to ensure the security of all business functions.

General Enterprise Architecture Principles

Federal agencies are now required to establish an enterprise architecture that will be used to streamline the collection, storage, and analysis of information, and the provision of applicable information to the general public. The process for the identification and documentation of information required to establish a federal enterprise architecture has aspects that can be applied to private industry as well.

Excerpts from the Chief Information Officer (CIO) Council guide[5] provide principles that help in the establishment of a enterprise architecture and, for our purposes, the establishment of an enterprise information security architecture:

- Architectures must be appropriately scoped, planned, and defined based on the intended use of the architecture.
- Architectures must be compliant with the law.
- Architectures facilitate change.
- Architectures must reflect the organization's strategic plan.
- Architectures continuously change and require transition toward the target architecture.
- Target architectures should project no more than three to five years into the future.
- Architectures provide standard business processes and common operating environments.
- The quality of the associated architecture documentation is dependent upon the information obtained from subject matter experts and business owners.
- Architectures minimize the burden of data collection, streamline data storage, and enhance data access.
- Target architectures should be used to control the growth of technical diversity.[6]

Although the CIO architecture model mentions security as a concept[7] that "overlies" the enterprise life cycle, and the Interoperability Clearinghouse, a nonprofit organization that develops architectures,[8] includes security as a domain architecture, the impact that security should have in the establishment of the architecture is not fully presented. The implementation of an enterprise information security architecture requires the establishment of strong, far-reaching business practices that ensure system compliance with the security architecture and needs continuous assessment to enforce compliance (with the full support of senior management). Otherwise, there is no way to assure that the enterprise information security architecture meets the established business needs and functions at an acceptable level of risk.

General Enterprise Information Security Architecture Principles

Objectives of an enterprise information security architecture, in support of the business mission, must include the following:

- Not impede the flow of authorized information or adversely affect user productivity
- Protect information at the point of entry into the enterprise
- Protect the information throughout its useful life
- Enforce common processes and practices throughout the enterprise
- Be modular to allow new technologies to replace existing ones with as little impact as possible
- Be virtually transparent to the user
- Accommodate the existing infrastructure[9]

Inputs to the Security Architecture

Exhibit 124.2 depicts the inputs to the initial process in formulating an enterprise information security architecture. The process should, at a minimum, consider the following inputs:

- Business-related inputs:
  - Business goals and objectives for protecting the organization's business interests, assets, personnel, and the public; and the future direction of the business and supporting information systems
  - Business operational considerations of how the business will operate day to day (e.g., centralized or decentralized approach to security administration)
  - Current business directions and initiatives for the installed information systems and those under development
  - Business information system requirements (e.g., access requirements, availability requirements, business partner connectivity)
  - Business policies and processes defining what is acceptable and what is not acceptable business behavior
  - Business assets to be protected by the architecture
  - Existing infrastructure including a characterization of the current technical environment and what may help or negatively affect information security

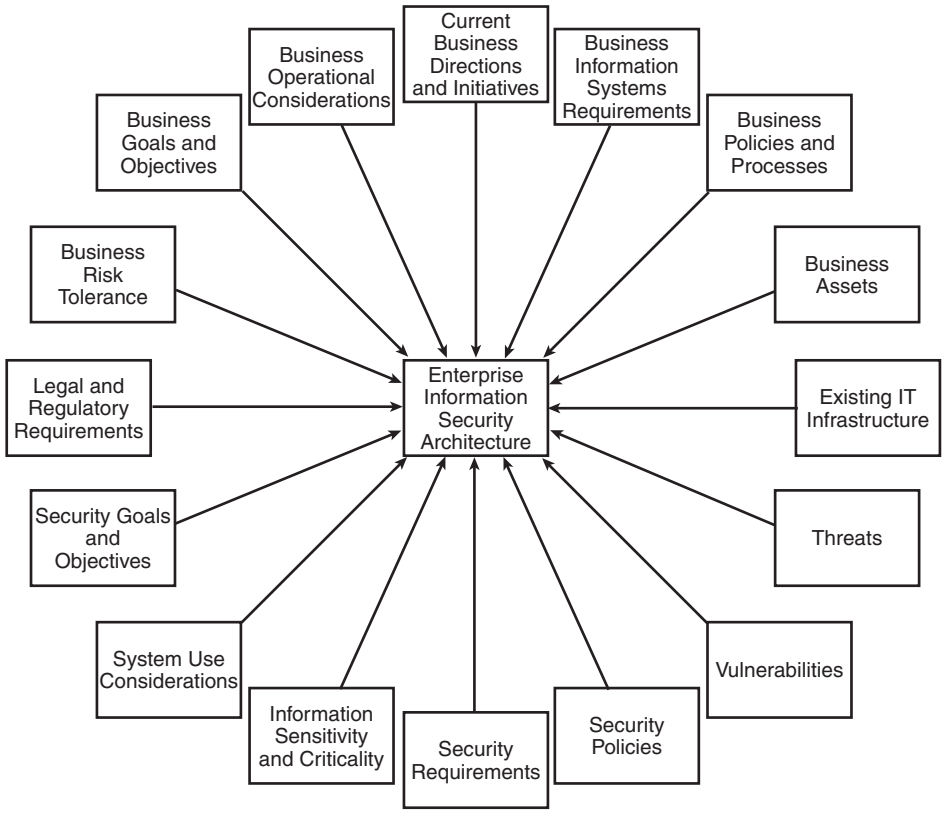


EXHIBIT 124.2 Considerations for formulating an enterprise information security architecture.

  - Business risk tolerance for information disclosure, unauthorized modification and loss, unavailability, downtime due to hackers and viruses, and defaced Web pages
  - Legal and regulatory requirements including laws and regulations such as privacy, basic due care and due diligence, and sentencing guidelines
  - Threats to the existing infrastructure or business operations
  - Vulnerabilities associated with the existing infrastructure or computing operations
- Security-related inputs:
  - Security goals and objectives (e.g., safeguard information assets from unauthorized and inappropriate use, loss, or destruction; protect sensitive information from unauthorized disclosure and manipulation; and protect the availability of critical information)
  - System use considerations including who will use the information systems (employees, contractors), what level of background screening, when (time of day, days of the week), where (office, home, travel), why (inquiries, file updating, research), etc.
  - Sensitivity and criticality of the information to be protected, including the impact due to unavailability or loss
  - Security requirements to protect information, applications, platforms, and networks based on the sensitivity and criticality of the information (e.g., label sensitive media, back up information, store backups off site, encrypt information stored in nonsecure locations or transmitted over untrusted networks)
- Security policies on what is and what is not acceptable security behavior

Moving from Design to Deployment

Building a Secure Computing Environment

As depicted in Exhibit 124.3, a well-defined enterprise information security architecture provides the foundation for a secure infrastructure and a secure computing environment. The building blocks of a secure computing environment include:

- Well-defined enterprise information security architecture, with accountability, deployment strategies, technology, and security services
- Effective information security processes, procedures, and standards, derived from policies, but dealing with specific components and technologies and providing detailed specifications that can be audited
- Effective information security training, including new-hire training; job-related operational training for executives, managers, supervisors, privileged users, and general users; and periodic awareness training
- Effective information security administration and management, including configuration management, information resources management (IRM), hardened platforms with the latest security patches and virus signature files, virus scanning, vulnerability scans, intrusion detection, penetration testing, logging, alarms, and reviews of common vulnerabilities and exposures (CVEs)
- Aggressive information security assurance, including certification, accreditation, self-assessments, inspections, audits, and independent verification and validation (IV&V)
- Secure infrastructure, including DMZ, routers, filters, firewalls, gateways, air gaps, protected distribution systems (PDSs), virtual private networks (VPNs), secure enclaves, and separate test environments
- Secure applications, including well-designed, structured, and documented modules; software quality assurance; code review; file integrity checking or change detection software, including products such as Tripwire and Advanced Intrusion Detection Environment (AIDE); and access based on the principles of clearance, need-to-know, and least privilege
- Secure information, including encryption, backups, and integrity checking software

Information Security Life Cycle

Exhibit 124.4 indicates how the information security life cycle interacts with the foundation and core components of an information security program. As the outer ring illustrates, organizations should continuously perform the following functions during the information security life cycle:

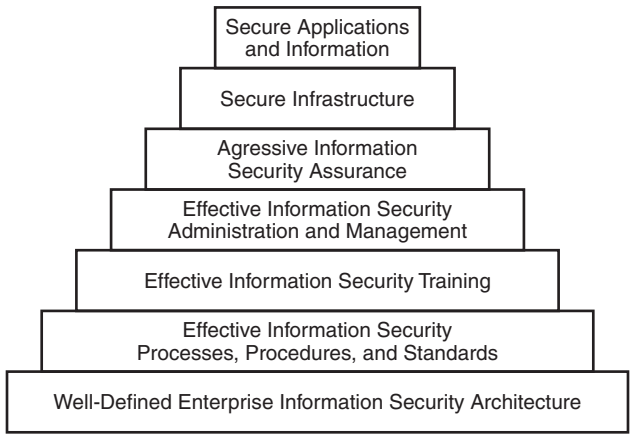


EXHIBIT 124.3 Building blocks of a secure computing environment.

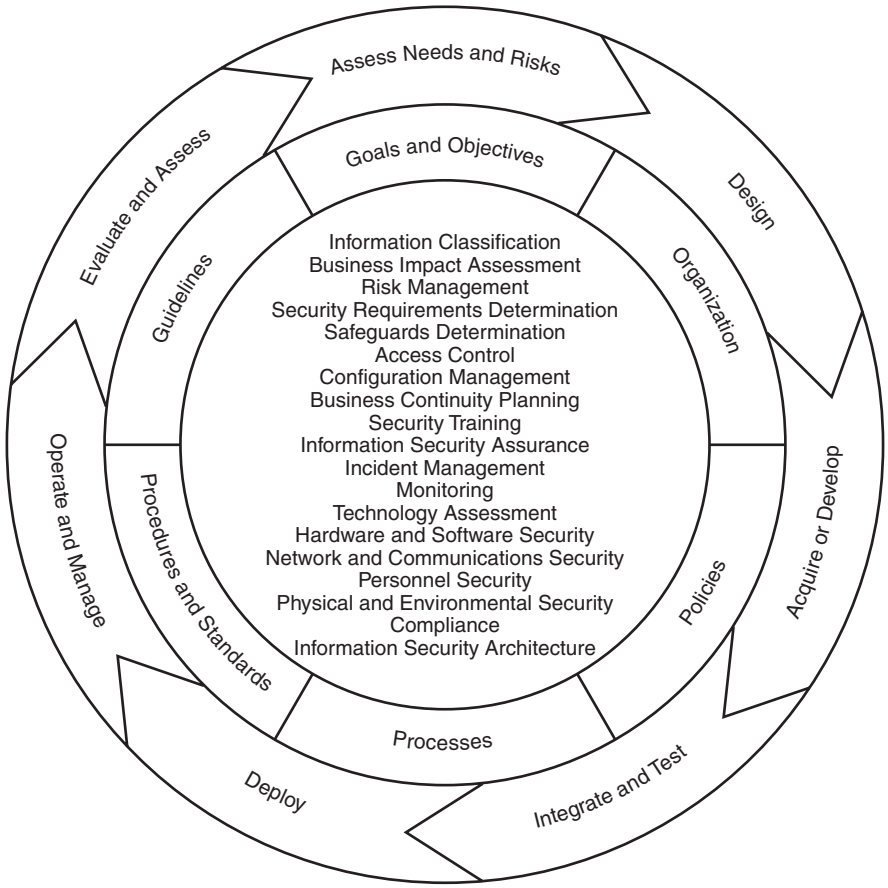


EXHIBIT 124.4 Information security lifecycle and the information security program.

- Assess business security needs and the risks to the organization
- Design security solutions to appropriately address the assessed risks
- Acquire or develop security solutions
- Integrate and test security solutions
- Deploy security solutions
- Operate and manage security solutions
- Evaluate and assess security solutions to assure their effectiveness

The organization can perform these functions directly or outsource them, and ensure they are implemented effectively. These functions should be performed continuously because security is an ongoing process, not a one-time destination. Business, technology, risk, and organization structure are not static.

The inner ring illustrates the foundation or essential ingredients of an information security program:

- *Goals and objectives:* Confidentiality and possession, integrity and authenticity, availability and utility, accountability, non-repudiation, and assurance
- *Organization:* Full-time and *ad hoc* personnel identified to implement the information security programs
- *Policies:* High-level management instructions that support an enterprisewide information security program that incorporates prudent practices from industry and government
- *Processes:* Methodologies that support the information security policies and cost effectively implement information security in the enterprise

- *Procedures and standards:* Detail components, technologies, and step-by-step actions that support the policies and processes
- *Guidelines:* Recommended activities to provide a more secure environment

The inner elements are the functional core components of an information security program:

- *Information classification:* The process and consulting support by which the sensitivity of each application is determined.
- *Business impact assessment:* The process and consulting support by which the criticality of each application is determined.
- *Risk management:* The process and consulting support for the identification and assessment of assets, threats, vulnerabilities, and the resulting risks and their successful mitigation, transfer, or acceptance.
- *Security requirements determination:* The process and consulting support for identifying the information security requirements given the sensitivity, criticality, and risks.
- *Safeguards determination:* The process and consulting support for identifying information security safeguards or controls that will satisfy the security requirements.
- *Access control:* The process of identification and authentication of users, maintaining audit records of their access, and enforcing individual accountability that prevents unauthorized access to information systems.
- *Configuration management:* The rigorous management of the change process that provides hardware and software integrity, and change and version control.
- *Business continuity planning:* The process and consulting support that implements effective planning for continued business operations under all conditions and situations.
- *Security training:* The operational and awareness guidance that ensures all employees are trained in the security aspects of their jobs and their associated security responsibilities, and the secure, appropriate use of information systems and data.
- *Information security assurance (also known as certification and accreditation):* The formal security evaluation and management approval process that ensures the information system is protected at a level appropriate to its sensitivity and criticality classifications; identifies the controls that satisfy the security requirements, and are documented in a security plan. Determines the residual risk before the information system is put into production as it is, and periodically reviewed over the life of the information system. Periodically tests and evaluates the effectiveness of protection mechanisms, based on current threats and vulnerabilities.
- *Incident management:* The process and consulting support that ensures appropriate actions for detecting, reporting, and responding to information security incidents. Receives and tracks information security incident reports through resolution, escalates serious incidents, and incorporates "lessons learned" into ongoing security awareness and operational training programs.
- *Monitoring:* The monitoring of logs and activities to verify the security stance, ensure appropriate resource use, and defend resources from attack.
- *Technology assessment:* The review, evaluation, and recommendation of advanced security technologies. Evaluates infrastructure and commercial-off-the-shelf (COTS) products for common vulnerabilities and exposures (CVEs).
- *Hardware and software security:* The procurement, configuration, installation, operation, and maintenance of hardware and software in a manner that ensures information security. Includes platform hardening and software integrity checking.
- *Network and communications security:* Perimeter protection, intrusion detection, vulnerability scans, penetration testing, remote access management, and control of modems. Determines the criteria for the evaluation of firewalls, recommends encryption solutions, determines when secure enclaves are required, and provides consulting support for the review of network connectivity requests.
- *Personnel security:* Identifies sensitive positions and ensures individuals assigned to those positions have an appropriate clearance. Includes information security in job descriptions, and through performance appraisals holds individuals accountable for carrying out their information security responsibilities and for their actions.
- *Physical and environmental security:* Protects hardware, software, and information through physical and environmental controls.

- *Compliance:* Administrative inspections, reviews, evaluations, audits, and investigations for the purpose of maintaining effective information security. Consulting support on best practices from industry and government on remedial action to address any significant deficiencies. Confiscation and removal of unauthorized hardware and software, and hardware, software, and data required for use as evidence of wrongdoing.
- *Information security architecture*: The framework for information security and the road map for implementation to ensure the confidentiality, integrity, and availability of applications and information.

Defense-in-Depth for a Secure Computing Environment

Exhibit 124.5 depicts the requirements for a secure computing environment. The lack of security in any one of these components is going to negatively impact the security of the computing environment. If there is no policy, there can be no uniform management direction on how to protect the business, its operations, its people, and its information. If there are no processes and procedures with associated standards, implementation of policy will be based on an individual's interpretation of policy — which is likely to vary from person to person. If there is no physical security, then logical and administrative controls can be easily circumvented without being discovered. The lack of environmental controls can bring down the enterprise and cause more destruction than a malicious agent. If there is inadequate personnel security, the likelihood of insider threat increases dramatically and the impact may not be detected for a significant period of time. The need for communications and network security is obvious; we live in a connected world. However, the unapproved use and unknown presence of a modem or wireless network access points will circumvent

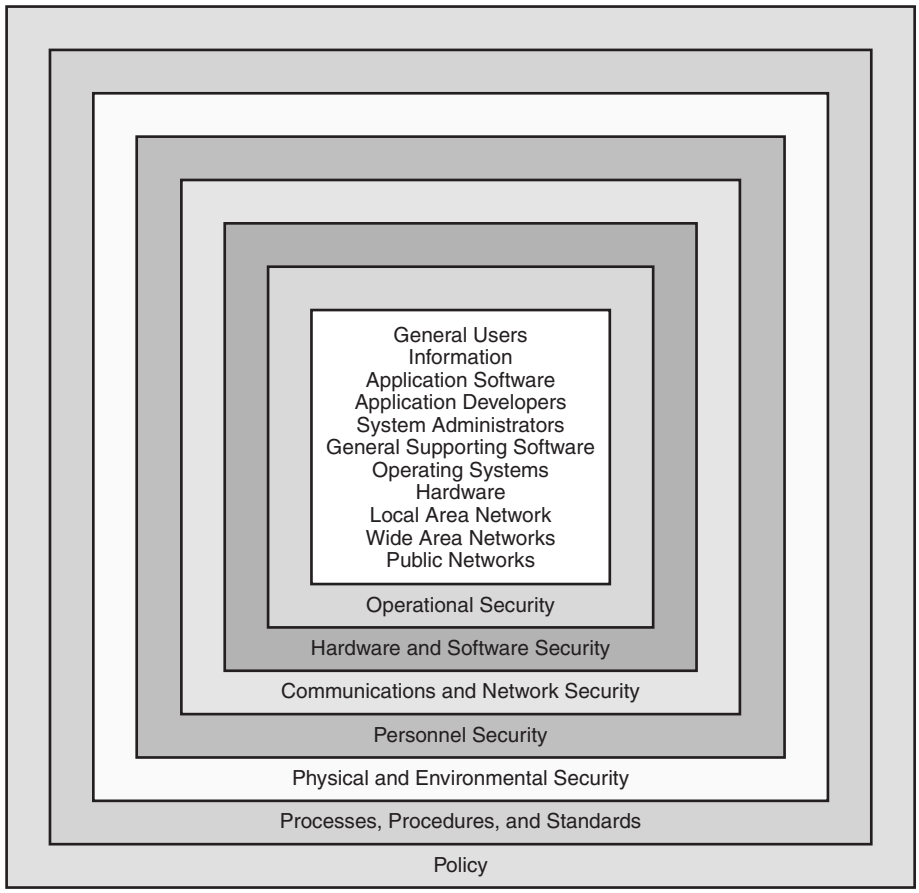


EXHIBIT 124.5 Defense-in-depth.

firewall protection. Hardware controls must be in line with the equipment functionality, e.g., servers must be hardened before deployment if it is going to be effective. Software and its associated controls must be up to date, including patches and updated virus signature files. Employees, contractors, vendors, and visitors must know what is expected of them to support enterprise information security. Public networks, although vital to many business operations, must be viewed as untrusted components of the enterprise architecture and handled appropriately. Wide area networks (WANs) and local area networks (LANs) have certain operational requirements that must be implemented to ensure information confidentiality, integrity, and availability. Hardware must be assessed on its ability to perform the required functions, and must be protected so it cannot be reconfigured to perform unauthorized functions. Software must be licensed, purchased from a trusted source, and assessed to ensure it does not contain malicious code even if it is shrink-wrapped. System administration and application developers must be trusted personnel with the appropriate clearances who have been trained to perform their job responsibilities accurately and effectively. Application software must be accurately designed, developed, and implemented to protect information and the business environment. Information is the lifeblood of the organization and must be protected from unauthorized disclosure, while being made available when required in an accurate, usable, and complete format. General users represent a significant threat to the secure computing environment, accidentally or with malicious intent. The actions of users must be controlled, and users must be trained in secure operations and use of information and computing and communications resources. The user is the weakest component of the secure computing environment, and carelessness or social engineering can result in established controls being circumvented. Therefore, defense-in-depth must also include checks and balances, with multiple security functions and associated components to address the security requirements. The standardization of security components is represented in this chapter as information security services.

Defining the Enterprise Information Security Architecture

Information Security Services

Information security services provide the enterprise information security architecture with standard methods to support the integration and implementation of information security across the organization infrastructure. These services must be standardized, shareable, and reusable. Information security services include people and technology services.

- *Accountability:* Associates each unique identifier (e. g., user account or log-on ID) with one and only one user or process to enable tracking of all actions of that user or process.
- *Assurance:* Provides a formal information security evaluation and management approval process to ensure information applications and the supporting infrastructure are protected at a level appropriate to their sensitivity and criticality.
- *Authentication:* Verifies the claimed identity of an individual, workstation, or process.
- *Authorization:* Determines whether and to what extent access should be granted to specific information, applications, and information systems.
- *Availability:* Ensures information, applications, and information systems will be accessible by authorized personnel or other information resources when required.
- *Confidentiality:* Ensures that information is not made available or disclosed to unauthorized individuals, entities, or processes.
- *Identification:* Associates a user with a unique identifier by which that user or process is held accountable for the actions and events initiated by that identifier.
- *Integrity:* Ensures the correct operation of applications and information systems, consistency of data structures, and accuracy of the stored information.

Information Security Functions

Each information security service consists of one or more security functions that further identify and define the security action or process needed to secure the information and information systems. Examples of such

information security functions include, but are not limited to, authorization, identification, authentication, accountability, risk assessment, confidentiality, encryption, physical access control, logical access control, digital signatures, integrity, intrusion protection, virus protection, non-repudiation, availability, security administration, audit logging and reviews, information security assurance, incident handling, monitoring, and compliance.

Enterprise Information Security Services Matrix

Exhibit 124.6 summarizes information security services and their related security functions. The exhibit is organized as follows:

- *Information Security Service.* Names the information security service that addresses one or more specific security needs or requirements identified to secure information and information systems and comply with applicable laws, statutes, regulations, policies, and best industry practices. Securing information and information systems may require the use of one or more information security services.
- *Security Function.* Lists the security functions that comprise an information security service.
- *Security Function Description.* Provides a brief description of the security function.
- *Vehicle.* Enumerates the mechanisms, processes, controls, and technologies that support, contribute, and implement the named information security service. Each information security service may be implemented through multiple processes and technologies.

Assessing the Enterprise Information Security Architecture

Controlling the Growth of Technical Diversity

The priorities established for the enterprise architecture will have to address all enterprise information security considerations. On the flip side, security projects, like business projects, will have to be reviewed in light of several considerations:

- *Business alignment:* Does the project support established strategic plans, goals, and objectives?
- *Business case solution:* What is the impact on the organization's information technology and business environments?
- *Sequencing plan:* Is the proposed investment consistent with the sequence (plan) and priorities established to reach the target architecture?
- *Technical plan compliance:* Does the proposed project comply with the enterprise standards and the architecture levels?

Ensuring Continued Support by Addressing Design Principles

The establishment of an enterprise security architecture is a significant undertaking. The perceived (or actual) complexity of the product could entice the viewer to assume that the architecture can successfully support the design, development, operation, and retirement of an information system. Periodically throughout the implementation of the architecture, it is good to look at the model in light of a system that supports a business function and see if it complies with security principles that support the system throughout its life cycle: initiation, development or acquisition, implementation, operation and maintenance, and disposal. Have the following design principles been addressed?

- Establish a sound security policy as the "foundation" for design.
- Treat security as an integral part of the overall system design.
- Clearly delineate the physical and logical security boundaries governed by associated security policies.
- Reduce risk to an acceptable level.
- Assume that external systems are insecure.
- Identify potential trade-offs between reducing risks and increased costs and decreases in other aspects of operational effectiveness.
- Implement layered security (ensure no single point of vulnerability).

EXHIBIT 124.6 Enterprise Information Security Services

Information Security Service	Security Function	Security Function Description	Vehicle
Accountability	Non-repudiation	Assures the sender cannot deny he sent the message and recipient cannot claim that he received a different message	Digital signature and certificates
	User deterrence	Places restraint on deviant activities by increasing the likelihood of identification and prosecution of personnel conducting such activities	Security awareness training Operational security training Policy, processes, and procedures
Assurance	Data designation	Determines the sensitivity and criticality of information and information systems	Data element assessment
	Monitoring	Provides surveillance of the activity being performed within the information systems as well as at its boundaries; the surveillance service is carried out on networks and on servers/hosts: network monitoring and host/server-based monitoring	Intrusion detection systems (IDS) Host-based IDS
	Intrusion detection	Detects attempts at system break-ins, behavior patterns, and anomalies with respect to activities at the boundaries of the information system (e.g., network, mainframe, or other device)	IDS
	Malicious code protection	Security code review provides assurance that the information system does and will only execute authorized operations that ensure, preserve, and maintain the integrity of the system and all the information systems accessed	Security code review
		Virus protection monitors, analyzes, and protects the information resource from possible virus attacks	Virus scanning Pattern distribution
	Security administration	Implements management constraints, operational procedures, and supplemental controls established to provide adequate protection of an information system	Configuration management Information resource life cycle Database administration
	Acceptable use monitoring	Ensures information resources will be used in an approved, ethical, and lawful manner to avoid loss or damage to operations, image, or financial interests	Audit logging Monitoring Content filtering
	Compliance	Reviews and examines the records, procedures, and activities to assess the information system security posture and ensure adherence with established criteria	Audit logging Monitoring Content filtering Inspection Independent assessment Penetration testing
	Audit	Provides the information systems with reviews as well as examination of records and activities to test for adequacy of the security controls, compliance with established policies, and operational procedures, and possibly recommends changes to policies and procedures	Audit logging Inspection Independent assessment

EXHIBIT 124.6 Enterprise Information Security Services (continued)

Information Security Service	Security Function	Security Function Description	Vehicle
	Assessment of business impact	Determines the level of sensitivity, criticality, recovery time objective (RTO); the potential consequences due to information and information system unavailability or loss; and the identification of security requirements	Business impact assessment
	Assessment of risk	Identifies vulnerabilities, threats, likelihood of occurrence, potential loss or impact, expected effectiveness of security measures, and residual risk for an information resource	Risk assessment COTS vulnerability assessment
	Security testing and evaluation	Provides support for testing to determine if all the required security controls and countermeasures described in the security plan are in place and functioning correctly	Security test and evaluation plan
	Certification	Establishes the extent to which the information system meets a specified set of security requirements	C&A process
	Accreditation	Provides support to management in their formal acceptance of the residual risk for operating the information system and approval to deploy	C&A process
	Enclaving	Allows for configuration of special network areas that provide additional protections and access controls to secure information resources	Enclaving process Firewalls IDS Vulnerability scans
	Network connectivity	Protects network and communications infrastructure by managing network connectivity	Network connectivity process
	Penetration testing and vulnerability scans	Checks the robustness and effectiveness of the boundary countermeasures implemented for a given information resource	Vulnerabilities test plan
	Physical security	Identifies specific physical weaknesses, vulnerabilities, and threats for a facility, network, enclave, and information system and implements countermeasures	Site security review System security plan Locks, mantraps, locking turnstiles Guards Fences Lighting CCTV Motion detectors
	Environmental security	Identifies specific environmental weaknesses, vulnerabilities, and threats to a facility, network, enclave, and information system and implements countermeasures	Redundant power UPS Backup diesel generators Redundant telecommunications Backup HVAC
	Personnel security	Identifies sensitive positions and provides the structure to ensure personnel are cleared and their information security responsibilities are defined and included in their performance evaluation	Personnel clearances Job descriptions Performance appraisals Sanctions Conditions of continued employment

EXHIBIT 124.6 Enterprise Information Security Services (continued)

Information Security Service	Security Function	Security Function Description	Vehicle
			Job rotation
	Incident management	Provides security incident handling and analysis	Incident reporting process
Authentication	Authentication	Verifies the claimed identity of an individual, workstation, or originator	Passwords and PINs Biometrics Smart cards Tokens Digital certificates
Authorization	Authorization	Determines whether and to what extent personnel should have access to specific information and information systems	User registration and authorization management
Availability	Fault isolation	Hardware: Allows the detection of hardware malfunction and the identification of the component that caused it	System alerts Network management systems/ protocols
		Software: Allows the detection of software malfunction and the identification of the component that caused it	Audit logging Network management systems/ protocols
	Contingency planning	Provides contingency planning for information and information systems, personnel, and the facilities that house them	Emergency plan Contingency plan Facility recovery plan Personnel evacuation plan
Confidentiality	Confidentiality	Ensures information is not disclosed to unauthorized individuals, entities, or processes; confidentiality applies to hardcopy and electronic media in storage, during processing, and while in transit	Eradicate media Encryption Secure storage Key management Information classification Screen savers Physical access controls Physical access controls Public key infrastructure Logical access controls Separation of duties
Identification	Trusted identification	Associates a user with a unique identifier (e.g., user account or log-on ID) by which that user is held accountable for the actions and events initiated by that identifier	Unique user identifier
Integrity	Data integrity	Ensures the consistency of data structures and accuracy of transmitted or stored information	Hashing Checksum Digital signature
	Information system integrity	Ensures the correct operation of information system	System development methodology Independent security testing and evaluation Configuration management Session management Screen savers Test environment restrictions Server hardening

- Implement tailored system security measures to meet organizational security goals.
- Strive for simplicity.
- Design and operate an information technology system to limit vulnerability and to be resilient in response.
- Minimize the system elements to be trusted.
- Implement security through a combination of measures distributed physically and logically.
- Provide assurance that the system is, and continues to be, resilient in the face of expected threats.
- Limit or contain vulnerabilities.
- Formulate security measures to address multiple overlapping information domains.
- Isolate public access systems from mission-critical resources (e.g., data, processes).
- Use boundary mechanisms to separate computing systems and network infrastructures.
- Where possible, base security on open standards for portability and interoperability.
- Use common language in developing security requirements.
- Design and implement audit mechanisms to detect unauthorized users and to support incident investigations.
- Design security to allow for regular adoption of new technology, including a secure and logical technology upgrade process.
- Authenticate users and processes to ensure appropriate access control decisions both within and across domains.
- Use unique identities to ensure accountability.
- Implement least privilege.
- Do not implement unnecessary security mechanisms.
- Protect information while it is being processed, in transit, and in storage.
- Strive for operational ease of use.
- Develop and exercise contingency or disaster recovery procedures to ensure appropriate availability.
- Consider custom products to achieve adequate security.
- Ensure proper security in the shutdown or disposal of a system.
- Protect against all likely classes of "attacks."
- Identify and prevent common errors and vulnerabilities.
- Ensure that developers are trained in how to develop secure software.[10]

Conclusion

Benefits of Architectures

The profit margin for most businesses is small, and the reduction of costs is vital to the success of the business. The enterprise security architecture can "reduce the response time for impact assessment, trade-off analysis, strategic plan redirection, and tactical action" with regard to security.

Some additional benefits are:

- Support for capital planning and investment management.
- Capturing a "snapshot in time" of business and technology assets.
- Provision of a strategy for systems and business migration.
- Help to mitigate risk factors in enterprise modernization.
- Identification of possible sites for innovative technology deployment.
- Support for key management decision making throughout the organization.[11]

Some direct cost-saving benefits include:

- Discounts on new products through bulk purchasing.

- Capital planning assistance from department CIO offices to ease the paperwork burden on division CIOs.
- Better career opportunities for information technology and security workers because their skill sets can be used on any of the standard systems that will be deployed throughout the department (enterprise).
- Increased ability to provide standardized training with a higher return on investment, because the number of people being trained by the same curriculum is greater for all levels of training, including users, technical support, and administrators.
- Ability to allocate human resources to areas other than their usual assignments to address key security concerns or incidents.

Helpful Hints from a Security Architecture Practitioner

The security architect is becoming a key function in many organizations, and functions as "the 'corporate clutch,' providing an interface between the security policy-makers and those tasked with providing information systems solutions to businesses." Concepts supporting a successful deployment and utilization of an enterprise security architecture include:

- Available architectural frameworks will have to be modified to adequately address security at the enterprise level.
- Avoid product focus (and resulting product wars) in the establishment of the security architecture.
- Deviations from initial security requirements must be managed to ensure compensating controls are used to minimize risk.
- Architectural documentation must be current and complete, or decisions will be made on obsolete information and ultimately require reworking.
- Documentation is a key deliverable of the architecture team; the lack of it can be costly — more so than the personnel costs associated with creating and maintaining the documentation.
- Project management supports the timely completion of tasking and deliverables.
- Publish all the information that can be provided to all members of the architectural team to facilitate their understanding of the security target architecture.
- Risk assessments are a valuable tool for any security architecture initiative and help to support a responsive architecture that avoids obsolescence and addresses business needs.
- Use business cases as a forum to assign costs to risks, focus the team on providing cost-effective solutions, and to contrast the costs of alternative (less desirable) solutions.
- Make presentation of architectural concepts and associated requests to senior management.
- Architecture supports policy and serves as a policy advocate, working to shape security requirements into practical solutions.[11]

The Bottom Line

The enterprise information security architecture is a complex model that incorporates business functions, technology, security policy, physical security, configuration management, risk management, contingency planning, users, and business partners and vendors. Generally speaking, all of these concepts will have to be applied to every business function or application, and the justification for the associated resources will have to be presented to senior management. Business functions have to be linked to security functions, and then added value has to be presented in a way that makes sense to senior management and positively affects the business bottom line.

Notes

1. The National Strategy to Secure Cyberspace, Department of Homeland Security, February 2003, p. vii.
2. The National Strategy to Secure Cyberspace, Department of Homeland Security, February 2003, p. xi.
3. The National Strategy to Secure Cyberspace, Department of Homeland Security, February 2003, p. 11.
4. A Practical Guide to Federal Enterprise Architecture, Chief Information Officer Council, Version 1.0, February 2001, p. 40.

5. A Practical Guide to Federal Enterprise Architecture, Chief Information Officer Council, Version 1.0, February 2001.
6. A Practical Guide to Federal Enterprise Architecture, Chief Information Officer Council, Version 1.0, February 2001, Appendix E, Sample Architectural Principles.
7. A Practical Guide to Federal Enterprise Architecture, Chief Information Officer Council, Version 1.0, February 2001, p. 8.
8. ICHnet.org Enterprise Architecture Reference Model, Achieving Business-Aligned and Performance-Based Enterprise Architectures: An Interoperability Clearinghouse White Paper on Enterprise Architecture Frameworks and Methods, Interoperability Clearinghouse, May 22, 2002, p. 4, available at http://www.ICHnet.org.
9. Hare, C., Firewalls, Ten Percent of the Solution: A Security Architecture Primer, this volume.
10. Zyskowski, J., Building for the Future: Enterprise Architecture Emerges as a Blueprint for Better IT Management, *Federal Computer Week,* January 2, 2002.
11. Scammell, T., Security Architecture: One Practitioner's View, *Information Systems Control Journal,* 1, 24–28, 2003.

125

Security Architecture and Models

Foster J. Henderson, CISSP, MCSE and
Kellina M. Craig-Henderson, Ph.D.

> He is like a man who built a house, and digged deep, and laid the foundation on a rock: and when the flood arose, the stream beat vehemently upon that house, and could not shake it; for it was founded upon a rock. But he that heareth, and doeth not is like a man that without a foundation built a house upon the earth; against which the stream did beat vehemently, and immediately it fell and the ruin of that house was great.
>
> **— Luke 6:48–49, The Bible, King James Version**

As this passage illustrates, a strong foundation has been akin to protection from adversity since the beginning of time. It should not be surprising then that information security professionals must have a good foundation to implement successful security architecture. Following are the areas designated as the cement for our "virtual foundation." A commitment to successful security architecture requires a clear understanding of issues involving:

- Technology
- Environment
- Software

What follows is initially a brief description of the components to this "tripartite" conceptualization of the virtual foundation. This in turn is followed by a more detailed discussion of exactly what the information security professional must know about each component, as well as the interactive effects of each.

Sounds easy; so why are more people *not* implementing successful security architecture? There are probably a number of reasons, but when one considers that architecture involves "the manner in which the components of a computer or computer system are organized and integrated,"[1] the answer should be fairly obvious. Security involves a very fine synergy that represents the interaction between software, technology, and the environment.

No IT system can be secured unless you unplug it and have "Fort Knox" security protecting it. Security is not only anti-virus software (insert your favorite vendor name) and a firewall. Importantly, people and policy must be factored in as well. And, with respect to the latter, a policy that is too strict, or that does not integrate seamlessly, or is not transparent to its user, is one that will be circumvented, ignored, or not supported.

Technology is multifaceted, and can be thought of as Intel, AMD, Motorola, and RISC chip architectures, wireless standards, Voice-over-IP, biometrics, smart card, IPv4, IPv6, etc. Each one has it advantages, disadvantages, and unique limitations. For example, a few years ago it was common knowledge among IT professionals that if your business operations required performing graphic-intensive work (such as computer-aided design), then you chose the Motorola chip (found in Apple computers) over the Intel chips (found in IBM-compatible personal computers [PCs]).

Environment is the second bullet in our initial outline. However, it is arguably the hardest one to tackle. Here, "environment" refers to the people, business operations, and risks, as well as the threats to your security

architecture or model. We incorporate policy to change our business environment. If the policy is properly implemented, we can expect that the people in the environment will be influenced and guided by it. For example, think of the way in which the air conditioner (AC) modifies the environment of the office, the home, or the car. Here, the AC represents a "policy" to the extent that it changes the environment. The best way to ensure that the environment is up to par is to perform an information security (InfoSec) risk assessment. By not performing one, you cannot or will not understand the environment in which a business operates. You will also be able to identify what environmental threats are lurking out there, such as insiders (i.e., disgruntled employees), hackers, and social engineers. Performing a business impact analysis will enable you to identify the critical practices and tasks essential to a business' survival.

Information Assurance

Information assurance is a term you now see a lot in publications, or job postings on the Internet or in newspapers — or you may even have heard it tossed around at professional meetings. So, what is information assurance? Information assurance consists of the following five areas:

1. *Integrity:* This refers to the quality or condition of being complete or unaltered, i.e., protecting information from unauthorized alterations or destruction.
2. *Confidentiality:* This has to do with having the assurance that the information is not disclosed to unauthorized persons, processes, or devices.
3. *Availability:* Information resources must be available and accessible to its user(s) in a timely manner.
4. *Authentication:* This entails validation and verification of the user and involves determining whether the user should be granted access.
5. *Non-repudiation:* This occurs when the sender is provided with proof of delivery, and the recipient is provided with proof of the sender's identity. It assures that neither party can deny possession of the data at a later time.

Not surprisingly, information assurance should be considered a requirement for all systems used to enter, process, store, display, or transmit national security information.[2] What is perhaps the easiest way to think about information assurance is to think of it as the process that ensures that the correct, unaltered information always gets delivered to its intended and authorized recipient(s) at the correct place and time. The U.S. government, it could be argued, is more concerned with confidentiality, integrity, and availability than is the commercial sector, whose primary focus is availability and integrity. An understanding of the information assurance concept will enable you to determine which solution is best for your environment.

Software Applications

Software refers to the set of instructions that cause the hardware to carry out specific physical tasks. Within this context, "software applications" refers not only to the obvious, but it also refers to "anti-virus," "mobile code," "malicious logic," as well as the various popular operation systems and more. Hopefully, you get the picture.

If you are thinking that what we have just outlined to discuss in this section is daunting, you are correct. But do not despair. At the end of this section you should have a firm grasp of the requisite concepts and ideas to successfully implement security architecture. We will discuss concepts, security practices, preventive, detective, and corrective controls (i.e., the environment), equipment, platforms, networks (i.e., technology), and applications (i.e., software) necessary to ensure information assurance. At various points you will note that the discussion will necessarily reflect the interactive nature of technology, environment, and software. For example, although we begin by discussing aspects of technology, this invariably entails a discussion of software.

Technology

Address Space

Address space refers to the set of all legal addresses in memory for a given application. The address space represents the amount of memory available to a program.[3] By using a technique called *virtual memory* or *virtual storage,* address space can be made larger than primary storage (i.e., RAM; primary storage is the main

memory assessed by the CPU).[4] Think of it this way: FJH is a National Football League fan who plans to see his favorite team, the Dallas Cowboys, at Texas Stadium, which has 65,846 seats.[5] Think of each seat as representing an address in memory. In his fantasy, FJH purchases an entire row of seats in section 28A, directly behind the Cowboys' bench. Think of the actual purchase of the row of seats as a program running in physical memory, which is the stadium. Imagine that, after a sensational season (yes, we said "imagine"), the Cowboys host the NFC Championship game at Texas Stadium, and tickets are sold out. So FJH goes to the local sports bar to watch the televised game on the big screen. The sports bar has a seating capacity of 200. Taking this metaphor a step further, this is represented by the hard drive. To tie all of this together, think of the combination of seating at Texas Stadium (i.e., the physical memory) and that of the sports bar (i.e., the hard drive) as making up virtual storage.

To understand when this process is used, its helpful to describe some related terms. To begin with, keep in mind that an operating system accesses virtual memory when it detects that physical RAM is close to being depleted. Once that limit has been reached, swapping — the process whereby information is transferred from RAM to secondary storage — begins. In contrast, paging is the process of moving information from the input/output device to primary storage. The operating system (OS) has to keep track of all of this movement. A good metaphor for an OS is the conductor of a symphony orchestra. Just as the conductor must account for and direct the movements of each musician, so too must the operating system keep track of all movement between primary, secondary, and virtual storage. Consequently, address space, which can consist of virtual storage, "includes the range of addresses that a processor or process[6] can access, or at which a device can be accessed."[7] Each process will have its own address space, which may be all or a part of the processor's address space. For example, to better understand address space, below is a list of common devices that should look familiar to you to demonstrate address space. It is a list of the most common interrupt request lines (IRQs [i.e.]) and includes the items listed in Exhibit 125.1.

Types of Addressing

The Texas Stadium example of address space pertains to physical addressing. It is an actual location. Relative addressing involves an expressed location from a known point. For example, imagine that you have ordered something from Amazon.com that will be shipped via United Parcel Service (UPS) to your address at 1 Main Street. You know that you will not be home for the delivery, so you leave a message for the driver to deliver the package to your next-door neighbor (3 Main Street). So the address to which the package is actually delivered is 3 Main Street.

Logical addressing is a little more complicated. It is the opposite of physical addressing; its location involves the translation of the physical address. Keep in mind that addressing does not apply to memory only, as is the case in programming, but it can also refer to mass storage as well. Examples include the file allocation table (FAT), the new technology file system (NTFS), or the compact disc file system (CDFS).

EXHIBIT 125.1 Interrupt Request Lines (IRQ)

IRQ 0	System timer
IRQ 1	Keyboard
IRQ 2	Cascade interrupt for IRQ 8–15
IRQ 3	COM 2: 2nd serial port
IRQ 4	COM 1: 1st serial port
IRQ 5	Sound card
IRQ 6	Floppy disk controller
IRQ 7	1st parallel port
IRQ 8	Real-time clock
IRQ 9	Open interrupt
IRQ 10	Open interrupt
IRQ 11	Open interrupt
IRQ 12	Mouse
IRQ 13	Coprocessor
IRQ 14	Primary IDE channel
IRQ 15	Secondary IDE channel[a]

[a] See broadbandreports.com

As you probably know, a central processing unit (CPU) is the heart of the computer. Although CPUs are made by various manufacturers, a few commonly known ones include Intel's Pentium 4, AMD's Athlon, and the PowerPC G4 chips.[8] Both the CPU and bus (the internal components of the CPU that are wired to the primary storage) are physical assets. Consequently, we say that physical addressing is used.[9] Because software is virtual or logical, relative and logical addressing is used. For example, think of using Excel to run a large spreadsheet. The phone rings; after the call has terminated, you return to your spreadsheet and ask yourself, "Which cell am I currently working in?"

Memory

RAM was discussed briefly in the section on address space, and it refers to volatile memory. The term "volatile" is an apt one given that once the power is turned off, all information held in RAM is lost. Nonvolatile memory is the opposite — when power is turned off, the information contained in the memory space is still there. A good example of nonvolatile memory is read-only memory (ROM), which is used in laser printers (the fonts are actually stored in ROM), in calculators, and in portions of the PC that boots the computer.[10] In addition, there is programmable read-only memory (PROM), erasable–programmable read-only memory (EPROM), as well as electrically erasable–programmable read-only memory (EEPROM).

What is the difference between the different types of memory? PROM is blank memory where a set of instructions that have been recorded cannot be used again; EPROM is like PROM, but with instructions that are erased by ultraviolet light. In contrast, EEPROM is PROM with an electric charge that is used to erase the set of instructions.

By the way, have you ever performed an update for a basic input/output system (i.e., BIOS) from a vendor with the latest update, or upgraded your modem with the latest vendor software? Or, have you changed the personal identification number (i.e., PIN) on a smart card? If you have answered "yes" to any of these questions, then you have most certainly had some experience with flash memory. And, guess what? Another name for EEPROM is flash memory. When programs are stored in them, this family of ROM products is also called *firmware,* which refers to the combination of hardware and software.

While we are still discussing the many aspects of memory, it is worth mentioning cache. Cache refers to the reserved section of main memory for high-speed reading and writing of instructions. When data is found, it is called a "hit" and a "miss," depending on whether the information is maintained in cache.

Why are we spending so much time discussing memory and addressing? The easy answer is that some viruses propagate in memory. The more complex answer has to do with the fact that buffer overflow attacks involve sending a set or block of instructions that overflows the set address space of the memory. A few blocks of a malicious code slip in at the tail end of a program being executed, for example, in a privileged state. Buffer overflows occur when programs do not adequately check for the appropriate length in value, and consequently, the malicious code gets executed. Because there is more input than expected, it spills into another program waiting to be executed by the CPU.[11]

For example, Sun Microsystems' Java Virtual Machines executes in memory or in temporary files in various operating systems. Java will run on just about anything that has storage space and a powerful enough CPU. Java applets are on some smart cards and cell phones, so the CPU required is not as large or as powerful as you may have thought. It is when those applets (i.e., Java programs) execute outside the sandbox (i.e., address space limitations) within your browser, or in temp folders on the hard drive, or in allocated memory space, that the trouble usually begins. A note of advice: Be aware of the environment!

We have discussed memory and the various kinds of memory, whether it is physical or symbolic. Now we will consider the importance of machine types.

Machine Types

We have briefly discussed one machine type — the virtual machine, which is the case when a program is being executed in memory (for example, Java Virtual Machine [VM], anti-virus heuristics technology). Symantec's white papers explain the basic principle behind heuristic technology. In a nutshell, Symantec's program, in addition to emulating the program in a virtual machine, is also monitoring requests being made to the operating system (OS).[12] The conceptual opposite of a virtual machine is the common three-dimensional, physical PC, which is "real." There are at least three other types of machines that we will discuss here: (1) the multistate, (2) the multitasking, and (3) the multiprogramming machines.

A multistate machine actually processes different classification levels at the same time. Think of it as a system enabling users with different authorized classifications to access information from the same workstation rather than using two workstations. For example, with classified documents a user would turn a switch on a box representing nonclassified information on the display screen. Think of it as maintaining confidential, public, and proprietary information.[13] In contrast, a multitasking machine exists when the OS slices out CPU time to different programs to execute specific tasks, or when each program can control the CPU as long as it needs to. For example, Windows 95, Windows NT, and UNIX workstations switch back and forth to give the appearance of executing tasks at the same time. An example of a multitasking machine is best demonstrated by the Windows 3.1 OS. By the way, this explains why 3.1 "locked" more than NT: it did not incorporate memory protection.[14]

The multiprogramming machine is similar to the multitasking machine. However, rather than switch between tasks, it involves execution of two or more programs by one processor. This should not be confused with the multiprocessor, which refers to the number of CPUs used to execute tasks or programs. With a multiprocessor, more than one CPU is being used; Novell's and Microsoft's various server application products support multiprocessors.

Operating Modes

Following a recent house move, we unpacked and I was happy to find a Netware 4.1 reference book. Do not laugh! The principles are still the same today. UNIX, Windows NT, and Novell Netware all use memory protection.

Consider the following example. Imagine a dartboard. Do you have the image in your mind? The smallest circle is a red area or "bull's eye." This circle is ring "0" (or ground zero for you military folks). There are four rings (0 to 3), and each circle gradually radiates outward, getting larger. Now, think of ring 0 as the area where operating systems such as UNIX, NT, and Novell operate. Netware 4.1 servers use this area as a default, although the system administrator could of course change the default setting. Whereas ring 0 is for the OS kernel and provides the least restriction to the CPU, ring 3 (i.e., the outermost ring for Netware 4.1) provides the most restrictions to the CPU. Ironically, although ring 0 is the smallest ring, it offers the fastest performance. As you move from the center outwards (that is, from ring 0 to ring 3), you take a hit in performance. As for the other rings, ring 1 is for the operating system (not the security portion), ring 2 is for the various drivers, and ring 3 is where the programs are executed.

Personally, I have always preferred Novell's security approach over the other OSs. The reason I developed this preference has to do with a little bit of history. Back then, Netware 4.1 would place things in ring 3 as a test or trial area. The process might run a little slower, but at least it did not crash the server! How is that possible? Because Netware 4.1 is operating in ring 0 memory address space, as noted earlier. For example, if the OS receives a request from a process or program to use the memory space in ring 0, the request is blocked; this process is called memory protection.[15] Data may be accessed on the same ring or from a less privileged ring by a program. Resources may be requested in the opposite manner; at the same ring level or from a higher-privileged ring. Processes operating in the inner ring are called "supervisor" or "privileged" state, and those working on the outer rings are called "user" state.[16]

CPU States

CPUs exist in two types of states. Supervisory state exists when a program can access an entire system (i.e., meaning the OS on the mainframe). It is in the supervisory state where both privileged and nonprivileged instructions can be executed. In contrast, a problem state is where nonprivileged instructions and application instructions are executed. For example, telecommunications, ports, and protocols were discussed in Domain 2. The more well-known ports — 1024 and below — operate in a privileged state.[17] As it happens, Microsoft defines eight process states for NT. However, we have cut down the first and last states to come up with a series that looks a lot like the four more commonly known states. This results in a total of six states and includes those listed in Exhibit 125.2.

To summarize, in this section on resource management we have discussed addressing, as well as swapping, paging, caching, storage types, and memory protection.

EXHIBIT 125.2 Process States

1	Ready	Ready to run on the next available processor
2	Running	Program currently being executed
3	Standby	Assigned a queue and about to run
4	Terminated	Finished executing the program
5	Waiting	Not ready for the processor
6	Transition	Ready, waiting on resources other than the CPU (e.g., input from the user, completing a print job, etc.)[a]

[a] See http://support.microsoft.com/support/ntserver/serviceware/nts40y60.asp

Environment

Now that we have discussed memory, CPUs, buses, logical and physical organizations, the basic technology concepts, and a little sprinkling on software, we will address the environment and software applications. In Domain 1, Access Control Systems and Methodology, control types were discussed. As a reminder, the control categories mentioned were "PAT." This is, of course, the easiest way to remember the following:

- Physical: Refers to locks, guards, alarms, badge systems, lights, etc.
- Administrative: Refers to policies and procedures, security awareness, auditing, etc.
- Technical: Refers to anti-virus, firewalls, intrusion detection systems (IDS), etc.

As stated before, it is important to know your environment. Consider the fact that Internet stock fraud is estimated at $10 billion per year, or $1 million per hour,[18] or as the FBI's Deputy Assistant Director recently stated, "Cyber crime continues to grow at an alarming rate, and security vulnerabilities contribute to the problem."[19] As evidence of this, results of the Seventh Annual 2002 Computer Crime and Security Survey revealed that:

- 94 percent detected security intrusions within the last year.
- 80 percent acknowledge financial loss.
- Financial losses caused by theft of proprietary information cited as the most severe cases again.
- 74 percent indicated their Internet connection as the most frequent point of attack.
- 78 percent detected employee abuse.
- 85 percent detected computer viruses.

As a result of findings like these and others, it should be clear that protection mechanisms are required now more than ever. Keep in mind that no system can be totally secured. Sooner or later an incident will occur. However, it is those actions and responses used to mitigate damage combined with corrective actions to ensure the same incident does not reoccur that distinguishes the superior (i.e., more secure) system from the others.

Layering

Layering is a concept that is important to understand when designing a security architecture. Remember the earlier discussion of memory protection as it was associated with Netware 4.1? That was actually layering in that the kernel is located in the center with programs located on the outer edge; drivers (for secondary storage) are located in between. Layering refers to the organization of separate functions that interact in a hierarchal sequence or order.[20] A good example for layering is the OSI model: there are seven component layers stacked upon each other. Whether you start from the bottom layer and work up, or the reverse order, there is an interaction among those layers.

Abstraction

Abstraction is something system administrators and programmers should be familiar with in their normal duties. Object-oriented programming uses abstraction. Abstraction (as the definition implies) involves the removal of characteristics from an entity in order to easily represent its essential properties. For example, it is easier for a system administrator to grant group rights to a group of 25 people called "Human Resources" than

to grant 25 individual rights to each HR member. Windows 2000 Professional provides six built-in local groups straight from the "jewel box," including:

- Administrators
- Backup operators
- Power users
- Users
- Guest
- Replicator

Each local group has a set of predefined rights for the user group. If you are "security smart," you have disabled the guest account and renamed the administrator group!

Data Hiding

This also has to do with object-oriented programming. Graphical user interfaces (GUIs) use object-oriented programming. For example, I am using the 2000 Professional OS. The printer icon, which is an object, contains information related to a specific printer. The information on this specific object is predefined. The object only needs to know certain information to complete its task. Think of the items recently learned in this section. Which IRQ, port, and protocol should be used to execute this task? What is the memory space address? Does the user have sufficient rights to print? In other words, anything not specifically needed to carry out the print task is hidden from the printer object.

Principle of Least Privilege

This brings us to the principle of least privilege that applies to programs as well as people. Programs and people should only be given access to those resources necessary to complete a specific task, execute a program, or accomplish their job. Once a process has been accomplished, depending on the circumstances, access to privileged resources should be removed. For example, your organization's work hours are from 7 A.M. to 6 P.M. You have decided to restrict outgoing fax calls between 6:30 P.M. and 7 A.M., preventing the cleaning crew or security guard from abusing the system. Data hiding, abstraction, and hardware segmentation each fall under the principle of least privilege. This principle is critical to understand to properly secure Novell Directory Services (NDS), Microsoft's Active Directory, Lightweight Directory Access Protocol (LDAP), and file, fax, server, and printer access within an organization. Failure to implement the correct assignation of administrative properties to objects, users, and resources, or to properly understand how inheriting rights are transferred, will lead to a security incident each and every time.

Now, as Emeril Lagasse says, "Let's take it up a few notches…." Bam! Remember the PAT acronym? We will begin focusing on a few additional concepts to tie it all together.

Security Practices

Remember that no system is totally secured unless you unplug it. Consequently, a secure system needs preventive, detective, and corrective controls in place in order to take proper action when incidents occur.

Preventive Controls

Preventive controls are measures carried out to block anticipated aggression from hostile forces. Locks, fences, alarms, guards, lighting, access control lists (ACL), IDS, anti-virus software, firewalls, logical access controls (smart cards, biometrics, PINs), demilitarized zones, and policies and procedures are all used to do the job so that those "hostile forces" are less likely to impact the operations. Just how can policy and procedures help? Consider that when an employee is terminated, resigns, or transfers positions, the user's profile must be removed from the network. This means that the third or at least the fourth person who should be notified within the organization is the senior IT security professional, who should remove that person's log-in account.

Exhibit 125.3 shows a list of preventive control tips, though not inclusive of all possible ones, which should provide you with an understanding of what is being discussed.

EXHIBIT 125.3 Preventive Control Tips

- Audit active employee names against user accounts or profiles currently assigned network file access privileges
  - — Remove/disable those accounts where there are discrepancies
- Use incremental and full backups and test backups
- Prepare contingency plans and test regularly
- System administrators should not access personal e-mail while logged into networks with system administration privileges (create a regular user profile to perform this task)
- Harden an operating system prior to placing it online
- Use standard integrated desktops for users
- Use log-in restrictions when it is feasible
- Develop educational and awareness programs for users and system administrators
- Clearly mark and label files (both soft and hard copies, and secondary storage devices)
  - — Sanitize electronic media (reminiscence security) and properly dispose of classified documents whether you are in the private or government sector[a]
- Apply critical patches (software bug fixes) to affected systems (automated tools are available)
- Use a test LAN (certification and accreditation process)[b]
- Use external connectivity controls
- Practice configuration management control
  - — Configure firewalls to allow only those services required for users to accomplish their tasks; restrict all other services or protocols
  - — Change default user passwords, disable guest, and rename administrator group accounts
  - — Set servers to retrieve anti-virus updates at least weekly[c]
    - Use a mobile code software tool to complement anti-virus software (layering technique)
  - — Use a trusted computing base (TCB) model (sorry, Millennium 9x or earlier does not qualify)
  - — Discourage placing Web server software running on top of e-mail server (double ouch!)

[a] This entails proper disposal of classified documents, whether private or federal. Keep in mind it also means proper sanitization of electronic media/equipment before turning it over to schools, charities, etc.

[b] We strongly encourage you to develop a test LAN that is representative of your local network (enclave) environment. Why? Would you want to install something on your main system network and then have to wait for the software interactions and trouble in having it impact the operational network? Instead, would you rather prefer the alternative of having problems on the test LAN segment and being able to work through the problems without impacting the operational network? A certification and accreditation process will minimize the potential for these sorts of problems. If your resources are scarce, do not throw away those old computers, routers, etc. Instead, place them in your test LAN.

[c] Anti-virus alone is not good enough to protect servers/clients, nor does it stop all malicious code. Anti-virus should be used in conjunction with mobile code software, because anti-virus is only as good as the installed definitions.

Detective Controls

Sooner or later someone will try to breach your security. As network professionals, we need to be vigilant about employing effective methods to catch cyber crooks. Below is a listing of a few detective controls:

- Enable logging for system changes, unsuccessful log-ins, system policy changes, access to files.
- Review those logs, outsource logging tasks, or automate the process (few good automated tools available).
- Conduct incident investigations.
- Use an IDS.
- Use anti-virus software. (*Note:* Can also be called preventive.)
- Make sure to have supervision oversight, job rotations, mandatory vacations.[21]

Corrective Controls

Zero-day incidents are here to stay and will probably increase in the near future. Zero-day incidents are attacks that are exploited in the wild before they are reported to the rest of the security community by groups such as the National Infrastructure Protection Center (NIPC), the Computer Emergency Response Team (CERT), or the Common Vulnerability Exposures (CVE) list. Not surprisingly, hackers exploit those exposed vulnerabilities. What can a network security professional do? Our recommendation: Develop work-arounds and apply the patches as they become available. Addressing audit deficiencies (company or government auditors) and incident investigations will allow the update of security policies and updating IDS databases.

Trusted Computing Base (TCB)

At this point, it is useful to restate that security involves a very fine synergy that represents the interaction between software, technology, and the environment. It should be clear to you that security is not only of the utmost importance, but it is multifaceted. Consider the following point from the National Information Systems Security Glossary on TCB:

> The totality of protection mechanisms within a computer system, including hardware, firmware, and software, the combination of which is responsible for enforcing a security policy. The ability of a trusted computing base to enforce correctly a unified security policy depends on the correctness of the mechanisms within the trusted computing base, the protection of those mechanisms to ensure their correctness, and the correct input of parameters related to the security policy.[22]

The tips we have suggested for preventive and detective controls fall under trusted computing base (TCB). Think of the TCB as a baseline model to obtain a level of trust. Newton's third law of motion states, "For every reaction there is an equal and opposite reaction."[23] Although Newton was discussing physics, the same case can be made for the various configuration and security policy settings a person can make to the hardware, software, and firmware of a system. We now turn to yet another aspect of security related issues.

Social Engineering

> People are the weakest link. You can have the best technology, firewalls, intrusion detection systems, biometric devices — and somebody can call an unsuspecting employee. That's all she wrote, baby. They got everything.
>
> **— Kevin Mitnick[24]**

Today, almost everyone who has at least a passing familiarity with the Internet and Internet-linked systems knows that the integrity of any good system lies in its ability to protect itself from intruders or would-be attackers. As a way of responding to the threat that computer hackers pose, organizations with public and private networks have implemented a variety of strategies ranging from static authentication — whereby a would-be intruder can gain access only by guessing at a legitimate user's authentication data, to more sophisticated intrusion detection systems that effectively discover unauthorized activity and in some cases identify intruders.[25] Although each of the different strategies varies in complexity and component parts, they are similar in that they each represent a deliberate attempt to discourage or at least minimize the threat of potential intruders.

Yet, there is an additional threat to which even the most secure systems are vulnerable. This additional threat, as illustrated by the above quote, has a decidedly human aspect to it, and occurs when would-be attackers try to access a system by manipulating and deceiving company employees or other legitimate system users. In its most egregious form, "social engineering" practices permit intruders to gain unrestricted access to closed systems by talking and interacting with company employees. In a slightly more benign form, it involves intruders gaining unauthorized information about employees or company business practices. In short, hackers[26] and would-be intruders use their "social skills" (e.g., persuasion, coercion, deception) to feign legitimacy in order to obtain compliance from unsuspecting employees. When this occurs, employees find themselves on the receiving end of an earnest request for information from what is ostensibly a legitimate user or company employee. Oftentimes the intruder poses as a senior executive of the company whose power and prestige make compliance with the intruder's request (however unusual) especially likely.

Consider the scenario in which a would-be intruder poses as a company executive and asks a help-desk employee to provide an access code he or she claims to have accidentally left in the office. Alternatively, imagine the hacker who telephones the CEO's executive secretary with an elaborate ruse that concludes with a request for the CEO's password. Although neither employee can be certain of the legitimacy of the request, they will feel a personal sense of obligation to comply with the actual request. Here, compliance occurs when the employee does what he or she is asked to do (by providing the unknown person with privileged information) even though he or she might prefer not to do so. In cases like these, successful computer hackers and intruders are able to influence company employees in a way that brings about compliance.

Social engineering represents a form of persuasive manipulation. Use of social engineering techniques involves the exploitation of common and basic human attributes — namely, that of helpfulness and trustworthiness. Although the term "social engineering" is specific to the computing industry, the techniques involved

are common to a host of situations and industries. Across all settings in which people are dependent on the compliance of others, social engineering techniques are at work. For example, the parent who wishes to influence a child to brush its teeth, the husband who seeks to convince his wife of the necessity of an expensive purchase, and the panhandler who requests money from passers-by each use social engineering skills to bring about compliance.

There are a number of well-known cases in the computer industry in which intruders have succeeded at social engineering. To be sure, the actual mediums through which these manipulative techniques are transmitted are varied, and include the telephone, e-mail, trash pilferage, in-person site visits, and, of course, snail mail. Regardless of the medium employed, would-be intruders intent on accessing a system hone their social skills to gain information, manipulate policies, and acquire resources, all with the unwitting assistance and compliance of company employees.

Students of human behavior know well the tendency for people to be compliant with requests emanating from people who they believe to be legitimate authority figures. Researchers in social psychology,[27] for example, have conducted numerous empirical studies investigating the conditions under which compliance is most likely, as well as those circumstances or factors that may limit its occurrence. Research findings pertaining to the latter would seem to be most relevant for computer professionals who are committed to ensuring the security of their network systems.

What does the research tell us about the effectiveness of efforts to resist social pressure? In other words, how can network administrators inoculate employees against the social engineering efforts of would-be intruders? Can anything be done to combat the would-be intruder and keep systems users and data safe and secure? The answer is "yes."

Fortunately, research on best business practices has revealed that there are important limitations to social engineering techniques. Knowledge of the limitations to social engineering schemes can significantly enhance a network administrator's ability to ensure the integrity of a system. Although network administrators cannot eliminate the problem of computer hackers, they can take specific steps to reduce the effectiveness of their influence schemes. What follows is a brief discussion of at least three prescriptions for network systems administrators who are vulnerable to social engineering schemes.

Risk Awareness Training

First and foremost, employees must be made aware of the potential problems posed by computer hackers. Only when employees and other system users know about the existence and pervasiveness of social engineering schemes can they act against them. According to some writers in this area,[28] when it comes to user suspicion, "paranoia is good!" Whether this occurs in the context of new employee orientation training or specific security awareness training for users, individuals must be informed about the potential risk for social engineering schemes. Far too often, individuals assume that their systems are invincible, and that requests for information come from legitimate users.

System users and employees must realize the role that they personally play in the security of a company's information. Any information awareness session should be focused on getting people to appreciate the fact that there are people out there who are trying to access companies' networks, and that their role as employees or legitimate users of the system is to be both proactive and reactive in making it as hard as possible for would-be intruders to succeed. Proactively, this means that users must be cautious about whose requests they comply with, and reactively, when users encounter unusual or outrageous requests for information either in-person or online they should immediately alert their network administrator.

The theory of psychological reactance may be particularly useful for those most apt to encounter hackers employing social engineering schemes. This theory is most relevant in situations where employees are sensitized to the potential risk of social engineering schemes, and would-be intruders employ high-pressure tactics. According to the theory, too much pressure to comply with a request can actually have the opposite intended effect.[29] The idea that forms the basis of the theory is that people are motivated to maintain their sense of personal freedom and when they suspect that they are being pressured or feel that their freedom is being threatened, they will act so as to protect their freedom by refusing to comply. Hence, they react against the pressure to comply by doing the exact opposite of what they are being asked to do. Employees who are aware of the risk of social engineering schemes and who confront would-be intruders using high-pressure tactics are especially likely to experience the phenomenon of reactance. Consequently, in these situations, by being less willing to comply, they are more apt to thwart the would-be intruder's efforts. Network administrators should

work to ensure that the risk of social engineering schemes remains salient for employees and other legitimate users with access to company information.

Formulate a Written Policy for Procedures

Sometimes employees who are approached with a request for information may suspect that something is amiss, but because they are unsure about what to do about their suspicions, they wind up complying with the would-be intruder's request. Even the most conscientious employees who are usually vigilant about information distribution may encounter a situation in which they are faced with a novel request. They may simply be at a loss to know the appropriate procedure. Although written policy cannot possibly speak to every potential request that a world-be intruder can come up with, existing policy should inform employees that "when in doubt, be conservative, do not comply."

The policy that is ultimately formulated with the help of users should be comprehensive and clear. Employees and others who are approached for information should clearly be able to distinguish a legitimate request for information from an illegitimate one, whether the person requesting the information is a legitimate user or not. What is more, employees must be able to feel that they can ask questions about the request in an environment in which they do not appear silly or ignorant. In some cases, network operators have initially failed to take users' requests for information seriously enough and in the end are burned. One way of ensuring that this does not happen is to create a policy that permits, indeed encourages, employees who are uncertain about information requests to verify the legitimacy of the request with a network operator. This may take more time up front, but in the long run it may prove to be extremely beneficial.

Eliminating Paper Trails and Staying Connected

The final set of prescriptions aimed at securing network systems from would-be intruders employing social engineering schemes has to do with the elimination of company documents and materials, and maintaining contact with organizations specializing in security. With respect to the former, although there has been considerable discussion about trash pilferage in the popular press, with the exception of a few highly secure federal agencies, people in general are lax about discarding their trash. Organizations must provide employees with convenient ways of discarding sensitive documents and material. Finally, companies should keep in touch with those organizations and agencies that can be trusted to provide up-to-date, dependable information about security issues.

Certification and Accreditation (C&A)

Remember the admonition to know your environment? Understanding which laws, policies, and service-level agreement contracts are in place is critical to effective implementation and testing of a security policy or architecture. Although there is no law that formally requires companies to perform a C&A, shareholders enforce policies within the private sector. Conversely, within the federal arena, Congress ultimately regulates such practices. But the question remains as to why C&A is important, and perhaps more importantly, what are its implications and impact for the network administrator?

The National Institute of Standards and Technology (NIST) defines certification as

> the comprehensive evaluation of the technical and nontechnical security controls of an IT system to support the accreditation process that establishes the extent to which a particular design and implementation meets a set of specified security requirements.[30]

What are the implications of this for private industry and the federal government? What motivates each to comply? For the private sector it may be argued that the primary motive is an economic one; in other words, "the wallet." When companies fail to do so, the consequences can be grave. Consider the following statement made by one attorney:

> We have seen several recent incidents where our clients have threatened legal action against trading partners who have been the cause of a security breach or virus infection. All of these cases have been settled out of court, primarily because of the unwanted publicity connected with court cases.[31]

Thus, having a C&A package that has demonstrated effective implementation is one manner of showing the courts due diligence.

As for the federal government, it is mandated by laws such as the Health Insurance Portability and Accountability Act of 1996 (HIPAA), Clinger–Cohen, and the Federal Information Security Management Act of 2002 (FISMA), to name a few. Although it depends on which part of the federal government you are referring to, the major policies are DoDD 8500.1, DoDD 5200.40, and NIST's Special Publication 800-37.

Accreditation

Accreditation refers to

> the authorization of an IT system to process, store, or transmit information, granted by a management official. Accreditation, which is required under OMB Circular A-130, is based on an assessment of the management, operational, and technical controls associated with an IT system.[32]

Simply stated, this amounts to management's formal approval of the certification process that essentially says that it can live with the risks to the IT system and the mitigation of those risks. It also means that there is help to assist someone through the process. Admittedly, this is a complicated process, but there are automated tools available from appropriate vendors. Just ensure that the information entered is valid and not "pencil-whipped."

Security Models

Taking the principle of trust further, we will discuss the more commonly used security models, including:

- Bell–LaPadula
- Biba
- Clark–Wilson

The Bell-LaPadula Model

In 1973, Drs. Bell and LaPadula from the MITRE Corporation developed a security model for the Department of Defense. (As you may recall, mainframes were common during this period.) The Bell–LaPadula model controls information flow. For example, Novell's and Microsoft's training literature discuss access rights to objects and resources. Those various access privileges (read, write, delete, modify, etc.) form a "woven lattice." One concept of the model states that a user cannot read an object of a higher classification than granted.

For example, if you have a government security clearance of Secret, you are allowed to read Secret and below classification level documents; accordingly, you have no access to Top Secret information. The Bell–LaPadula model incorporates the "* property," i.e., it states that a user cannot write from a higher classification level to a lower one. Using the previous example, Secret e-mail messages or documents cannot be sent to recipients who do not have a Secret or higher clearance or written or stored to file servers designated for Unclassified information. It is for this reason that the Bell–LaPadula model is considered a confidentiality model.

Biba Model

This model, developed in the late 1970s, uses a process similar to the Bell–LaPadula model (i.e., the subject cannot write to a higher integrity source). However, the Biba model is an integrity model. Whereas the Bell–LaPadula model was concerned with protecting the release of information to unauthorized users, the Biba model was developed strictly for the developing computer systems of that period. With this model, unauthorized objects are blocked from making modifications. The "* property" is used to block subjects from writing to objects of higher integrity, the "read property" keeps subjects from corruption by objects of lower integrity, and subjects cannot request services from objects maintaining a higher integrity model.[33]

For example, imagine that you work for the CIA as a low-level analyst with a Secret clearance. You do the leg work for a report to gather raw intelligence from various sources addressing sonar technology. You take your proposal to your supervisor for review; your supervisor makes further input to the report and hands it off to ex-Naval officers and an expert who has published extensively on bats' acute hearing techniques. They further refine the report to a finished product that lands on the Secretary of the Navy's desk. Although the Secretary would never read your report in its raw state, he would read the finished CIA product. Although you may wish to update the report in its finished form, you are actually blocked from write access to it because you are now at a lower level of integrity than the report. However, you would be allowed to read the report in accordance with Biba. The same principle could be used for a database recognized as the authoritative source such as that produced by the Bureau of the Census.

Clark–Wilson Model

This is another model developed to address integrity and uses a broader approach than the Biba model, which addressed only subjects and objects. Clark–Wilson addresses a special type of program called a "well-formed transaction." In this case, changes to a process or to data can be made only through this trusted program because the subject can access the object through the trusted program only. This concept binds the subject to the program and the program to the object, creating a "triple" instead of the subject–object "tuple" used in Biba. The trusted program is constructed to only make authorized changes. Think of it as incorporating a program to complete transactions, and one that incorporates the policy of separation of duties as well. Separation of duty involves breaking a task or operations into parts where no one person can complete a process. For example, this would prevent someone in Acquisition from cutting a check to purchase office furniture for use in a private home. Access control prevents unauthorized personnel from making alterations or changes to data. Separation of duty helps prevent authorized personnel from making unauthorized modifications.

Software Applications

At this point, it is worth asking, "How does a person know how to distinguish between the good, the bad, and the ugly software in order to develop a valid C&A package? What is the TCP based on?" These are good questions that you may already have considered asking. The answers have to do with the efforts of the federal government, which has provided a number of valuable resources to IT professionals through the National Information Assurance Partnership (NIAP). NIAP is an initiative to increase information technology security by collaborating with industry in security testing, research, and the development of information assurance methodologies.[34] From NIAP came the Common Criteria Evaluation and Validations Scheme (CCEVS), which is jointly managed by the National Security Agency and NIST. The CCEVS established a national program for the evaluation of information technology products. This program is known as Common Criteria (CC) and it is identified as International Standards Organization (ISO) 15408. Under CC there are seven protection profiles. A firm understanding of the CC's protection profiles, which also include seven evaluation assurance levels (EAL), is important for various reasons. If you are working in the federal government sector, following CC is a requirement mandated by policy. For evidence of this, see the reference cited in Note 2.

Minimizing the Need for Applying Patches

Why make life difficult? By using an enterprisewide architecture, which employs a layered approach, you will minimize the need to apply patches. Keep this in mind as your organization begins to perceive a shortage of resources and starts looking to cut resources from somewhere. Rather than being on the short end of the stick, be progressive: pitch the use of an enterprise architecture. Incidentally, this has been the direction in which the federal government is moving.

For example, using the Command, Control, Communications, Computers, Intelligence, Surveillance, Reconnaissance (C4ISR) architecture model, which applies to the federal government and to some extent carries over to the private sector, you can minimize a lot of the work down the road through detailed planning (for patches). The enterprise architecture can list software by version (i.e., an interim technical reference) for approving software prior to placing it on the desktop or network. The approval can include supporting the software, training, life cycle, etc. For example, imagine instituting a standardized integrated desktop configuration that includes the minimum standard for clients and network connectivity. For the desktop (as an example), you would only support Windows NT 4.0 Service Pack 6a. This minimizes the need to support the previous service packs as well as the time required for installing patches. The architecture would detail setting retirement dates (for applications and operating system) and planning for new technology insertion dates, thus minimizing the "software zoo."[35] This in turn reduces legacy applications, vendors' support (for phased-out software, much like NT 4.0 now), the associated security vulnerabilities, and time mitigating those vulnerabilities (applying patches, policy, etc.).

Not surprisingly, there are at least two cost-saving benefits associated with this effort. First, minimized desktop support is achieved by narrowing various operating system platforms to a few (even within the Microsoft family there are various versions of Office for the same OS). Second, by narrowing the software applications supported, you reduce manpower needs and patches to be maintained or applied. In this way, organizations can arrange individuals into groups (power user, standard user, sys-admin/support, as an example

for software applications) and apply the manpower to group configurations rather than the individual desktop zoo.

We conclude by listing the benefits of using an enterprise architecture, which are numerous:

- Capturing facts on operations and functions in an understandable manner to drive better planning and decision-making
- Supporting analyses of alternatives, risks, and trade-offs for the investment-management process, which reduces the risk of:
  - — Building systems that do not meet operational needs
  - — Expending resources on developing unnecessary duplicative functionality
- Improving consistency, accuracy, and timeliness of information shared collaboratively across the enterprise[36]

Notes

1. Merriam-Webster Dictionary, http://www.m-w.com.
2. National Security Telecommunications and Information Systems Security Policy (NSTISSP) 11, January 2000, available at NIST.gov.
3. Available at http://www.webopedia.com.
4. In contrast, secondary storage refers to the floppy disk, tape drives, hard disk, and optical media we are so familiar with handling. You know the terms: terabytes, gigabytes, megabytes, or kilobytes.
5. From http://www.theboys.com.
6. A process is a program being executed, and is discussed in more detail in the section on machine types.
7. Denis Howe, The Free Online Dictionary of Computing, 1993–2001.
8. According to information available at Apple.com, the PowerPC G4 is a collaborative effort between Apple, Motorola, and IBM.
9. Harris, S. (2002). *All in One CISSP Certification,* Berkeley, California: Osborne/McGraw-Hill.
10. See www.webopedia.com.
11. For a more-detailed discussion of this process, see McClure, S., Scambray, J., and Kurtz, G. (2001). *Hacking Exposed,* 3rd ed., Berkeley, California: Osborne/McGraw-Hill.
12. For more information, see Understanding Heuristics: Symantec's Bloodhound Technology.
13. Further discussion of multistate machines is found in the section on Security Models.
14. This is explained further in the section on Operating Modes.
15. Lawrence, B. (1996). *Using Netware 4.1,* 2nd ed. Indianapolis: Que
16. Harris, S. (2002). *All in One CISSP Certification,* Berkeley, California: Osborne/McGraw-Hill.
17. RFC 793, Internet Assigned Numbers Authority (IANA).
18. According to Louis J. Freeh, Director of the FBI, March 28, 2000, Congressional Statement on Cyber Crime.
19. Farnan, J.E., Deputy Assistant Director, 4/3/03, Congressional Statement on Fraud: Improving Information Security.
20. See http://www.whatis.com.
21. Consider the fact that most large banks require forced vacations. It is harder, for example, to keep an embezzlement scam running while a person is away on vacation and another employee is filling his/her position during the absence.
22. National Information Systems Security (InfoSec) Glossary, NSTISSI No. 4009, June 5, 1992.
23. Sir Isaac Newton, Laws of Motion, 1686.
24. Kevin Mitnick, the notorious computer hacker, was arrested for computer crimes in 1995, and is one of the first people to be convicted and jailed for unauthorized access of someone else's computer.
25. Tipton and Krause, *Information Security Management Handbook, 4th Ed.*, 2000. Boca Raton, Florida: Auerbach Publications.
26. For ease of discussion, the term "hacker" is used throughout this section. However, it is acknowledged that this discussion also applies to the efforts of crackers, coders, and cyber punks.
27. There are many different studies in this area investigating the effectiveness of a host of compliance techniques. They have included studies of car salespeople, professional fundraisers, and con artists.
28. See p. 587 in McClure et al. (2001).

29. For a classic discussion of psychosocial work investigating this, see Brehm, J.W. (1996). *A Theory of Psychological Reactance,* New York: Academic Press.
30. NIST Special Publication 800-37, Guidelines for the Security Certification and Accreditation of Federal Information Technology Systems, October 2002.
31. Kitt Burden, http://computerweekly.com, February 14, 2002.
32. From NIST SP 800-37.
33. We have borrowed this from http://www.cccure.org/Documents/HISM/023-026.html.
34. See NIAP brochure for 2003.
35. For example, for your architecture, you would support either Microsoft Office, Corel's Office Suite, or Sun's office package. To do so, you would probably have to migrate the majority to one or the other if you were not currently supporting it.
36. Air Force (C4ISR) architecture plan, November 2002.

Security Models for Object-Oriented Data Bases

James Cannady

Payoff

Object-oriented (OO) techniques are a significant development in the management of distributed data. Data base design is influenced to an ever-greater degree by OO principles. As more DBMS products incorporate aspects of the object-oriented paradigm, data base administrators must tackle the unique security considerations of these systems and understand the emerging security model.

Introduction

Object-oriented (OO) programming languages and OO analysis and design techniques influence data base systems design and development. The inevitable result is the object-oriented data base management system (OODBMS).

Many of the established data base vendors are incorporating object-oriented concepts into their products in an effort to facilitate data base design and development in the increasingly object-oriented world of distributed processing. In addition to improving the process of data base design and administration, the incorporation of object-oriented principles offers new tools for securing the information stored in the data base. This article explains the basics of data base security, the differences between securing relational and object-oriented systems, and some specific issues related to the security of next-generation OODBMSs.

Basics of Data Base Security

Data base security is primarily concerned with the secrecy of data. Secrecy means protecting a data base from unauthorized access by users and software applications.

Secrecy, in the context of data base security, includes a variety of threats incurred through unauthorized access. These threats range from the intentional theft or destruction of data to the acquisition of information through more subtle measures, such as inference. There are three generally accepted categories of secrecy-related problems in data base systems:

- **The improper release of information from reading data that was intentionally or accidentally accessed by unauthorized users.** Securing data bases from unauthorized access is more difficult than controlling access to files managed by operating systems. This problem arises from the finer granularity that is used by data bases when handling files, attributes, and values. This type of problem also includes the violations to secrecy that result from the problem of inference, which is the deduction of unauthorized information from the observation of authorized information. Inference is one of the most difficult factors to control in any attempts to secure data. Because the information in a data base is semantically related, it is possible to determine the value of an attribute without accessing it directly. Inference problems are most serious in statistical data

bases, where users can trace back information on individual entities from the statistical aggregated data.

- **The improper modification of data.** This threat includes violations of the security of data through mishandling and modifications by unauthorized users. These violations can result from errors, viruses, sabotage, or failures in the data that arise from access by unauthorized users.

- **Denial-of-service threats.** Actions that could prevent users from using system resources or accessing data are among the most serious. SYN flood attacks against network service providers are an example of denial-of-service threats in which a barrage of messages is sent to the server at a rate faster than the system can deal with them. Such attacks prevent authorized users from using system resources.

Discretionary Versus Mandatory Access Control Policies

Both traditional relational data base management system (RDBMS) security models and object-oriented data base models make use of two general types of access control policies to protect the information in multilevel systems. The first of these policies is the discretionary policy. In the discretionary access control (DAC) policy, access is restricted based on the authorizations granted to the user.

The mandatory access control (MAC) policy secures information by assigning sensitivity levels, or labels, to data entities or objects. MAC policies are generally more secure than DAC policies and they are used in systems in which security is critical, such as military applications. However, the price that is usually paid for this tightened security is reduced performance of the data base management system. Most MAC policies also incorporate DAC measures as well.

Securing an RDBMS Versus OODBMS: Know the Differences

The development of secure models for object-oriented DBMSs has obviously followed on the heels of the development of the data bases themselves. The theories that are currently being researched and implemented in the security of object-oriented data bases are also influenced heavily by the work that has been conducted on secure relational data base management systems.

Relational DBMS Security

In traditional RDBMSs, security is achieved principally through the appropriate use and manipulation of view and the SQL GRANT and REVOKE statements. These measures are reasonably effective because of their mathematical foundation in relational algebra and relational calculus.

View-based Access Control.

Views allow the data base to be conceptually divided into pieces in ways that allow sensitive data to be hidden from unauthorized users. In the relational model, views provide a powerful mechanism for specifying data-dependent authorizations for data retrieval.

Although the individual user who creates a view is the owner and is entitled to drop the view, he or she may not be authorized to execute all privileges on it. The authorizations that

the owner may exercise depend on the view semantics and on the authorizations that the owner is allowed to implement on the tables directly accessed by the view. For the owner to exercise a specific authorization on a view that he or she creates, the owner must possess the same authorization on all tables that the view uses. The privileges the owner possesses on the view are determined at the time of view definition. Each privilege the owner possesses on the tables is defined for the view. If, later on, the owner receives additional privileges on the tables used by the view, these additional privileges will not be passed onto the view. In order to use the new privileges within a view, the owner will need to create a new view.

The biggest problem with view-based mandatory access controls is that it is impractical to verify that the software performs the view interpretation and processing. If the correct authorizations are to be assured, the system must contain some type of mechanism to verify the classification of the sensitivity of the information in the data base. The classification must be done automatically, and the software that handles the classification must be trusted. However, any trusted software for the automatic classification process would be extremely complex. Furthermore, attempting to use a query language such as structured query language (SQL) to specify classifications quickly becomes convoluted and complex. Even when the complexity of the classification scheme is overcome, the view can do nothing more than limit what the user sees—it cannot restrict the operations that may be performed on the views.

GRANT and REVOKE Privileges.

Although view mechanisms are often regarded as security “freebies” because they are included within SQL and most other traditional relational data base managers, views are not the sole mechanism for relational data base security. GRANT and REVOKE statements allow users to selectively and dynamically grant privileges to other users and subsequently revoke them if necessary. These two statements are considered to be the principal user interfaces in the authorization subsystem.

There is, however, a security-related problem inherent in the use of the GRANT statement. If a user is granted rights without the GRANT option, he or she should not be able to pass GRANT authority on to other users. However, the system can be subverted by a user by simply making a complete copy of the relation. Because the user creating copy is now the owner, he or she can provide GRANT authority to other users. As a result, unauthorized users are able to access the same information that had been contained in the original relation. Although this copy is not updated with the original relation, the user making the copy could continue making similar copies of the relation, and continue to provide the same data to other users.

The REVOKE statement functions similarly to the GRANT statement, with the opposite result. One of the characteristics of the use of the REVOKE statement is that it has a cascading effect. When the rights previously granted to a user are subsequently revoked, all similar rights are revoked for all users who may have been provided access by the originator.

Other Relational Security Mechanisms.

Although views and GRANT/REVOKE statements are the most frequently used security measures in traditional RDBMSs, they are not the only mechanisms included in most security systems using the relational model. Another security method used with

traditional relational data base managers, which is similar to GRANT/REVOKE statements, is the use of query modification.

This method involves modifying a user's query before the information is retrieved, based on the authorities granted to the user. Although query modification is not incorporated within SQL, the concept is supported by the Codd-Date relational data base model.

Most relational data base management systems also rely on the security measures present in the operating system of the host computer. Traditional RDMBSs such as DB2 work closely with the operating system to ensure that the data base security system is not circumvented by permitting access to data through the operating system. However, many operating systems provide insufficient security. In addition, because of the portability of many newer data base packages, the security of the operating system should not be assumed to be adequate for the protection of the wealth of information in a data base.

Object-Oriented DBMS Characteristics

Unlike traditional RDBMSs, secure object-oriented DBMSs (or OODBMSs) have certain characteristics that make them unique. Furthermore, only a limited number of security models have been designed specifically for object-oriented data bases. The proposed security models make use of the object-oriented concepts of:

- Encapsulation.
- Inheritance.
- Information hiding.
- Methods (in the OO paradigm, an object contains data and methods; methods are the components that act on the data and provide user access to the data. It helps to think of methods as functions from the structured programming environment.).
- The ability to model real-world entities.

The object-oriented data base model also permits the classification of an object's sensitivity through the use of class (of entities) and instance. When an instance of a class is created, the object can automatically inherit the level of sensitivity of the superclass. Although the ability to pass classifications through inheritance is possible in object-oriented data bases, class instances are usually classified at a higher level within the object's class hierarchy. This prevents a flow control problem, where information passes from higher to lower classification levels.

Object-oriented DBMSs also use unique characteristics that allow these models to control the access to the data in the data base. They incorporate features such as flexible data structure, inheritance, and late binding. Access control models for OODBMSs must be consistent with such features. Users can define methods, some of which are open for other users as public methods. Moreover, the OODBMS may encapsulate a series of basic access commands into a method and make it public for users, while keeping basic commands themselves away from users.

Proposed OODBMS Security Models

Currently only a few models use discretionary access control measures in secure object-oriented data base management systems.

Explicit Authorizations.

The ORION authorization model is probably the best OODBMS discretional access control security model available today. ORION permits access to data on the basis of explicit authorizations provided to each group of users. These authorizations are classified as positive authorizations because they specifically allow a user access to an object. Similarly, a negative authorization is used to specifically deny a user access to an object.

The placement of an individual into one or more groups is based on the role that the individual plays in the organization. In addition to the positive authorizations that are provided to users within each group, there are a variety of implicit authorizations that may be granted based on the relationships between subjects and access modes.

Data-Hiding Model.

A similar discretionary access control secure model is the data-hiding model proposed by Dr. Elisa Bertino of the Universita' di Genova. This model distinguishes between public methods and private methods.

The data-hiding model is based on authorizations for users to execute methods on objects. The authorizations specify which methods the user is authorized to invoke. Authorizations can only be granted to users on public methods. However, the fact that a user can access a method does not automatically mean that the user can execute all actions associated with the method. As a result, several access controls may need to be performed during the execution, and all of the authorizations for the different accesses must exist if the user is to complete the processing.

Similar to the use of GRANT statements in traditional relational data base management systems, the creator of an object is able to grant authorizations to the object to different users. The “creator” is also able to revoke the authorizations from users in a manner similar to REVOKE statements. However, unlike traditional RDBMS GRANT statements, the data-hiding model includes the notion of protection mode. When authorizations are provided to users in the protection mode, the authorizations actually checked by the system are those of the creator and not the individual executing the method. As a result, the creator is able to grant a user access to a method without granting the user the authorizations for the methods called by the original method. In other words, the creator can provide a user access to specific data without being forced to give the user complete access to all related information in the object.

Other DAC Models for OODBMS Security.

Rafiul Ahad has proposed a similar model that is based on the control of function evaluations. Authorizations are provided to groups or individual users to execute specific methods. The focus in Ahad's model is to protect the system by restricting access to the methods in the data base, not the objects. The model uses proxy functions, specific functions, and guard functions to restrict the execution of certain methods by users and enforce content-dependent authorizations.

Another secure model that uses authorizations to execute methods has been presented by Joel Richardson. This model has some similarity to the data-hiding model's use of GRANT/REVOKE-type statements. The creator of an object can specify which users may execute the methods within the object.

A final authorization-dependent model emerging from OODBMS security research has been proposed by Dr. Eduardo B. Fernandez of Florida Atlantic University. In this model the authorizations are divided into positive and negative authorizations. The Fernandez model also permits the creation of new authorizations from those originally specified by the user through the use of the semantic relationships in the data.

Dr. Naftaly H. Minsky of Rutgers University has developed a model that limits unrestricted access to objects through the use of a view mechanism similar to that used in traditional relational systems data base management systems. Minsky's concept is to provide multiple interfaces to the objects within the data base. The model includes a list of laws, or rules, that govern the access constraints to the objects. The laws within the data base specify which actions must be taken by the system when a message is sent from one object to another. The system may allow the message to continue unaltered, block the sending of the message, send the message to another object, or send a different message to the intended object.

Although the discretionary access control models do provide varying levels of security for the information within the data base, none of the DAC models effectively addresses the problem of the authorizations provided to users. A higher level of protection within a secure object-oriented data base model is provided through the use of mandatory access control.

MAC Methods for OODBMS Security.

Dr. Bhavani Thuraisingham of MITRE Corp. proposed in 1989 a mandatory security policy called SORION. This model extends the ORION model to encompass mandatory access control. The model specifies subjects, objects, and access modes within the system, and it assigns security/sensitivity levels to each entity. Certain properties regulate the assignment of the sensitivity levels to each of the subjects, objects, and access modes. In order to gain access to the instance variables and methods in the objects, certain properties that are based on the various sensitivity levels must be satisfied.

A similar approach has been proposed in the Millen-Lunt model. This model, developed by Jonathan K. Millen of MITRE Corp. and Teresa Lunt of SRI/DARPA(Defense Advanced Research Projects Agency), also uses the assignment of sensitivity levels to the objects, subjects, and access modes within the data base. In the Millen-Lunt model, the properties that regulate the access to the information are specified as axioms within the model. This model further attempts to classify information according to three different cases:

- The data itself is classified.
- The existence of the data is classified.
- The reason for classifying the information is also classified.

These three classification broadly cover the specifics of the items to be secured within the data base; however, the classification method also greatly increases the complexity of the system.

The SODA Model.

Dr. Thomas F. Keefe of Penn State University proposes a model called Secure Object-Oriented Data Base (SODA). The SODA model was one of the first models to address the specific concepts in the object-oriented paradigm. It is often used as a standard example of secure object-oriented models from which other models are compared.

The SODA model complies with MAC properties and is executed in a multilevel security system. SODA assigns classification levels to the data through the use of inheritance. However, multiple inheritance is not supported in the SODA model.

Similar to other secure models, SODA assigns security levels to subjects in the system and sensitivity levels to objects. The security classifications of subjects are checked against the sensitivity level of the information before access is allowed.

Polyinstantiation

Unlike many current secure object-oriented models, SODA allows the use of polyinstantiation as a solution to the multiparty update conflict. This problem arises when users with different security levels attempt to use the same information. The variety of clearances and sensitivities in a secure data base system result in conflicts between the objects that can be accessed and modified by the users.

Through the use of polyinstantiation, information is located in more than one location, usually with different security levels. Obviously the more sensitive information is omitted from the instances with lower security levels.

Although polyinstantiation solves the multiparty update conflict problem, it raises a potentially greater problem in the form of ensuring the integrity of the data within the data base. Without some method of simultaneously updating all occurrences of the data in the data base, the integrity of the information quickly disappears. In essence, the system becomes a collection of several distinct data base systems, each with its own data.

Conclusion

The move to object-oriented DBMSs is likely to continue for the foreseeable future. Because of the increasing need for security in distributed environment, the expanded selection of tools available for securing information in this environment should be used fully to ensure that the data is as secure as possible. In addition, with the continuing dependence on distributed data the security of these systems must be fully integrated into existing and future network security policies and procedures.

The techniques that are ultimately used to secure commercial OODBMS implementations will depend in large part on the approaches promoted by the leading data base vendors. However, the applied research that has been conducted to date is also laying the groundwork for the security components that will in turn be incorporated in the commercial OODBMSs.

Author Biographies

James Cannady

James Cannady is a research scientist at Georgia Tech Research Institute. For the past seven years he has focused on developing and implementing innovative approaches to computer security in extremely sensitive networks and systems in military, law enforcement, and commercial environments.

126

Common System Design Flaws and Security Issues

William Hugh Murray, CISSP

This chapter identifies and describes many of the common errors in application and system design and implementation. It explains the implications of these errors and makes recommendations for avoiding them. It treats unenforced restrictions, complexity, incomplete parameter checking and error handling, gratuitous functionality, escape mechanisms, and unsafe defaults, among others.

In his acceptance of the Turing Award, Ken Thompson reminded us that unless one writes a program oneself, one cannot completely trust it. Most people realize that although writing a program may be useful, even necessary, for trust, it is not sufficient. That is to say, even the most skilled and motivated programmers make errors. On the other hand, if one had to write every program that one uses, computers would not be very useful. It is important to learn both to write and recognize reliable code.

Historically, the computer security community has preferred to rely on controls that are external to the application. The community believed that such controls were more reliable, effective, and efficient. They are thought to be more reliable because fewer people have influence over them and those people are farther away from the application. They are thought to be more effective because they are more resistant to bypass. They are thought to be more efficient because they operate across and are shared by a number of applications.

Nonetheless, application controls have always been important. They are often more granular and specific than the environmental controls. It is usually more effective to say that those who can update the vendor name and address file cannot also approve invoices for payment than it is to say that Alice cannot see or modify Bob's data. Although it sometimes happens that the privilege to update names and addresses maps to one data object and the ability to approve invoices maps to another data object, this is not always true. Although it can always be true that the procedure to update names and addresses is in a different program from that to approve invoices, and although this may be coincidental, it usually requires intent and design.

However, in modern systems, the reliance on application controls goes up even more. Although the application builder may have some idea of the environment in which his program will run, his ability to specify it and control it may be very low. Indeed, it is increasingly common for applications to be written in cross-platform languages. These languages make it difficult for the author to know whether his program will run in a single-user system or a multi-user system, a single application system or a multi-application system. Historically, one relied on the environment to protect the application from outside interference or contamination; in modern systems one must rely on the application to protect itself from its traffic. In distributed systems, environmental controls are far less reliable than in traditional systems. It has become common, not to say routine, for systems to be contaminated by applications.

The fast growth of the industry suggests that people with limited experience are writing many programs. It is difficult enough for them to write code that operates well when the environment and the inputs conform to their expectation, much less when they do not.

The history of controls in applications has not been very good. Although programs built for the marketplace are pretty good, those built one-off specifically for an enterprise are often disastrous. What is worse, the same error types are manifesting themselves as seen 20 years ago. The fact that they get renamed, or even treated as novel, suggests that people are not taking advantage of the history. "Those who cannot remember the past are condemned to repeat it."[1]

This chapter identifies and discusses some of the more common errors and their remedies in the hope that there will be more reliable programs in the future. Although a number of illustrations are used to demonstrate how these errors are maliciously exploited, the reader is asked to keep in mind that most of the errors are problems *per se*.

Unenforced Restrictions

In the early days of computing, it was not unusual for program authors to respond to error reports from users by changing the documentation rather than changing the program. Instead of fixing the program such that a particular combination of otherwise legitimate input would not cause the program to fail, the programmers simply changed the documentation to say, "Do not enter this combination of inputs because it may cause unpredictable results." Usually, these results were so unpredictable that, while disruptive, they were not exploitable. Every now and then, the result was one that could be exploited for malicious purposes.

It is not unusual for the correct behavior of an application to depend on the input provided. It is sometimes the case that the program relies on the user to ensure the correct input. The program may tell the user to do A and not to do B. Having done so, the program then behaves as if the user will always do as he is told. For example, the programmer may know that putting alpha characters in a particular field intended to be numeric might cause the program to fail. The programmer might even place a caution on the screen or in the documentation that says, "Put only numeric characters in this field." What the programmer does not do is check the data or constrain the input such that the alpha data cannot cause an error.

Of course, in practice, it is rarely a single input that causes the application to fail. More often, it is a particular, even rare, combination of inputs that causes the failure. It often seems to the programmer as if such a rare combination will never occur and is not worth programming for.

Complexity

Complexity is not an error *per se.* However, it has always been one of the primary sources of error in computer programs. Complexity causes some errors and may be used to mask malice. Simplicity maximizes understanding and exposes malice.

Limiting the scope of a program is necessary but not sufficient for limiting its complexity and ensuring that its intent is obvious. The more one limits the scope of a program, the more obvious will be what it does. On the other hand, the more one limits the scope of all programs, the more programs one ends up with.

Human beings improve their understanding of complex things by subdividing them into smaller and simpler parts. The atomic unit of a computer program is an instruction. One way to think about programming is that it is the art of subdividing a program into its atomic instructions. If one were to reduce all programs to one instruction each, then all programs would be simple and easy to understand, but there would be many programs and the relationship between them would be complex and difficult to comprehend.

Large programs may not necessarily be more complex than short ones. However, as a rule, the bigger a program is, the more difficult it is to comprehend. There is an upper bound to the size or scope of a computer program that can be comprehended by a human being. As the size of the program goes up, the number of people that can understand it approaches zero and the length of time required for that understanding approaches infinity. Although one cannot say with confidence exactly where that transition is, neither is it necessary. Long before reaching that point, one can make program modules large enough to do useful work.

The issue is to strike a balance in which programs are large enough to do useful work and small enough to be easily understood. The comfort zone should be somewhere between and 10 and 50 verbs and between one complete function and a page.

Another measure of the complexity of a program is the total number of paths through it. A simple program has one path from its entry at the top to its exit at the bottom. Few programs look this way; most will have some iterative loops in them. However, the total number of paths may still be numbered in the low tens as

long as these loops merely follow one another in sequence or embrace but do not cross. When paths begin to cross, the total number of possible paths escalates rapidly. Not only does it become more difficult to understand what each path does, it becomes difficult simply to know if a path is used (i.e., is necessary) at all.

Incomplete Parameter Check and Enforcement

Failure to check input parameters has caused application failures almost since Day One. In modern systems, the failure to check length is a major vulnerability. Although modern databases are not terribly length sensitive, most systems are sensitive to input length to some degree or another.

A recent attack involved giving an e-mail attachment a name more than 64 kb in length. Rather than impose an arbitrary restriction, the designer had specified that the length be dynamically assigned. At lengths under 64 kb, the program worked fine; at lengths above that, the input overlaid program instructions. Neither the programmer, the compiler, nor the tester asked what would happen for such a length. At least two separate implementations of the function failed in this manner.

Yes, there really are people out there that are stressing programs in this way. One might well argue that one should not need to check for a file name greater than 64 kb in length. Most file systems would not even accept such a length. Why would anyone do that? The answer is to see if it would cause an exploitable failure; the answer is that it did.

Many compilers for UNIX permit the programmer to allocate the size of the buffer statically at execution time. This makes such an overrun more likely but improves performance. Dynamic allocation of the buffer is more likely to resist an accidental overrun but is not proof against attacks that deliberately use excessively long data fields.

These attacks are known generically as "buffer-overflow" attacks. More than a decade after this class of problem was identified, programs vulnerable to it continue to proliferate.

In addition to length, it is necessary to check code, data type, format, range, and for illegal characters. Many computers recognize more than one code type (e.g., numeric, alphabetic, ASCII, hexadecimal, or binary). Frequently, one of these may be encoded in another. For example, a binary number might be entered in either a numeric or alphanumeric field. The application program must ensure that the code values are legal in both code sets — the entry and display set and the storage set. Note that because modern database managers are very forgiving, the mere fact that the program continues to function may not mean that the data is correct. Data types (e.g., alpha, date, currency) must also be checked. The application itself and other programs that operate on the data may be very sensitive to the correctness of dates and currency formats. Data that is correct by code and data type may still not be valid. For example, a date of birth that is later than the date of death is not valid although it is a valid data type.

Incomplete Error Handling

Closely related to the problem of parameter checking is that of error handling. Numbers of employee frauds have their roots in innocent errors that were not properly handled. The employee makes an innocent error; nothing happens. The employee pushes the envelope; still nothing. It begins to dawn on the employee that she could make the error in the direction of her own benefit — and still nothing would happen.

In traditional applications and environments, such conditions were dangerous enough. However, they were most likely to be seen by employees. Some employees might report the condition. In the modern network, it is not unusual for such conditions to be visible to the whole world. The greater the population that can see a system or application, the more attacks it is likely to experience. The more targets an attacker can see, the more likely he is to be successful, particularly if he is able to automate his attack.

It is not unusual for systems or applications to fail in unusual ways when errors are piled on errors. Programmers may fail to program or test to ensure that the program correctly handles even the first error, much less for successive ones. Attackers, on the other hand, are trying to create exploitable conditions; they will try all kinds of erroneous entries and then pile more errors on top of those. Although this kind of attack may not do any damage at all, it can sometimes cause an error and occasionally cause an exploitable condition. As above, attackers may value their own time cheaply, may automate their attacks, and may be very patient.

Time of Check to Time of Use (TOCTU)

Recently, a user of a Web mail service application noticed that he could "bookmark" his Inbox and return to it directly in the future, even after shutting down and restarting his system, without going through log-on again.

On a Friday afternoon, the user pointed this out to some friends. By Saturday, another user had recognized that one of the things that made this work was that his user identifier (UID), encoded in hexadecimal, was included in the universal record locator (URL) for his Inbox page. That user wondered what would happen if someone else's UID was encoded in the same way and put into the URL. The reader should not be surprised to learn that it worked. By Sunday, someone had written a page to take an arbitrary UID encoded in ASCII, convert it to hexadecimal, and go directly to the Inbox of any user. Monday morning, the application was taken down.

The programmer had relied on the fact that the user was invited to logon before being told the URL of the Inbox. That is, the programmer relied on the relationship between the time of the check and the time of use. The programmer assumes that a condition that is checked continues to be true. In this particular case, the result of the decision was stored in the URL, where it was vulnerable to both replay and interference. Like many of the problems discussed here, this one was first documented almost 30 years ago.

Now the story begins to illustrate another old problem.

Ineffective Binding

Here, the problem can be described as ineffective binding. The programmer, having authenticated the user on the server, stores the result on the client. Said another way, the programmer stores privileged state in a place where he cannot rely on it and where he is vulnerable to replay.

Client/server systems seem to invite this error. In the formal client/server paradigm, servers are stateless. That is to say, a request from a client to a server is atomic; the client makes a request, the server answers and then forgets that it has done so.

To the extent that servers remember state, they become vulnerable to denial-of-service attacks. One such attack is called the Syn Flood Attack. The attacker requests a TCP session. The victim acknowledges the request and waits for the attacker to complete and use the session. Instead, the attacker requests yet another session. The victim system keeps allocating resources to the new sessions until it runs out.

Because the server cannot anticipate the number of clients, it cannot safely allocate resource to more than one client at a time. Therefore, all application states must be stored on the clients. The difficulty with this is that it is then vulnerable to interference or contamination on the part of the user or other applications on the same system. The server becomes vulnerable to the saving, replicating, and replay of that state.

Therefore, at least to the extent that the state is privileged, it is essential that it be saved in such way as to protect the privilege and the server. Because the client cannot be relied on to preserve the state, the protection must rely on secret codes.

Inadequate Granularity of Controls

Managers often find that they must give a user more authority than they wish or than the user needs because the controls or objects provided by the system or application are insufficiently granular. Stated another way, they are unable to enforce usual and normal separation of duties. For example, they might wish to assign duties in such a way that those who can set up accounts cannot process activity against those accounts, and vice versa. However, if the application design puts both capabilities into the same object (and provides no alternative control), then both individuals will have more discretion than management intends. It is not unusual to see applications in which all capabilities are bundled into a single object.

Gratuitous Functionality

A related but even worse design or implementation error is the inclusion in the application of functionality that is not native or necessary to the intended use or application. Because security may depend on the system doing only what is intended, this is a major error and source of problems. In the presence of such functionality,

not only will it be difficult to ensure that the user has only the appropriate application privileges but also that the user does not get something totally unrelated.

Recently, the implementer of an E-commerce Web server application did the unthinkable; he read the documentation. He found that the software included a script that could be used to display, copy, or edit any data object that was visible to the server. The script could be initiated from any browser connected to the server. He recognized that this script was not necessary for his use. Worse, its presence on his system put it at risk; anyone who knew the name of the script could exploit his system. He realized that all other users of the application knew the name of that script. It was decided to search servers already on the Net to see how many copies of this script could be found. It was reported that he stopped counting when he got to 100.

One form of this is to leave in the program hooks, scaffolding, or tools that were originally intended for testing purposes. Another is the inclusion of backdoors that enable the author of the program to bypass the controls. Yet another is the inclusion of utilities not related to the application. The more successful and sensitive the application, the greater the potential for these to be discovered and exploited by others. The more copies of the program in use, the bigger the problem and the more difficult the remedy.

One very serious form of gratuitous functionality is an escape mechanism.

Escape Mechanisms

One of the things that Ken Thompson pointed out is the difficulty maintaining the separation between data and procedure. One man's data is another man's program. For example, if one receives a file with a file name extension of .doc, one will understand that it is a document, that is, data to be operated on by a word processing program. Similarly, if one receives a file with .xls, one is expected to conclude that this is a spreadsheet, data to be operated on by a spreadsheet program. However, many of these word processing and spreadsheet application programs have mechanisms built into them that permit their data to escape the environment in which the application runs. These programs facilitate the embedding of instructions, operating system commands, or even programs, in their data and provide a mechanism by which such instructions or commands can escape from the application and get themselves executed on behalf of the attacker but with the identity and privileges of the user.

One afternoon, the manager of product security for several divisions of a large computer company received a call from a colleague at a famous security consulting company. The colleague said that a design flaw had been discovered in one of the manager's products and that it was going to bring about the end of the world. It seems that many terminals had built into them an escape mechanism that would permit user A to send a message to user B that would not display but would rather be returned to the shared system looking as if it had originated with user B. The message might be a command, program, or script that would then be interpreted as if it had originated with user B and had all of user B's privileges.

The manager pointed out to his colleague that most buyers looked at this "flaw" as a feature, were ready to pay extra for it, and might not consider a terminal that did not have it. The manager also pointed out that his product was only one of many on the market with the same feature and that his product enjoyed only a small share of the market. And, furthermore, there were already a million of these terminals in the market and that, no matter what was offered or done, they would likely be there five years hence. Needless to say, the sky did not fall and there are almost none of those terminals left in use today.

On another occasion, the manager received a call from another colleague in Austin, Texas. It seems that this colleague was working on a mainframe e-mail product. The e-mail product used a formatter produced by another of the manager's divisions. It seems that the formatter also contained an escape mechanism. When the exposure was described, the manager realized that the work required to write an exploit for this vulnerability was measured in minutes for some people and was only low tens of minutes for the manager.

The behavior of the formatter was changed so that the ability to use the escape mechanism could be controlled at program start time. This left the question of whether the control would default to "yes" so that all existing uses would continue to work, or to "no" so as to protect unsuspecting users. In fact, the default was set to the safe default. The result was that tens of thousands of uses of the formatter no longer worked, but the formatter itself was safe for the naïve user.

Often these mechanisms are legitimate, indeed even necessary. For example, MS Word for DOS, a single-user single-tasking system, required this mechanism to obtain information from the file system or to allow the

user access to other facilities while retaining its own state. In modern systems, these mechanisms are less necessary. In a multi-application system, the user may simply "open a new window;" that is, start a new process.

Nonetheless, although less necessary, these features continue to proliferate. Recent instances appear in MS Outlook. The intent of the mechanisms is to permit compound documents to display with fidelity even in the preview window. However, they are being used to get malicious programs executed. All such mechanisms can be used to dupe a user into executing code on behalf of an attacker. However, the automation of these features makes it difficult for the user to resist, or even to recognize, the execution of such malicious programs.

They may be aggravated when the data is processed in an exceptional manner. Take, for example, so-called "Web mail." This application turns two-tier client/server e-mail into three-tier. The mail agent, instead of running as a client on the recipient's system, runs as a server between the mail server and the user. Instead of accessing his mail server using an application on his system, the user accesses it via this middleware server using his (thin client) browser. If HTML tags are embedded in a message, the mail agent operating on the server, like any mail agent, will treat them as text. However, the browser, like any browser, will treat these tags as tags to be interpreted.

In a recent attack, HTML tags were included in a text message and passed through the mail agent to the browser. The attacker used the HTML to "pop a window" labeled "….Mail Logon." If the user were duped into responding to this window, his identifier and password would then be broadcast into the network for the benefit of the attacker.

Although experienced users would not be likely to respond to such an unexpected log on window, many other users would. Some of these attacks are so subtle that users cannot reasonably be expected to know about them or to resist their exploitation.

Excessive Privilege

Many multi-user, multi-application systems such as the IBM AS/400 and most implementations of UNIX contain a mechanism to permit a program to run with privileges and capabilities other than those assigned to the user. The concept seems to be that such a capability would be used to provide access control more granular and more restrictive than would be provided by full access to the data object. Although unable to access object A, the user would be able to access a program that was privileged to access object A but which would show the user a only a specified subset of object A.

However, in practice, it is often used to permit the application to operate with the privileges of the programmer or even those of the system manager. One difficulty of such use is manifest when the user manages to escape the application to the operating system, but retain the more privileged state. Another manifests itself when a started process, subsystem, or daemon runs with excessive privilege. For example, the mail service may be set up to run with the privileges of the system manager rather than with a profile created for the purpose. An attacker who gains control of this application, for example by a buffer overflow or escape mechanism, now controls the system, not simply with the privileges required by the application or those of the user, but with those of the system manager.

One might well argue that such a coincidence of a flawed program with excessive privilege is highly unlikely to occur. However, experience suggests that it is not only likely, but also common. One might further argue that the application programmer causes only part of this problem; the rest of it is the responsibility of the system programmer or system manager. However, in practice, it is common for the person installing the program to be fully privileged and to grant to the application program whatever privileges are requested.

Failure to a Privileged State

Application programs will fail, often for reasons completely outside of their control, that of their programmers, or of their users. As a rule, such failures are relatively benign. Occasionally, the failure exposes their data or their environment.

It is easiest to understand this by comparing the possible failure modes. From a security point of view, the safest state for an application to fail to is a system halt. Of course, this is also the state that leaves the fewest options for the user and for system and application management. They will have to reinitialize the system,

reload and restart the application. While this may be the safest state, it may not be the state with the lowest time to recovery. System operators often value short time to recovery more than long time to failure.

Alternatively, the application could fail to log on. For years, this was the failure mode of choice for the multi-user, multi-application systems of the time. The remedy for the user was to log on and start the application again. This was safe and fairly orderly.

In more modern systems like Windows and UNIX, the failure mode of choice is for the application to fail to the operating system. In single-user, multi-application systems, this is fairly safe and orderly. It permits the user to use the operating system to recover the application and data. However, although still common in multi-user, multi-application systems, this failure mode is more dangerous. Indeed, it is so unsafe that crashing applications has become a favored manner of attacking systems that are intended to be application-only systems. Crash the application and the attacker may find himself looking at the operating system (command processor or graphical user interface [GUI]) with the identity and privileges of the person who started the application. In the worst case, this person is the system manager.

Unsafe Defaults

Even applications with appropriate controls often default to the unsafe setting of those controls. That is to say, when the application is first installed and until the installing user changes things, the system may be unsafely configured. A widespread example is audit trails. Management may be given control over whether the application records what it has done and seen. However, out of the box, and before management intervenes, the journals default to "off." Similarly, management may be given control of the length of passwords. Again, out of the box, password length may default to zero.

There are all kinds of good excuses as to why a system should default to unsafe conditions. These often relate to ease of installation. The rationale is that if the system initializes to safe settings, any error in the procedure may result in a deadlock situation in which the only remedy is to abort the installation and start over. The difficulty is that once the system is installed and running, the installer is often reluctant to make any changes that might interfere with it.

In some instances, it is not possible for designers or programmers to know what the safe defaults are because they do not know the environment or application. On the other hand, users may not understand the controls. This can be aggravated if the controls are complex and interact in subtle ways. One system had a control to ensure that users changed their passwords at maximum life. It had a separate control to ensure that it could not be changed to itself. To make this control work, it had a third control to set the minimum life of the password. A great deal of special knowledge was required to understand the interaction of these controls and their effective use.

Exclusive Reliance on Application Controls

The application designer frequently has a choice of whether to rely on application program controls, file system controls, database manager controls, or some combination of these. Application programmers sometimes rely exclusively on controls in the application program. One advantage of this is that one may not need to enroll the user to the file system or database manager or to define the user's privileges and limitations to those systems. However, unless the application is tightly bound to these systems, either by a common operating system or by encryption, a vulnerability arises. It will be possible for the user or an attacker to access the file system or database manager directly. That is, it is possible to bypass the application controls. This problem often occurs when the application is developed in a single-system environment, where the application and file service or database manager run under a single operating system and are later distributed.

Note that the controls of the database manager are more reliable than those in the application. The control is more localized and it is protected from interference or bypass on the part of the user. On the other hand, it requires that the user is enrolled to the database manager and that the access control rules are administered.

This vulnerability to control bypass also arises in other contexts. For example, controls can be bypassed in single-user, multi-application systems with access control in the operating system rather than the file system. An attacker simply brings his own operating system in which he is fully privileged and uses that in lieu of the operating system in which he has no privileges.

Recommendations

The following recommendation should be considered when crafting and staging applications. By adhering to these recommendations, the programmer and the application manager can avoid many of the errors outlined in this chapter.

1. Enforce all restrictions that are relied on.
2. Check and restrict all parameters to the intended length and code type.
3. Prefer short and simple programs and program modules. Prefer programs with only one entry point at the top or beginning, and only one exit at the bottom or end.
4. Prefer reliance on well-tested common routines for both parameter checking and error correction. Consider the use of routines supplied with the database client. Parameter checking and error correcting code is difficult to design, write, and test. It is best assigned to master programmers.
5. Fail applications to the safest possible state. Prefer failing multi-user applications to a halt or to log-on to a new instance of the application. Prefer failing single-user applications to a single-user operating system.
6. Limit applications to the least possible privileges. Prefer the privileges of the user. Otherwise, use a limited profile created and used only for the purpose. Never grant an application systemwide privileges. (Because the programmer cannot anticipate the environment in which the application may run and the system manager may not understand the risks, exceptions to this rule are extremely dangerous.)
7. Bind applications end-to-end to resist control bypass. Prefer a trusted single-system environment. Otherwise, use a trusted path (e.g., dedicated local connection, end-to-end encryption, or a carefully crafted combination of the two).
8. Include in an application user's privileges only that functionality essential to the use of the application. Consider dividing the application into multiple objects requiring separate authorization so as to facilitate involving multiple users in sensitive duties.
9. Controls should default to safe settings. Where the controls are complex or interact in subtle ways, provide scripts ("wizards"), or profiles.
10. Prefer localized controls close to the data (e.g., file system to application, database manager to file system).
11. Use cryptographic techniques to verify the integrity of the code and to resist bypass of the controls.
12. Prefer applications and other programs from known and trusted sources in tamper-evident packaging.

Note

1. George Santayana, *Reason in Common Sense*.

Domain 7
Operations Security

Operations security is used to identify the controls over hardware, media, and the operators with access privileges to any of these resources. Operations security involves the administrative management of all types of information processing operations, the concepts of security of centralized as well as distributed operations, the various choices for operations controls, resource protection requirements, auditing operations, monitoring, and intrusion detection.

To obtain a complete understanding of "operations security," it is necessary to look at it from a historical perspective. In the 1960s and 1970s, the processing of information took place in one central location and was controlled by only a handful of people. The term "data center" referred to this central location, while "operations" referred to the day-to-day data processing that occurred within the data center. The "operators" included the experienced, knowledgeable staff who performed the day-to-day operation of the computers.

Although data center operations are still in existence today, the term "operations security" now refers to the central location of all IT processing areas, whether it is called a data center, server room, or computing center.

This domain focuses on the security needs for this type of information system operations — that is, the needs of the data centers, server rooms, computer rooms — the back-end locations where information system processing is accomplished, and the personnel who have privileged access to the resources in these areas.

Contents

Section 7.4 Intrusion Detection

Section 7.5 Operations Controls

127

Operations: The Center of Support and Control

Kevin Henry, CISA, CISSP

The operations security domain encompasses all of the other domains of information systems security. This domain is where theory and design meet the reality of daily operations. Ideas, once only a concept, become a critical part of an organization's infrastructure. The policies and procedures developed in a conference room or through a rigorous review and approval process are enacted for the benefit and protection of the organization, the employees, and the various other stakeholders.

Operations entails control, procedures, and monitoring. It involves support for users, communication with outside business partners, emergency actions and response, and in many cases 24-hour vigilance.

There are several areas of operations security that we will look at in this chapter: the importance and types of controls, the role of production support, the use of good supervision, and the protection and continuity of business operations through backups, maintenance, and incident response.

The operations group has evolved over the years from a console-based mainframe administration group to the widespread network administration techies that provide critical support for users halfway around the globe. However, regardless of the environment, whether mainframe, single office, or multinational and multiple platform organizations, the key elements are the same. The operators (for the most part I will include only network administrators in this group) have high-level access and the ability to make or break many companies by virtue of this level of access. Operators execute tasks that often require some of the highest levels of authority on the system. They can see, touch, and alter almost anything. They are required to make decisions in pressure situations that may affect the ability of the organization to continue normal or alternative business operations.

The importance of an understanding of security and best practices is crucial for operations personnel. Operators need to be aware of availability, and their critical role in keeping systems running. They need to understand the risks of disclosure and the need to enforce confidentiality, which includes the concepts of privacy, secrecy, and trust (or confidence). Organizations are under increasing pressure to maintain the privacy of individuals — whether they are customers or employees. Many organizations are either required to, or have chosen to, declare their privacy policy. This is a meaningful statement and the operations group needs to be aware of the risks and potential liabilities to the organization if these policies are violated or disregarded. An organization often depends on the confidence of its customers. A foolish or negligent act — or even a perceived breach of this confidence — may impair the business activity of the organization for years to come. The final part of the information security triad is integrity. Integrity in this instance includes proper, accurate, or reliable processing, change control, storage, and behaviors. Often an operations group may be bound by Service Level Agreements (SLAs) and a failure to provide the contracted level of service prescribed in the SLA can affect the respect, reputation, and even financial viability of an operations group.

Many organizations today outsource operations and network admin functions. This chapter does not deal extensively with outsourcing; however, the concepts and requirements are in many ways similar. Outsource suppliers need to respect and honor contractual obligations and provide the required level of service and support. Suppliers may need to provide more than basic functionality — they may need to provide advice, warnings, recommendations, expertise, and value-added services. They may be the source of hardware, soft-

ware, and applications support, but moreover they may be providing the expertise and technical skills an organization relies on. No doubt this is a responsible and challenging role.

The firm that has decided to choose an outsourcing solution is relying on the strength of another company to provide the support and service it requires. This decision may have been based on a need for expertise the firm did not have in-house; it may have been a financial decision; it may have been in response to an immediate need that could not be provided through other channels. Whatever the reason for choosing an outsource solution, the organization is under the same pressure it would be if it was an in-house support group — that is, ensuring that the promised services are delivered and that the services meet the cultural, operational, and security requirements of the organization.

Controls

We will take a look at types of controls and how they may be used in an operations setting. First of all, it is important that controls are seen as a tool to be used prudently and reasonably. A control is a restriction or restraint. Moreover, a control is required to be used as a response to a risk. Once a risk has been identified — that is, we have established what the threats are and the likelihood that these threats will become a reality (or exposure) — then we need to set up controls to respond to these risks. A control may try to prevent a risk or it may be a way to detect a problem.

Preventive Controls

An ounce of prevention is worth a pound of cure. A preventive control is designed to stop an event from happening. It is a type of proactive control that relies on the establishment of procedures and tools that, hopefully, will catch and stop an adverse event from affecting the organization. There are many types of preventive controls and they are continuously changing as the risk environment, threats, cultures, markets, and regulatory conditions change. For example, a programmer who includes an edit in the data entry fields of an online system has implemented a preventive control.

Detective Controls

A detective control recognizes that some untoward activity either has taken place or is taking place, and institutes mechanisms to report, mitigate, limit, or contain the damage. It may also include logging or tracking functionality to record the details of the activity for use in subsequent analysis or possible disciplinary action. Detective controls include reviews and comparisons, audits, account reconciliations, input edit checks, checksums, and message degests.

Corrective Controls

Corrective controls are used when an event has caused some damage and it is necessary to restore or reconstitute operations to a normal or alternative operational state. They may be procedures for network isolation, restriction of traffic, forced lockout of most users, etc.

Compensating Controls

Sometimes no other control is possible. For example, we would not usually grant a user root-level or high-level access to a system. This principle of least privilege — granting a user only the minimal amount of access, authority, or privilege required to do his or her job — is an effective control.[1] It often prevents misuse, accidental errors, and curiosity-based discoveries, and mitigates many risks created by poor access control. However, in the case of network administrators and operators this control is not possible. Such personnel require a high level of access to run the utilities, execute jobs, change configurations, etc., that are a part of their routine duties. Because of this, we require compensating controls, controls that compensate or address a weakness in the control infrastructure that cannot be eliminated using normal controls. Compensating controls often use greater levels of supervision, monitoring, review of activity logs and separation of duties to prevent or detect the types of errors that may come from a weaker control environment.

The following control types are methods of implementing the types of controls listed earlier. An administrative control, for example, may be preventive, deterrent, or detective, depending on whether it is designed to be proactive or reactive. It may also be corrective where it sets forth escalation procedures and incident response programs.

Administrative

Administrative controls, often called "soft controls," are procedures and policies to provide direction and declare intent to users and affected personnel. Examples of administrative controls include change control, user registration, visitor logs, hiring and termination practices, punishment for failure to comply, roles, responsibilities and job descriptions, and privacy statements.

Technical or Logical

These types of controls are "hard" or functional controls, often depending on the use of tools, software, or hardware to restrict access, limit capabilities, or prevent virus infections, for example. A preventive technical control may be a firewall, or a detective control may include an intrusion detection system.

Physical

Physical controls are extremely important in this domain. Operators have responsibility for the core computing platforms and equipment used by the organization. Unauthorized access to these areas may result in catastrophic loss for an organization. All steps must be taken to protect equipment from damage — environmental (lightning, dust, smoke, extreme humidity or temperature conditions), utility-based (gas, water, sewer, or electrical problems), disaster (fire, flood, or structural failure), and man-made (vandalism, accidental damage). Physical controls include locking doors and telephone equipment closets, installing fire detection and suppression equipment, having uninterruptible power supplies and surge protectors and proper installation locations. The principle of separation of duties also applies to segregating the operations staff from other staff (especially programmers) so that no one can usurp the normal workflow procedures and the checks and balances that were established.

Documentation

One of the most important resources an operations department has is knowledge. It is remarkable therefore how many organizations do not have adequate documentation. Documentation is a key to understanding, maintaining, and reacting to system activities. When we look at incident response later, one of the key factors in mitigating the damage from an incident is to recognize that something is happening. In far too many cases an untoward event is not noticed in a timely manner just because no one knew what "normal" was. They had no record of usual or unusual activity, or if they did, no one looked at it, with the result that an attack or error was allowed to continue much longer than it should have.

When auditing an operations center, one of the first items reviewed should be the documentation of the systems. Where is it kept? Is there a copy off-site? Can it be accessed easily in a disaster? Is it up to date? Does it describe the systems? Does it show the interaction and interdependencies between systems? Does it show normal processing flows and does it contain lists of error codes and proper responses to errors?

Some of the documentation that must be provided includes inventory of equipment, location and configuration of hardware, networks, communications, storage, and support equipment. One firm recently had a major shutdown that lasted for several hours because an electrical circuit-breaker tripped and no one was able to find the electrical distribution panel that supplied the equipment.

A past incident log is often an excellent resource for an organization. It lists system failures, the actions taken, and people involved to correct the failures. Because certain failures may happen only occasionally and the same people may not be involved the next time there is a failure, an available listing of previous incidents and corrective procedures may dramatically reduce the time needed to repair this later failure. This document is also a valuable tool for the production support group, as we will review later in this chapter.

Operations

The Operations staff is responsible for the day-to-day operation and maintenance of a system. Whether the system is mainframe, client/server, PC based, or stand-alone, there needs to be personnel who are knowledge-

able about the system to ensure it is functioning properly, to perform maintenance and backup routines, upload patches and new configuration files, and schedule jobs, maintenance, and upgrades. These tasks may be performed by one group or a series of groups, depending on the size of the organization, the skill level of the staff, the risk involved, and the complexity of the network. Ideally, there still needs to be an exact series of checks and balances to ensure that all work is being done, that backups are performed (it is surprising how many times I have found instances where the backups encountered an error and had not run for several days and no one noticed)

Roles and Responsibilities within the Operations Area

The Operator

The operator is the person whose finger is on the pulse of the system. He or she is responsible for daily operations of the systems and applications, performing the routine maintenance work, and monitoring the system for failures, exceptions, and often balancing completed job runs to ensure correct completion.

The Scheduler

The scheduler's role in many organizations is to set up and coordinate jobs in preparation for execution. The scheduler is the person usually responsible for exceptional job runs or running tasks out of the ordinary job flow. The separation of scheduling and operations tasks allows a double check of the duties of the scheduler and, quite often, the scheduler is also tasked with double-checking the work of the operations group. It is imperative that all exception processing is documented and reviewed. When a job is run as an "override" or exception, the job may also need to be removed from the normal job stream so that it does not continue to run. All exceptions need to be submitted for approval and have backout or recovery procedures. A person knowledgeable about the exception should also be on call to ensure that recovery procedures can be enacted in the event of a failure.

The Librarian

The librarian is responsible for maintaining the various media that are entering or leaving production. Tapes, microfiche, CDs, DVDs, and reports may be passed between departments, business partners, regulatory agencies, clients, or vendors. The librarian is responsible to ensure that discarded media do not contain sensitive information, keeping an inventory of the various media and protecting the organization from corrupt or contaminated media. Distributing backup tapes to offsite storage and recovering aged backups for reuse are important tasks of the librarian. Finally, the librarian is usually responsible for moving updated programs and accompanying documentation into production, as one of the final steps in the change control process.

The Help Desk

One of the most visible activities of an operations group is the help desk, which in many cases provides a first-level support for the users. Often it is backed up with a second tier of support by applications or systems experts who respond to problems encountered that are beyond the skill of the help desk personnel or would require more time. The help desk is often the front line between the users and the information technology department. The responsiveness, availability, and friendliness of the help desk staff will often affect the overall attitude of the users to the IT department. Whether the users like or dislike systems and applications may be influenced by their interaction with the help desk. For that reason, continuous supervision of help desk functions should be utilized to gauge the attitude of the users and whether they feel that the help desk personnel are knowledgeable, helpful, and responsive.

The help desk requires specific training in social engineering. This department has tremendous power and privilege, and is often a target of manipulation by internal and external customers. One of the easiest methods of gaining unauthorized access to systems or data can be through cultivating a "friendship" with the help desk personnel. Access also may be gained through intimidation or coercion of help desk personnel and "bullying" them into providing an exception to the normal rules or procedures. A help desk is sometimes staffed by fairly low-paid and inexperienced personnel; oftentimes they are supporting personnel that they will never meet and at odd hours when managers or other experts may not be readily available. Therefore, care must be taken to set up procedures and workflows to assist the help desk personnel in executing their duties in a secure manner. If a person requires or demands some form of exception to the rules, the manner of approving this must be

established so that the help desk personnel are not forced or persuaded into breaking policy and jeopardizing operations.

One of the most common calls to a help desk is for password resets. This is a critically problematic area. Who is on the other end of the line? And how do we know that the person requesting the password reset is actually the true owner of the ID? Especially if the password is for an ID with high-level access, some form of controls must be set up to ensure that only the rightful owner of that ID can gain a password reset.

The help desk is often one of the last to know about a change to an application or system. This causes them grief when they begin to receive calls about an application they know nothing about. Therefore, help desk managers should be a part of all change control workflow so that they can ensure their staff is notified and trained on the new system prior to implementation. During a major revision to a system, it is good to have some applications or systems experts on call or even working in the help desk area to assist with problems and other questions.

All calls to a help desk should be logged and the logs reviewed regularly. Review of these logs may indicate problem areas or the need for training users or revising procedures to reduce repeated calls. This can also be put into a knowledge-based system to assist in answering future calls or in setting a menu option on an Integrated Voice Response (IVR) system. A help desk should also have a good communications system through phone and e-mail, including answering queues in case of high-traffic loads, and the ability to take messages instead of users reaching a busy or extended on-hold waiting period.

Production Support

Often closely related to the help desk function is a production support group. This group may operate as a second tier to the help desk, handling the production failures, user problems, and emergency fixes to applications. This group needs to be knowledgeable in systems, applications, programming, networks, security, and business unit requirements. Production support is often one of the first groups to learn about problems with applications, user interfaces, and external threats. In the event of a failure, production support should always review the actions taken by the response team. Thorough analysis may lead to better responses in the future, changes to procedures, but most importantly as a double check to detect errors in the recovery process. There have been several documented cases where an error made in the recovery process after hours should have been caught the following morning by production support, and yet, because this crucial double check was missing, it led to the failure of the entire corporation.

Production support is closely linked with quality assurance. When a change to an application, change to configurations, or new network connections are about to take place, production support should be aware of the changes and possibly review the changes to ensure that they are effective, complete, and follow organizational standards.

Monitoring of system activity is an important role of a production support group. Whereas the operators review at a level of job completion, error codes, etc., the production support personnel need to review CPU, bandwidth, and memory usage. Closely monitoring these activities may allow better forecasting of future resource needs so that equipment can be installed before availability becomes a concern, and some applications that are on the verge of failure due to insufficient resources may be provided additional support prior to a full-scale production failure. This data also assists in the scheduling of jobs so that production and maintenance windows can be maximized for ideal efficiency.

Incident Response

In the event of a system, application, communication, or peripheral component failure, the operations staff is commonly the first group to know of the failure. As mentioned before, this requires careful monitoring of network activity so that an abnormal condition is noticed as rapidly and identified as accurately as possible. Once identified, a proper and effective response is often detailed in procedural documentation. This may require notification of other departments, capturing of event information (for future analysis or forensic investigation), the alerting of key personnel, or the containment of the event through shutdowns or isolation.

Many operations are migrating toward automated alarm reporting or lights-out operations. These remove the reliance on the operators to be present or vigilant to detect abnormalities. These automated alerts may indicate anything from environmental problems such as fire or temperature, to network or hardware failures. These alarms need to be tested on a regular basis to ensure that they are functioning correctly and that they will alert the proper people. Often the call pattern for the alarm does not get changed when the personnel responsible for answering the alarm changes jobs.

All incidents should be documented so that analysis of the event can be performed. This will also permit the organization to learn from the event and establish new policies, countermeasures, or training to prevent future incidents.

Although operations staff may be familiar with recovery procedures, all recovery should be performed under the direct, careful supervision of skilled staff. This is similar to a medical setting where each person knows his or her limitations and a nurse, despite knowing the correct response, does not perform the responsibilities of a doctor. This allows checks and balances to prevent errors or omissions, or in some cases perhaps even malicious activity on the part of operations personnel.

Escalation procedures and guidelines should also be established. These will provide direction for operations staff about when and how to notify higher management of incidents. In most cases, it is best to notify too early rather than too late!

If the event is a major failure that will require extended recovery procedures, the operations room may become extremely busy and stressful. It is good to have conference rooms and communications set up nearby to permit the coordination of the recovery procedures without having overcrowded and poorly communicated facilities.

The operations group should also be represented on the Business Continuity Planning team. This team is responsible for continuity of business operations or recovery of operations in the event of a major failure to normal operations. The operations group should be knowledgeable about BCP plans and their role in a disaster. They also need to know the corporate priorities for recovery operations in the event that more than one system, application, or department is affected.

Supervision

Supervision is one of the most important factors in preventing, detecting, and mitigating errors, malfeasance, or other types of violations of policy, procedure, and operations. Because operations personnel have elevated authority and access to a system, they need extra oversight as a compensating control for this vulnerability. Quite often, many administrator and operator positions are considered entry-level jobs and the people in those positions may not be familiar with corporate policy, culture, loyalty, and regulations. They need frequent review and training to assist them in addressing their tasks securely and effectively. Because much of the effort for an operations group takes place after hours and during times of reduced network usage, the manager must also be prepared to attend the workplace and be available during off hours. This includes performing tests and drills after hours as well — fire, emergency response, network attack, etc.

Summary

Operations can be described as the heartbeat of most organizations today. For this reason, it requires careful maintenance, oversight, training, and coordination. When all of these factors are addressed, this department can be relied on to provide support and impetus for the organization — resulting in reliable processing, secure data handling, and the confidence of business units, business partners, users, shareholders, and regulatory groups.

Note

1. The author also likes to incorporate the condition of timing into the concept of least privilege — that is, that the user is granted the minimum amount of rights necessary to do his or her tasks for the shortest possible time.

128

Why Today's Security Technologies Are So Inadequate: History, Implications, and New Approaches

Steven Hofmeyr, Ph.D.

They grab headlines as they cut a wide swath of destruction through corporate America: viruses, worms (such as Code Red and Nimda), and hackers. The unfortunate fact is that even in organizations with extensive deployment of firewall, encryption, and intrusion detection systems (IDS), attacks still occur with alarming frequency. According to a Computer Security Institute/FBI survey of Fortune 1000 organizations that have suffered attacks, 91 percent had deployed firewalls and 61 percent had installed intrusion detection systems.

So, it is evident that although they provide some initial layers of protection for corporate systems, today's security tools have a distressing tendency to be several steps behind the latest exploits. This chapter explains why security technologies have evolved the way they have and describes the ways in which security systems need to adapt and change to meet the new demands of corporate information protection in the post-September 11 world.

Historical Perspective

Security for the first isolated mainframes focused primarily on physical access and the authentication and control of users. Experts believed that a provably correct security system could be built, based on the notion of a security kernel; that is, core security code that was verifiably secure. Confidence in this formal methods approach was so strong that researchers declared in 1973 that, "It is our firm belief that by applying these principles we can have secure shared systems in the next few years."[1] In the government's 1983 Orange Book, the most secure system is one that uses formal methods to prove the integrity of a "trusted code base." But these efforts failed because it is not possible to build a nontrivial, provably correct security system, any more than it is possible to write bug-free code.

In addition to security kernels, security teams in those days relied on monitoring user behavior. Audit systems collected extensive logs of user actions that human experts scanned periodically for potential threats. The emphasis was on accountability rather than on timely detection: if a compromise was detected, it was, by definition, an insider job. In the military/government context in which most computing took place, knowing which insiders were involved was key because they represented an ongoing threat to the organization.

With the increasing use of private networking, as well as the advent of the Arpanet and its evolution into the Internet in the late 1980s, it became possible for outsiders to penetrate computer systems. In addition, the interconnectedness of networked computers created a viable environment for new automated threats such as

worms. The most famous of these early automated threats was the Morris worm, which took down 25 percent of the Internet in 1988.

Security responses to these new threats continued to rely heavily on human expertise but there was a growing shift toward the network or perimeter, and away from the host, with such technologies as firewalls, which restrict network traffic, and network intrusion detection systems (NIDs), which scan network traffic for signatures of known attacks. However, these technologies are severely limited: firewalls cannot protect vulnerable applications that are legitimately accessed through the firewall, and NIDs suffer from notoriously high rates of false alarms and can only detect attacks already known to the signature writers.

Finally, another major source of security problems emerged with the advent of the desktop computer, which proved a fertile environment for viruses. Security solutions for protection against viruses focused on the host computer itself, in the form of anti-virus (AV) software. AV software maintains a database of virus signatures and scans files to determine if any are infected with known viruses. This technology is similar in principle to NIDs that use signatures and consequently has similar limitations; for example, it cannot detect new types of viruses. However, it is still successful at increasing the security of the desktop — the adoption of AV technology on the desktop is almost universal.

The Ever-Changing IT Landscape

Of course, the number of computers connected to the Internet continues to grow at a tremendous rate, and with this growth comes a dramatic increase in the numbers and types of threats. In particular, automated threats such as worms and e-mail viruses are on the rise. The notorious ILOVEYOU virus in 2000 is estimated to have affected upward of 10 million users and, more recently, the Code Red worm in 2001 infected over 150,000 systems in a mere 14 hours, resulting in billions of dollars in damages. In addition, the large number of vulnerable desktops connected to the Internet has encouraged distributed denial-of-service (DDoS) attacks, in which a collection of individual machines targets a single victim, bombarding it with traffic.

Not only are the numbers of connected computers increasing, but the patterns of connectivity are also changing. As more business is transacted over the Web, the boundaries between the "trusted" internal network and external networks are dissolving, requiring increasing use of encryption to protect communications in potentially hostile environments. Consequently, network-based security systems are becoming obsolete because they are predicated on the notion of a perimeter and need to scan the contents of network packets, which is not possible if the packets are encrypted.

In addition, today's IT environments are becoming exponentially more complex, incorporating a wider range of applications, middleware, and integration software. There are simply not enough experts to manage such complex systems, and experts cannot react fast enough to deal with the problems seen today. Meanwhile, few businesses today are as concerned about accountability as government organizations have been in the past. With mission-critical corporate data residing in vulnerable enterprise systems, today's corporations place a premium on prevention of attacks, rather than on catching or prosecuting the perpetrators after the fact.

From Human Expertise to Machine Intelligence

In response to these trends, security solutions are moving away from the network and back onto the host. Securing each host computer individually does not depend on defining a perimeter, and all processing of information can be done after traffic is decrypted at the host. Although this move back to the host is promising, other "old" ideas hold less potential. For example, just as in the days of mainframe computing, the security community is gravitating once again to the idea of "trusted" or "secure" systems, this time with a focus on trusted operating systems. Such operating systems attempt to put all applications and users into specific compartments and then limit functionality based on those compartments. However, if trusted computing could not be made to work in the single-machine mainframe era, it can hardly be expected to succeed now, in today's world of highly complex, interconnected, and vulnerable systems. The task of designing and verifying a set of policies for every variation of every application and operating system for any conceivable user requirement is, quite simply, infeasible.

What, then, is the answer? The security community must embrace a fundamental change in the way security systems are designed and built. A new security paradigm is needed, one based on machine intelligence, not human expertise. Security systems need to be self-aware, adaptive, and autonomous. They also need to focus

on prevention rather than just detection and source identification. Key to achieving these goals is the use of anomaly detection methods. With these methods, the computer security system observes the normal behavior of the computer to be protected, learns the profile of that normal behavior, and subsequently detects deviations (anomalies) from the profile that are indicative of attacks. The use of anomaly detection methods is the only way of detecting entirely new attacks; knowledge-based approaches that require knowing what the attacks look like beforehand will always fall short.

To ensure accuracy and avoid high false-alarm rates with anomaly detection, the system must monitor the appropriate characteristics. These characteristics should lead to a compact and stable profile under normal conditions but result in clear deviations from the normal profile during attacks. A poor choice of characteristic is exemplified by early research into anomaly detection that focused on user behavior. Users are inherently variable and thus any anomaly detection system profiling their behavior will generate masses of false alarms. A much better characteristic is paths through program code. If the program being profiled is a server, then its behavior is likely to be very consistent because servers repeatedly perform a few tasks and those tasks have predictable, regular code paths. The behavior of the server program is still driven by user behavior but that behavior is aggregated across many individuals and restricted through the program to a constrained, well-defined set of options.

Every anomaly detection system must have a training phase, during which the anomaly detection system develops a profile of normal behavior. In general, each system to be protected will exist in a different environment, with different configuration requirements and different usage patterns. These differences mean that it is essential for the anomaly detection system to learn the normal profile within each specific local environment. Moreover, even within a single system, the environment will vary over time. New software is added, old software is patched, configurations are changed, machines are removed or added, etc. Every time the system changes it has an effect on the normal profile of the system. Therefore, a key requirement of any useful anomaly detection system is the ability to adapt autonomously to changes in the environment. For example, each time a profiled program is updated to a more recent version, the anomaly detection system should "relearn" the normal behavior. The more similar the program's new behavior is to the old behavior, the more rapid the relearning; that is, the anomaly detection system does not need to throw away all the information it has previously learned.

Of course, the danger in making a system self-aware and more intelligent is that it becomes more difficult to understand what the system is doing and why. This is why any good anomaly detection system should have a comprehensive set of secondary analytics — additional information gathered about the anomaly that is not essential to detecting the anomaly. For example, it may be that anomalies are detected by simply monitoring program code paths, but when an unusual code path is reported, the network connections occurring at the time are also reported, to give a human operator more understanding of the anomaly. Secondary analytics can also be enhanced by the use of signatures, which can help humans understand the attacks through categorization of anomalies. This is different from traditional signature-based systems because the signatures are not used for detection, but only for informing human operators. In this way, the limitations and pitfalls of the signature-based approach are avoided.

Learning from History

A study of the history of computer security yields some guidelines that should be adhered to by any designer of a security technology for today's open, distributed, and highly interconnected systems.

- *Do not hard-code knowledge.* When designing security systems, people are often tempted to hard-code in their specific expertise about the problem at hand. For example, the designer might fervently believe that an application should never carry out a particular kind of behavior, and so hard-code in a restriction on that behavior. Succumbing to such temptations is shortsighted, destroys flexibility and adaptability, and is subject to human error and bias. It is well known in programming that hard-coding solutions to specific instances of a problem is a bad idea; the same applies to security.
- *Avoid the central weak point.* Designers like to have a system in which all information is gathered centrally at one point so that a human operator can control and monitor a large number of nodes from one location. This in itself is not a bad idea. However, placing too much dependence on the central location is. There is a trend today toward centralized correlation and analysis. Data is taken in from various sensors around the network, then analyzed and correlated in one location. If that one location should be compromised, then the entire security system will fail. The sensors themselves should be able to

react autonomously and independently so that even if the central location is compromised, they can continue to protect the network. And if correlation from multiple sensors is required, then this is more robust if done in a distributed peer-to-peer fashion.

- *There is no such thing as a trusted code base.* The resurgence of trusted operating systems is predicated on the belief in a trusted code base. In reality, there is no such thing. Only the most trivial, useless bit of code will be provably secure. Designers should operate under the assumptions that any part of the system is insecure and could be compromised. For example, across a set of distributed sensors, it must be assumed that some of them could be compromised or be in error. However, it can reasonably be assumed that not all of them will be compromised immediately, and so solutions can be designed that rely on voting and other forms of Byzantine agreement to isolate compromised sensors.
- *Profile actions, not data.* A detection system should monitor actions that are a consequence of an application receiving data, and not the data itself. Data, such as network packets, can be forged to look anomalous and flood a data-monitoring detection system with spurious alarms. Actions, by contrast, cannot be forged; if an action is successful, it means the system is truly vulnerable. Furthermore, it is difficult for a detection system to interpret data in exactly the same way as the application it is protecting. Errors in data interpretation are exploited by attackers to evade a detection system; for example, an attack can be hidden in fragmented packets if an NIDs does not properly reconstruct entire packets. Monitoring actions after the data has been interpreted by an application avoids this problem.
- *Do not compromise functionality for security.* A common mistake made when designing security systems is to focus on security measures, without regard to how those measures impact the functionality of the system being protected. The consequence is overly restrictive security systems. The problem with such overly restrictive systems is that legitimate users, in addition to attackers, find ways around the system. A good example is the firewall that restricts all access to and from the Internet, allowing only http traffic. This is sufficiently restrictive that nonmalicious users design their applications to run on top of http so that they can pass through firewalls. Consequently, more and more traffic is now running on top of http and the firewall is progressively more useless. Security systems must be designed with a clear regard for how they compromise functionality.

Summary

Today's computing world will never replicate the simplicity and central control of early mainframe environments; for better or worse, enterprise networks today are highly complex, interconnected, and vulnerable to automated and human threats. In moving away from the focus on accountability and the over-dependence on human expertise of past approaches, it is essential to embrace an automated, flexible, and highly adaptive approach — one that applies the best lessons from the past to consistently and reliably protect computing assets in the future.

A cornerstone of this new approach is anomaly detection systems that run on the host computers. These systems must be able to operate autonomously, monitoring appropriate characteristics to accurately profile normal and detect attacks, and using automated responses to stop attacks before they do harm. They should be able to adapt to legitimate changes with minimum human intervention. In addition, these anomaly detection systems should have comprehensive secondary analytics, including signature-based interpretation of anomalies. Of course, such systems will not guarantee security but, if implemented correctly, will raise security to a new level, taking a step ahead in the constant arms race between defender and attacker.

Information Warfare and the Information Systems Security Professional

Jerry Kovacich

Although the Cold War has ended, it has been replaced by new wars. These wars involve the use of technology as a tool to assist in conducting information warfare. It encompasses electronic warfare, techno-terrorist activities, and economic espionage. The term "information warfare" is being referred to as the twenty-first century method of waging war. The U.S., among other countries, is in the process of developing cyberspace weapons.

These threats will challenge the information security professional. The threats from the teenage hacker, company employee, and phreakers are nothing compared with what may come in the future. The information warfare warriors, with Ph.D.s in computer science backed by millions of dollars from foreign governments, will be conducting sophisticated attacks against U.S. company and government systems.

THE CHANGING WORLD AND TECHNOLOGY

The world is rapidly changing and, as the twenty-first century approaches, the majority of the nations of the world are entering the information age as described by Alvin and Heidi Toffler. As they discussed in several of their publications, nations have gone or are going through three waves or periods:

- The agricultural period, which according to the Tofflers ran from the time of humans to about 1745.
- The industrial period, which ran from approximately 1745 to the mid-1900s.
- The information period, which began in 1955 (the first time that white-collar workers outnumbered blue collar workers) to the present.

Because of the proliferation of technologies, some nations, such as, Taiwan and Indonesia, appear to have gone from the agricultural period almost directly into the information period. The U.S., as the information technology leader of the world, it is the most information systems-dependent country in the world and, thus, the most vulnerable.

What is meant by technology? Technology is basically defined as computers and telecommunications systems. Most of today's telecommunications systems are computers. Thus, the words telecommunications, technology, and computers are sometimes synonymous.

Today, because of the microprocessor, its availability, power, and low cost, the world is building the Global Information Infrastructure (GII). GII is the massive international connections of world computers that will carry business and personal communications, as well as those of the social and government sectors of nations. Some contend that it could connect entire cultures, erase international borders, support cyber-economies, establish new markets, and change the entire concept of international relations.

The U.S. Army recently graduated its first class of information warfare hackers to prepare for this new type of war. The U.S. Air Force, Army, and Navy have established information warfare (IW) centers. Military information war games are now being conducted to prepare for such contingencies.

INFORMATION AGE WARFARE AND INFORMATION WARFARE

Information warfare (IW) is the term being used to define the concept of twenty-first century warfare, which will be electronic and information systems driven. Because it is still evolving, its definition and budgets are unclear and dynamic.

Government agencies and bureaus within the Department of Defense all seem to have somewhat different definitions of IW. Not surprisingly, these agencies define IW in terms of strictly military actions; however, that does not mean that the targets are strictly military targets.

Information warfare, as defined by the Defense Information Systems Agency (DISA) is "actions taken to achieve information superiority in support of national military strategy by affecting adversary information and information systems while leveraging and protecting our information and information systems." This definition seems to apply to all government agencies.

The government's definition of IW can be divided into three general categories: offensive, defensive, and exploitation. For example:

- Deny, corrupt, destroy, or exploit an adversary's information or influence the adversary's perception (i.e, offensive).
- Safeguard the nation and allies from similar actions (i.e., defensive), also known as IW hardening.

- Exploit available information in a timely fashion to enhance the nation's decision or action cycle and disrupt the adversary's cycle (i.e., exploitative).

In addition, the military looks at IW as including electronic warfare (e.g., jamming communications links); surveillance systems, precision strike (e.g., if a telecommunications switching system is bombed, it is IW); and advanced battlefield management (e.g., using information and information systems to provide information on which to base military decisions when prosecuting a war).

This may be confusing, but many, including those in the business sector, believe that the term *information warfare* goes far beyond the military-oriented definition. Some, such as Winn Schwartau, author and lecturer, have a broader definition of IW and that includes such things as hackers attacking business systems, governments attacking businesses, even hackers attacking other hackers. He divides IW into three categories, but from a different perspective. He believes that IW should be looked at by using these categories:

- *Level 1: Interpersonal Damage*. This is damage to individuals, which includes anything from harassment, privacy loss, and theft of personal information, for example.
- *Level 2: Intercorporate Damage*. This is attacks on businesses and government agencies, which includes such things as theft of computer services and theft of information for industrial espionage.
- *Level 3: International and Intertrading Block Damage*. This relates to the destabilization of societies and economies, which includes terrorist attacks and economic espionage.

There seems to be more of the traditional, business-oriented look at what many call computer or high-tech crimes. By using the traditional government view of information warfare, the case can be made for Level 2 and Level 3 coming closest to the government's (i.e., primarily the Department of Defense) view of information warfare.

Then, there are those who tend to either separate or combine the term information warfare and information age warfare. To differentiate between these two terms is not that difficult. By using the Tofflers' thoughts about the three waves as a guide, as previously discussed information age warfare can be defined as warfare fought in the information age, with information age computer-based weapons systems, primarily dominated by the use of electronic and information systems. It is not this author's intent to establish an all-encompassing definition of IW, but only to identify it as an issue to consider when discussing information and information age warfare. Further, those information systems security professionals within the

government, and particularly those in the Department of Defense, will probably use any definition as it relates to military actions.

Those information systems security professionals within the private business sector (assuming that they were interested in using the term information warfare) would probably align themselves closer to Mr. Schwartau's definition. Those information systems security professionals within the private sector who agree with the government's definition would probably continue to use the computer crime terminology in lieu of Mr. Schwartau's definition.

The question arises if information warfare is something that the nongovernment business-oriented information systems security professional should be concerned about. Each information systems security professional must be the judge of that based on his or her working environment and also on how he or she see things from a professional viewpoint. Regardless, information warfare will grow in importance as a factor to consider, much as viruses, hackers, and other current threats must be considered.

The discussion of information warfare can be divided into three primary topics:

- Military-oriented war.
- Economic espionage.
- Technology-oriented terrorism (i.e., techno-terrorism).

MILITARY-ORIENTED WAR

The military technology revolution is just beginning. In the U.S., the military no longer drives technology as it once did in the 1930s through the 1970s. The primary benefactor of early technology was the government, primarily the Department of Defense (DoD), which in those early days of technology (e.g., ENIAC) the DoD had funding and the biggest need for technology. This was the time of both hot wars and the Cold War. The secondary benefactor was NASA (e.g., space exploration).

Between these government agencies, and to a lesser extent others, hardware and software products were developed with a derivative benefit to the private, commercial, and business sector. After all, these were expensive developments and only the government could afford to fund such research and development efforts. Today, the government has taken a back seat to the private sector. As hardware and software became cheaper, it became more cost effective for private ventures into technology research, development, and production. Now, technology is being business driven. Computers, microprocessors, telecommunications, satellites, faxes, video, software, networks, the Internet, and multimedia are just some of the technologies that are driving the information period. In the U.S., more than 95% of military communications are conducted over commercial systems.

In the next century, an increased use of technology will be used to fight wars. Stealth, surveillance, distance, and precision strike will be key concepts. As information age nations rely more and more on technology and information, these systems will obviously become the targets during information warfare.

The information warfare techniques are necessary due, in part, to economics. Every economics student learns about the "guns or butter" theory. It is believed that society cannot afford to adequately fund those programs that support society, while at the same time provide for a strong military structure. As the world continues to increase competitively the resources, for example, funding for expensive weapons systems, are competing with the resources needed to support society and the economic competition, which can also be considered as a type of warfare. Thus, commercial off-the-shelf (COTS), cheap, and secure weapons are being demanded.

Another important factor forcing the use of information warfare as a type of warfare is that the majority of civilized nations, because of world communications systems, can witness the death and destruction associated with warfare. They demand an end to such death and destruction. Casualties are not politically acceptable. Furthermore, as in the case of the U.S., why should a country continue to be destroyed and, then after peace is restored, spend billions of dollars to rebuild what had been destroyed? In information warfare, the death and destruction will be minimized, with information and information systems primarily being the target for destruction.

This new environment will cause these changes:

- Large armies will convert to smaller armies.
- More firepower will be employed from greater distances.
- Ground forces will only be used to identify targets and assess damages.
- A blurring of air, sea, and land warfare will occur.
- E-mail and other long-range smart information systems weapons will be available.
- Smaller and stealthier ships will be deployed.
- Pilotless drones will replace piloted aircraft.
- Less logistical support will be required.
- More targeting intelligence will be available.
- Information will be relayed direct from sensor to shooter.
- Satellite transmissions will be direct to soldier, pilot, or weapon.
- Military middle-management staff will be eliminated.
- Field commanders will access information directly from drones, satellites, or headquarters on the other side of the world.
- Friend or foe will be immediately recognized.

Technology, Menu-Driven Warfare

Technology is available that can build a menu-driven system, with data bases to allow the IW commanders and warriors to "point and click" to attack the enemy. For example, an information weapons system could provide these menu-driven computerized responses:

- Select a nation.
- Identify objectives.
- Identify technology targets.
- Identify communications systems.
- Identify weapons.
- Implement.

The weapons can be categorized as attack, protect, exploit, and support systems. For example:

- *IW-Network Analyses (Exploit).* Defined as the ability to covertly analyze networks of the adversaries to prepare for their penetration to steal their information and shut them down.
- *Crypto (Exploit and Protect).* Defined as the encrypting of U.S. and allies' information so that it is not readable by those who do not have a need to know; the decrypting of the information of adversaries is to be exploited for the prosecution of information warfare.
- *Sensor Signal Parasite (Attack).* Defined as the ability to attach malicious code (e.g., virus, worms) and transmit that signal to the adversary to damage, destroy, exploit, or deceive the adversary.
- *Internet-Based Hunter Killers (Attack).* Defined as a software product that will search the Internet, identify adversaries' nodes, deny them the use of those nodes, inject disinformation, worms, viruses, or other malicious codes.
- *IW Support Services (Services).* Defined as those services to support the preceding or to provide for any other applicable services, including consultations with customers to support their information warfare needs. These services may include modeling, simulations, training, testing, and evaluations.

Some techniques that can be considered in prosecuting information warfare include:

- Initiate virus attacks on enemy systems.
- Intercept telecommunications transmissions and implant code to dump enemy data bases.
- Attach a worm to enemies' radar signal to destroy the computer network.
- Intercept television and radio signals and modify their content.
- Misdirect radar and content.

- Provide disinformation, such as bushes that look like tanks and trees that look like soldiers.
- Information overload enemy computers.
- Penetrate enemies' GII nodes to steal or manipulate information.
- Modify maintenance systems information.
- Modify logistics systems information.

ECONOMIC ESPIONAGE: A FORM OF INFORMATION WARFARE

In looking at rapid technology-oriented growth, there are nations of haves and have-nots. There are also corporations that conduct business internationally and those that want to. The international economic competition and trade wars are increasing. Corporations are finding increased competition and looking for the competitive edge or advantage.

One way to gain the advantage or edge is through industrial and economic espionage. Both forms of espionage have been around since there has been competition. However, in this information age the competitiveness is more time-dependent, more crucial to success, and has increased dramatically, largely due to technology. Thus, there is an increased use of technology to steal that competitive advantage and, ironically, these same technology tools are also what is being stolen. In addition, more sensitive information is consolidated in large data bases on internationally networked systems whose security is questionable.

Definitions of Industrial and Economic Espionage

Industrial espionage is defined as an individual or private business entity sponsorship or coordination of intelligence activity conducted for the purpose of enhancing a competitor's advantage in the marketplace. According to the FBI, economic espionage is defined as: "Government-directed, sponsored, or coordinated intelligence activity, which may or may not constitute violations of law, conducted for the purpose of enhancing that country's or another country's economic competitiveness."

Economics, World Trade, and Technologies

What has allowed this proliferation of technologies to occur? Much of it was due to international business relationships among nations and companies. Some of it was due to industrial and economic espionage.

The information age has brought with it more international businesses, more international competitors, and more international businesses working joint projects against international competitors. This has resulted in more opportunities to steal from partners. Moreover, one may be a business partner on one contract while competing on another; thus, providing the opportunity to steal vital economic information. Furthermore, the

world power of a country, today, is largely determined by its economic power. Thus, in reality, worldwide business competition is viewed by many as the economic war. This world competition, coupled with international networks and telecommunications links, has provided more opportunities for more people such as hackers, phreakers, and crackers to steal information through these networks. The end of the Cold War has also made many out-of-work spies available to continue to practice their craft, but in a capitalistic environment.

Proprietary Economic Information

This new world environment makes a corporation's proprietary information more valuable than previously. Proprietary economic information according to the FBI is "…all forms and types of financial, scientific, technical, economic, or engineering information including but not limited to data, plans, tools, mechanisms, compounds, formulas, designs, prototypes, processes, procedures, programs, codes, or commercial strategies, whether tangible, or intangible… and whether stored, compiled, or memorialized physically, electronically, graphically, photographically, or in writing…". This statement assumes that the owner takes reasonable measures to protect it, and that it is not available to the general public.

A security association's survey taken among 32 corporations disclosed that proprietary information had been stolen from their corporations. These thefts included research, proposals, plans, manufacturing information, pricing, and product information. The costs to these corporations were substantially in terms of legal costs, product loss, administrative costs, lost market share, security cost increases, research and development costs, and loss of corporate image in the eyes of the public.

Economic Espionage Vulnerabilities

The increase in economic espionage is also largely due to corporate vulnerabilities to such threats. Corporations do not adequately identify and protect their information, nor do they adequately protect their computer and telecommunications systems. They do not have adequate security policies and procedures; employees are not aware of their responsibilities to protect their corporation's proprietary information. Many of the employees and also the management of these corporations do not believe that they have any information worth stealing or believe that it could happen to them.

Economic Espionage Risks

When corporations fail to adequately protect their information they are taking risks that will in all probability cause them to lose market share, profits, business, and also help in weakening the economic power of their country.

These are some actual cases of economic espionage:

- A foreign government intelligence service compiled secret dossiers of proprietary proposals of two companies from two other countries. Then, they gave that information to one of their country's companies, also bidding on the same contract. Their country's company won a billion dollar contract.
- A company contracted with a foreign government for a product. After disagreements, the government gave the proprietary information to one of their own companies.
- Foreign businessmen were arrested in a government agent sting operation for stealing proprietary information from their competitor.
- An employee of a U.S. microprocessor corporation admitted selling technology information from two companies where he had been employed. The information was alleged to have been sold to China, Iran, and Cuba.
- A foreign company, which could be a foreign government-fronted company, buys into a contract at a bid below its costs. They used the opportunity to steal technology information to be used by their country.

How Safe Are We? According to the International Trade Commission, the loss to U.S. industries due to economic espionage in 1987 was $23.8 billion and in 1989 was $40 billion. Today, these losses are projected to be over $70 billion. During the same time, the American Society for Industrial Security found that U.S. companies only spent an average of $15,000 per year to protect their proprietary information.

It was determined by one survey that only 21% of the attempted or actual thefts of proprietary information occurred in overseas locations, indicating that major threats are U.S. based. A CIA survey found that 80% of one country's intelligence assets are directed towards gathering information on the U.S. and to a lesser degree towards Europe. The FBI indicates that of 173 nations, 57 were actively running operations targeting U.S. companies and over 100 countries spent some portion of their funds targeting U.S. technologies. It was determined that current and former employees, suppliers, and customers are said to be responsible for over 70% of proprietary information losses. No one knows how much of those losses are due to foreign government-sponsored attacks.

Economic Espionage Threats

Economic espionage — that espionage supported by a government to further a business — is becoming more prevalent, more sophisticated, and easier to conduct due to technology. Business and government share a responsibility to protect information in this information age of international business competition.

Businesses must identify what needs protection; determine the risks to their information, processes, and products; and develop, implement, and maintain a cost-effective security program. Government agencies must understand that what national and international businesses do affects their country. They must define and understand their responsibilities to defend against such threats, and they must formulate and implement plans that will assist their nation in the protection of its economy. Both business and government must work together, because only through understanding, communicating, and cooperating will they be able to assist their country in the world economic competition.

It is quite obvious from the preceding discussion that when it comes to economic espionage, a new form of information warfare, the information systems security professional must play an active role in the economic information protection efforts. These efforts will help protect U.S. companies or government agencies and will enhance the U.S.'s ability to compete in the world economy.

TERRORISTS AND TECHNOLOGY (TECHNO-TERRORISTS): A FORM OF INFORMATION WARFARE

The twenty-first century will bring an increased use of technology by terrorists. Terrorism is basically the use of terror or violence, or the use of violent and terrifying actions for political purposes by a government to intimidate the population or by an insurgent group to oppose the government in power. The FBI defines terrorism as: "…the unlawful use of force or violence against persons or property to intimidate or coerce a government, the civilian population, or any segment thereof, in furtherance of political or social objectives."

The CIA defines international terrorism as: "…terrorism conducted with the support of foreign governments or organizations and/or directed against foreign nations, institutions, or governments." The Departments of State and Defense define terrorism as: "…premeditated, politically motivated violence perpetrated against a non-combatant target by sub-national groups or clandestine state agents, usually intended to influence an audience. International terrorism is terrorism involving the citizens or territory of more than one country." Therefore, a terrorist is anyone who causes intense fear and who controls, dominates, or coerces through the use of terror.

Why Are Terrorist Methods Used?

Terrorists generally use terrorism when those in power do not listen, when there is no redress of grievances, or when individuals or groups oppose current policy. Terrorists find that there is usually no other recourse available. A government may want to use terrorism to expand its territory or influence another country's government.

What Is a Terrorist Act?

In general, it is what the government in power says it is. Some of the questions that arise when discussing terrorism are

- What is the difference between a terrorist and a freedom fighter?
- Does "moral rightness" excuse violent acts?
- Does the cause justify the means?

The Results of Terrorist Actions

Acts of terrorism tend to increase security efforts. It may cause the government to decrease the freedom of its citizens to protect them. This, in turn, may cause more citizens to turn against the government, thus supporting the terrorists. It also causes citizens to become aware of the terrorists and their demands.

The beginning of this trend can be seen in the U.S. Americans are willing to give up some of their freedom and privacy to have more security and personal protection. Examples include increased airport security searches and questioning of passengers.

Terrorists cause death, damage, and destruction as a means to an end. Sometimes, it may cause a government to listen, and it may also cause social and political changes. Current terrorist targets have included transportation systems, citizens, buildings, and government officials.

Terrorists' Technology Threats

Today's terrorists are using technology to communicate and to commit crimes to fund their activities. They are also beginning to look at the potential for using technology in the form of information warfare against their enemies. It is estimated that this use will increase in the future.

Because today's technology-oriented countries rely on vulnerable computers and telecommunications systems to support their commercial and government operations, it is becoming a concern to businesses and government agencies throughout the world. The advantage to the terrorist of attacking these systems is that the techno-terrorist acts can be done with little expense by a few people and yet cause a great deal of damage to the economy of a country. They can conduct such activities with little risk to themselves, because these systems can be attacked and destroyed from a base in a country that is friendly to them. In addition, they can do so with no loss of life; thus not causing the extreme backlash against them as would occur had they destroyed buildings, causing much loss of life.

These are some actual and potential techno-terrorist actions:

- Terrorists, using a computer, penetrate a control tower computer system and send false signals to aircraft, causing them to crash in mid-air or fall to the ground.
- Terrorists use fraudulent credit cards to finance their operations.
- Terrorists penetrate a financial computer system and divert millions of dollars to finance their activities.
- Terrorists bleach $1 bills and, by using a color copier, reproduce them as $100 bills and flood the market with them to destabilize the dollar.
- Terrorists use cloned cellular phones and computers over the Internet to communicate, using encryption to protect their transmissions.
- Terrorists use virus and worm programs to shut down vital government computer systems.
- Terrorists change hospital records, causing patients to die because of an overdose of medicine or the wrong medicine. They may also change computerized tests and alter the results.
- Terrorists penetrate a government computer and causes it to issue checks to all its citizens.
- Terrorists destroy critical government computer systems processing tax returns.
- Terrorists penetrate computerized train routing systems, causing passenger trains to collide.
- Terrorists take over telecommunications links or shut them down.
- Terrorists take over satellite links to broadcast their messages over televisions and radios.

Some may wonder if techno-terrorist activities can actually be considered as information warfare. Most IW professionals believe that techno-terrorism is part of IW, assuming that the attacks are government sponsored and that the attacks are done in support of a foreign government's objectives.

DEFENDING AGAINST INFORMATION WARFARE ATTACKS

To defend against information warfare attacks, the information systems security professional must be aggressive and proactive. Now, as in the past, the basic triad of information security processes are usually installed:

- Individual accountability.
- Access control.
- Audit trail systems.

This passive defense kept the honest user honest, but did not do much to stop the more computer-literate user such as the hacker, cracker, or phreaker. Management support was not always available unless something went wrong. Then, management became concerned with information systems security — albeit only until the crisis was over. This passive

approach, supported by short-lived proactive efforts, was and continues to be "how information security is done."

With the advent and concerns associated with information warfare, government agencies, businesses, and the U.S. in general can no longer afford to take such a passive approach. As a profession, the possibility of an information systems Pearl Harbor is discussed. Most of the time, this is dismissed as rhetoric, and that security people are trying to justify their budgets. This approach will no longer work, and security professionals would be remiss in their responsibilities if they did not start looking at how to "information warfare-harden" (IW-H) computerized systems. IW-H means to provide a defensive shield — an early warning countermeasures system to protect government and business information infrastructures in the event of IW attacks.

Attacking a Commercial Target May Be a Prelude to War

In a time of war, would government systems be the primary target? A new age in warfare, commonly known as the Revolution in Military Affairs (RMA), is being entered. As previously discussed, there is a worldwide economic war being waged, where balance of trade statistics determine the winners and losers, along with the unemployment trends and the trends indicating the number of businesses moving overseas. In the information systems business, that trend also continues and may be increasing. Microprocessors are made in Malaysia and Singapore, software is written in India, and systems are integrated and shipped from Indonesia, for example. No one checks to determine if malicious code is embedded in the firmware or software, waiting for the right sequence of events to be activated to release that new, devastating virus or to reroute information covertly to adversaries.

Consideration must also be given to networking with other information systems security professionals to establish an IW early warning network, as well as to share IW defensive and IW countermeasures information. This can be equated somewhat with the early warning radar sites that the Department of Defense has scattered throughout the U.S.'s sphere of influence. These systems warn against impending attacks. If such a system was in place on the Internet when the Morris Worm was initiated, the damage could have been minimized and the recovery completed much quicker. If the U.S. is the object of all-out IW attacks, the Morris Worm type of problem would be nothing compared with the work of government-trained IW attack warriors.

SUMMARY

When a government agency or business computer system is attacked, the response to such an attack will be based on the type of attacker. Will the attacker be a hacker, phreaker, cracker, or just someone breaking in for

fun? Will the attacker be an employee of a business competitor, or in the case of an attack on a business system will it be a terrorist or a government agency-sponsored attack for economic reasons? Will the attacker be a foreign soldier attacking the system as a prelude to war?

These questions require serious consideration when information systems are being attacked, because it dictates the response. Would one country attack another because of what a terrorist or economic spy did to a business or government system? To complicate the matter, what if the terrorist was in a third country but only made it look like as though he or she was coming from a potential adversary? The key to the future is in information systems security for defense and information warfare weapons. As with nuclear weapons used as a form of deterrent, in the future, information weapons systems will be the basis of the information warfare deterrent.

Steps for Providing Microcomputer Security

Douglas B. Hoyt

Payoff

Microcomputers are most often associated with end-user computing, which traditionally have not been the systems development manager's responsibility. Nevertheless, senior management often holds the systems development manager responsible for an organization's entire computing resources, which include microcomputer systems. In addition, the rapid-proliferation of microcomputers throughout most organizations makes it important for the systems development manager to take an interest in microcomputer security issues. This article examines how the systems development manager can provide security for microcomputers and examines issues particular to microcomputer security.

Problems Addressed

Microcomputers have brought many benefits to business, but these benefits can be undermined if these desktop machines are not properly secured. sensitive information stored in a microcomputer is vulnerable to theft or to being copied by unauthorized individuals. Lost data can be difficult, costly, or impossible to reconstruct, especially if its source is no longer available. Microcomputer systems are also vulnerable to hardware malfunctions, fluctuations in source power, operator errors, software bugs, and viruses. In addition, the quality of information produced by microcomputer systems can be questionable, because microcomputer systems are seldom subjected to the controls that apply to mainframe or minicomputer systems.

Ensuring the security of microcomputers is a complicated issue for systems development managers, who are typically responsible for the development of information systems and the supervision of centralized computer operations. Systems development managers may have less than full— or no—control over microcomputer systems (including LANs). Because senior management generally relies on systems development managers to ensure the integrity of an organization's overall computer systems, however, systems development managers not only must be able to ensure proper controls for computer-based operations under their direct control but must manage, or influence, the proper use of microcomputers or networks by end users not directly under their control. Their indirect influence over microcomputer security can be strengthened if systems development managers educate senior management about the importance of such security.

This article can help systems development managers educate senior management as well as themselves about microcomputer security. It examines the security vulnerabilities particular to microcomputer systems and ways of protecting these susceptible areas. Guidelines and a checklist for establishing and implementing microcomputer security practices are also provided.

Steps for Securing Microcomputers

Assessing Vulnerabilities and Needs

Potential vulnerabilities in microcomputer systems should be reviewed periodically and systematically. One possible method of review is to establish a schedule that lists all the organization's microcomputer systems, the potential areas of vulnerability for those systems (rated as a percentage or as high, medium, or low risk), the value of what could be

lost as a result of failure of the system, and steps for preventing such a loss. Updating this list every one or two years can ensure that major areas of vulnerability are not allowed to exist unprotected or undetected. This type of review requires appropriate fact gathering and analysis.

Such a structured analysis of microcomputer vulnerabilities can be used to organize a presentation to senior management to help gain its support for security measures. It could also help justify expenditures for security products that the analysis indicates are worthwhile. It is essential that senior management be made aware of microcomputer security vulnerabilities and needs, and it is often the systems development manager's responsibility to see that that is accomplished.

Ensuring Information and Software Backup

Most systems development managers have established methods and schedules for backing up software programs, data bases, and other transaction data so that the information can be available to resume interrupted operations. However, additional safeguards are needed to restart operations promptly after a major disaster. It is preferable to have two backups, each at a different location, so that recovery of operations does not depend on one backup alone. In addition, extra backups reflecting current transactions should be kept off site.

The systems development managers should first determine which microcomputer files, programs, and data bases are critical to operations. Then, each item should be analyzed and an appropriate backup method and locations selected. The off-site location may be another place within the organization, or it even could be furnished by a service that warehouses backup copies in electronic, microfilm, and paper form. These services usually provide carefully controlled storage environments, pickups and deliveries, and records of deliveries to and from their facilities.

Although most microcomputer and network users are aware of the value of backing up data and software, many do not do it properly. It is human nature to procrastinate with matters that do not seem urgent. To correct this tendency, a firm schedule must be set and followed. Even organizations that do keep backups often store them near the original; they must correct this situation by maintaining duplicate backup copies at another location.

There are three basic ways to back up Disk Operating System systems copies: the COPY command, the XCOPY command, or a backup and restore utility. The COPY command is limited to the capacity of a diskette. The XCOPY command can be used to copy a subdirectory with many files on it. A backup and restore utility dumps all data onto as many diskettes as are needed.

Protecting Hardware and Guaranteeing Availability

Common security measures for microcomputer hardware include locks and guards for the areas in which equipment is located. Microcomputer equipment can be protected from thieves by means of cables, locks, adhesives, bolts, and even alarms that sound when a turned-off microcomputer is moved. These are effective in deterring thefts, but there is no way of completely preventing the theft of such equipment.

Most microcomputer equipment can be readily replaced. In addition to the purchase costs, the time and cost of replacing the software and data sometimes can be comparable to the equipment loss itself, even if the software and data have been properly backed up.

One possible aid to recovering stolen equipment is keeping records of serial numbers and other such identification numbers that help police identify and retrieve microcomputer equipment. Another aid to identifying stolen equipment is the branding of disks with the owner's identification; branding is done by imprinting the owner's name or tax identification number on unused disk space. Some organizations have a large inventory of

microcomputers and maintain spare computers, parts, and peripherals to replace items that may be stolen or have become inoperable; such inventories are helpful in minimizing replacement time.

Implementing Access Controls

Access controls are needed to safeguard against such dangers as stealing data, wiping out or altering critical data, spying, and other malicious actions. Such dangers are all real and can justify the considerable time and cost involved in implementing protective measures. However, most damage from intrusions is inadvertently caused by employees who are not properly trained or who accidentally access a wrong program or data because of weak controls. The variety of risks from unauthorized access makes it necessary for systems development managers to study the possible dangers, weigh their consequences, systematically analyze employee functions, and install controls to ensure that access procedures to view and modify programs and data are logically designed. Easy access, especially in networks, can increase risk as well as efficiency.

For many systems, sufficient access control can be provided by requiring users to enter their individual IDs and passwords. Those passwords and IDs should allow users access to only those programs and data bases that they need to fulfill their individual responsibilities. The passwords must not be names or words that others might guess and should be changed regularly. An added control can be the documentation of who accesses what system and data base and when; this audit trail can verify conformity to authorize activities and may gave clues to any improper use.

When access to a microcomputer or network is over telephone lines, there is the danger that someone who has surreptitiously learned a password and ID could access the system. Several vendors have developed callback security products that prevent unauthorized access over the telephone. With these products, a caller enters a password, an ID, or both; the system then terminates the connection and places a call to the telephone number known to be the authorized source for the particular password or ID. The authorized caller is allowed to complete the transaction. A list of callback security products is given in Exhibit 1.

Callback Security Products

Networks increase the complexity of the access control problem. When microcomputers and workstations are connected to one or more servers and maybe to a minicomputer or mainframe, the users at each station must be prevented from having access to network data and programs that they do not need for their individual responsibilities. Their password and ID authorizations must be defined accordingly.

The network security problem is further complicated when networks are connected to networks. Without restriction, users can move from network to network through gateways, bridges, and routers. Several special security systems have been developed to address this multiple network problem. One such system, for example, is Intrusion Detection Expert System (IDES) designed by Stanford Research Institute International. IDES analyzes users' normal behavior patterns and alerts administrators to deviations that are symptoms of violations to be investigated.

Absolute computer security is impossible to achieve because any access obstacle can be overcome by a clever and determined technician. The more obstacles that are established, however, the less likely it will be that they will be surmounted. For example, invisible characters can be embedded in file or directory names; only authorized users know the key invisible characters, which prevent easy access to unauthorized persons. File names in the directories of DOS systems can be hidden from unauthorized persons using such utility programs as the Norton Utilities from Symantec Corp. or alarm, lockup, and encryption

The Modem Security Enforcer
IC Engineering, Inc.
P O Box 321
Owings Mills MD 21117

The 24E5 Secure Modem
Anchor Automation, Inc.
20675 Baham St.
Chatsworth CA 91311

The 424 Line Backer Security Modem
Western DataCom
P O Box 45113
Cleveland OH 44145

TraqNet 2001
LeeMah DataCom Security Corp.
3948 Trust Way
Hayward CA 94547

FDX 9696S V.32 High Speed Modem
Fastcomm Communications
24347 East Sunrise Valley Dr.
Reston, VA 22091

0S1821N Network Management System
Octocom Systems, Inc.
225 Ballardvale St.
Wilmington MA 01887

Auditor System 32
Millidyne, Inc.
3645 Trust Dr.
Raleigh NC 27604

Hack Attack! Modem Security
Calta Computer Systems, Ltd.
P O Box 815
Calgary AB, Canada T2H 2H3

Term Serv Modem Security
Qualtrak Corp.
3315 San Felipe Rd.
San Jose CA 95135

COMPUSAFE Network Security
COMPUSAFE
113 South Main St.
Nazareth PA 18064

Western Telematic SM-21
Modem Security
Western Telematic, Inc.
5 Sterling St.
Irvine CA 92718

features available with DOS systems. Data encryption can provide another layer of protection for extra-sensitive data. This is especially effective for data that is to be transmitted between locations.

Providing Power Protection and Backup

Another area that needs to be protected is the microcomputer system's power supply. Too much or too little power can do great harm if proper protective measures are not taken. Blackouts, brownouts, and sags can cause loss of data and disrupt computer operations. Strong surges such as those caused by lightning can seriously damage hardware. The two main preventives for power problems are surge protectors and uninterruptable power supplies (UPSs).

Surge protectors are used frequently because they are relatively inexpensive and prevent damage to data from minor power fluctuations that commonly occur. They cost from $25 to $30 and are installed simply by being inserted between the electrical outlet and the microcomputer.

There are two types of UPSs—online and offline. They both can safeguard against damage from lightning as well as ensure 15 minutes or more of power if the regular source is stopped. This gives time to save data and shut down the computers in an orderly fashion. Online UPSs, which can cost several thousand dollars, run continuously, providing even current from batteries. Offline UPSs also furnish even current from batteries, but their batteries provide power only when the regular source ceases or fluctuates. Because online UPSs are relatively expensive, their use is justified only for larger networks or for highly important operations. Offline UPSs, on the other hand, are available for less than $100.

Performing Maintenance and Housekeeping

Maintenance and housekeeping can reduce the likelihood of sudden microcomputer system failures. These failures are rare but should be avoided.

Hardware with moving parts is most prone to breakdown. Following are some procedures that can help avoid failure problems:

- Carefully following the manufacturer's recommendations for caring for equipment.
- Keeping the microcomputer vicinity free from dust and undue moisture, allowing air to flow freely around microcomputer equipment, and forbidding eating, drinking, and smoking in the area.
- Avoiding excessive moves of equipment.
- Minimizing static buildup by using grounded antistatic mats or antistatic carpet spray as well as by using a humidifier if the heating system dries the air.
- Using electrical outlets that are not connected to such devices as motors, heating appliances, for fluorescent lights.
- Keeping wires neatly bundled and away from where they could be hit by passersby.
- Cleaning disk drives and tape drives periodically with specially designed cleaning disks and tapes.
- Keeping copies of backup data off site to protect against theft or other disasters.

· Establishing plans for regular maintenance.

Providing Written Policies and Instructions

The auditors of a manufacturing company recently reviewed the company's applications on microcomputers and found many of them to contain such vital information as strategic plans, pricing, and invoices. Much of this key information was not backed up, and the auditors were concerned about the vulnerability of such important operations.

To correct this security weakness, the auditors initiated programs to establish manuals of standards and policies and a training program to instruct all concerned in proper safeguarding principles. In fact, manuals for microcomputer users are more essential than those for the mainframe operations. Mainframe operations are usually managed by a centralized organization, but microcomputer users do not have centralized authority to guide them and therefore need the guidance of manuals much more.

Whether an organization has two microcomputers or two thousand, the policies, procedures, and standards for safeguarding microcomputers and their information must be documented. Those drafting the manual should review the guidelines and adapt them in a form that would be clear and meaningful to the microcomputer users of the organization. One important part of all such manuals should be a summary of the responsibilities of all concerned—the users, their managers, the systems staff, auditors, and security administration. For example, Tompkins County Trust Co., Ithaca NY, developed a microcomputer security manual for reasons similar to those cited in the previous example and has made the manual available for sale.

Training Personnel.

The manufacturing company previously mentioned developed a videotape on microcomputer security and used it in a series of seminars to explain and reinforce the principles in the manuals they had distributed. Their approach was to encourage rather than mandate compliance, and allowed for variations in security procedures. Whatever the approach used, it is worthwhile to conduct some form of regular training for microcomputer users to foster understanding of security needs and practices and to help motivate them to carry out the proper procedures on a continuing basis.

Performing Security Audits

Auditors at the manufacturing company included a review of microcomputer security practices as a routine part of their EDP audit programs. They checked compliance with the principles and standards in the company's manuals and reported deviations and suggested improvements to the management of the areas that used microcomputers. Although flexibility and independence were allowed and encouraged, significant deficiencies were detected and corrected. In situations in which the auditors do not review microcomputer security compliance, systems managers should regularly verify conformity to security principles and practices and initiate corrective action where necessary.

Guarding Against Viruses

Viruses have become of great concern because of their increasing prevalence and capability to do harm. New viruses continually appear and nullify security assurance gained from previous protection measures. Practices to minimize the possibility of virus infections include:[78]

---

[78] National Computer Security Association, *NCSA News 5*(March/April 1992), p. 3.

- Backing up the system regularly. Care should be taken because a recent backup may contain infected files.
- Using software from a known, trusted source. Pirated software should never be used. If shareware is used, it should be obtained from a source as close to the author as possible.
- Using a reliable virus scanner. Many such scanners are available as shareware.
- Scanning all new software before running it.
- Consulting an expert. Vendors of antivirus products generally offer assistance and advice.

The International Computer Security Association (ICSA), Washington DC, offers an interactive virus tutorial, ViruSchool, which explains virus protection techniques. Several other organizations have been formed with the common purpose of conducting research and disseminating information on virus controls. These organizations as well as the ICSA are listed in Exhibit 2.

Antivirus Organizations *ISPNews*, July/August 1992, p. 17.

Studies have indicated that virus detection and prevention software products do not always cover the viruses listed in their advertising. It has been suggested to use two or more such products to help ensure proper coverage and to gain the benefit of differing features of different software. It is essential that new or upgraded antivirus products be acquired and applied frequently to protect against new and modified viruses. Virus protection software should be checked against all existing programs as well as any new programs. Running a virus detection program before making backups ensures that the backups are free of viruses.

Technology Issues Affecting Security

LANs and WANs

Local area networks (LANs) and Wide Area Network (WANs) have an increased vulnerability and consequently an increased need for protective measures. For example, a worker at one network station can access data of another to which that worker is not entitled. A system moved from a mainframe to a network is no longer in a centralized environment. A virus implanted in one microcomputer in a network can infect programs in the other network's stations. (Of the microcomputers with viruses, 71% are said to be in networks.) The benefits of these networks—flexibility and lower costs—can be impaired if suitable security steps are not taken.

Where networks are not under the direct control of the systems manager, the manager should use every means available to influence and coordinate the application of proper security standards and procedures. These means include keeping current records on the networks' hardware, software, and applications; setting standards for and influencing the use of access control procedures and antivirus software; establishing training programs and standards manuals; and gaining the support of auditors to help monitor the established security policies and standards. Of course, the systems manager's coordination of security standards should apply as well as standalone microcomputers that are not connected to networks.

Antivirus Methods Congress
New York University
609 West 114 St.
New York NY 10025
(212) 663-2315

Computer Virus Industry
Association
PO Box 391703
Mountain View CA 94039

Computer Antivirus Research
Organization
Virus Test Center
University of Hamburg
Vogt-Koln-Strasse 30, D-2000
Hamburg 54
Germany

European Institute for Computer
Antivirus Research
c/o S&S International Ltd.
Berkely Ct., Mill St.
Berkhamsted, Hertfordshire HP42HB
United Kingdom

International Computer Society
Association
5435 Connecticut Ave. NW
Suite 33
Washington DC 20015

SOURCE: ISPNews, July/August 1992, p.17.

International Computer Virus
Institute
1257 Siskiyou Blvd.
Suite 179
Ashland OR 97520
(503) 488-3237

National Computer Security
Association U.S.
Anti-Virus Product Developers
Advisory Board
227 West Main St.
Mechanicsburg, PA 17055

National Computer Security
Association Australia
1177 Logan Rd., Unit 4
Holland Par QLD 4121
Australia

National Computer Security
Association Taiwan
6F, 28-1 Li-Shui St.
Taipei, Taiwan, Republic of China

Virus Security Institute
PO Box 908
Margaretville NY 12455

Disk mirroring is a technique for providing a high level of network security. This technique makes use of two disk drive servers and fault-tolerant computing. If one server commits an error, the other takes over, and users do not detect the change in service. The faulty drive is corrected and resumes parallel operation with the other server.

Diskless Workstations

Some organizations have found that diskless workstations connected to networks can provide added security features as well as other benefits. Viruses cannot be introduced directly into diskless workstations nor can unauthorized programs or games. The diskless workstation prevents theft of critical data by eliminating the capability of copying data onto diskettes.

In addition to the security benefits, diskless workstations cost less, are more reliable because they have fewer moving parts, and take up less space on the desk. Applications for which diskless workstations are well suited include airline reservations and operations in such large offices as a state tax department, where several hundred diskless stations can be effective and economical. One disadvantage is that diskless workstations are dependent on the network so that if the network fails, the diskless workstations cannot operate.

Laptop Issues

Laptop computers present additional vulnerabilities. The main danger is the ease with which laptops can be stolen. The hardware and software of each lost laptop can cost a few thousand dollars, but a more serious loss is usually the data. Lost data can require many hours or days to reconstruct. Sometimes the data cannot be reconstructed because the source information no longer exists.

The most effective security measure for laptops is to keep them close by and keep an eye on them. Any important data should be copied onto a disk that is kept at a place separate from the laptop. If there is concern that data might fall into the wrong hands, an encryption program should be used to scramble the information.

Biometric Devices

Several mechanisms have been developed that can give greater assurance that only authorized persons can access a network. These devices include equipment that can record a person's eye features, signature, handwriting, hand- or fingerprints, or speech patterns. They can overcome the major weakness in most access control systems: people often give their IDs and passwords to other people. These biometric systems have not yet become widespread because of their cost, the inconvenience of putting them into practice, and user resistance.

Recommended Course of Action

Most systems development managers have implemented some microcomputer security measures. This article can help these managers determine what further security measures should be instituted to provide proper protection against existing vulnerabilities. It is recommended that these systems development managers:

- Review the steps for securing microcomputers and their information.

- Compare each item under those headings with the existing practices to evaluate what security areas need improvement.

- Plan and revise protective measures to accomplish the required improvements.
- Implement the new and revised protective measures.
- Ensure that security practices are kept up to date.
- Ensure that security policies and practices are audited and monitored to verify that they are properly carried out.

To aid in reviewing existing practices to evaluate the effectiveness of microcomputer security, a systems development manager should ask the following questions:

- Has an inventory been made of the existing microcomputer hardware, software, and applications?
- Has a systematic security evaluation been made of the microcomputer items on that inventory?
- Has senior management been educated about the microcomputer security vulnerabilities and needs of the organization?
- Are updated backup copies of software and data kept for all microcomputer systems?
- Are backup copies maintained off site for important data files?
- Are sufficient measures taken to deter the theft of hardware?
- Are users required to use both passwords and IDs?
- How frequently are passwords revised? Are checks made to see that they are not names or words? That they are not written in viewable places?
- Are callback controls applied for access by phone?
- Is each user restricted to applications and data on a strict need-to-know basis?
- Is encryption used when sensitive data is transmitted over wires?
- Are surge protectors provided for all microcomputers and uninterrupted power for vital microcomputer and network operations?
- Are housekeeping rules enforced regarding food, drink, smoking, dust, and related matters?
- Are written security policies and standards in the hands of all microcomputer and network users? Are they current?
- Is there a training program that covers microcomputer security practices?
- Do auditors review conformity to security policies and rules as a regular part of their audit program?
- Is virus detection and therapy software applied to each microcomputer and network?

- Is new software checked against the virus detection program before it is put into use?
- Are users of networks prevented from applications and data to which they are not entitled?
- Is owner's identification etched on equipment covers?
- Are logs maintained of who uses which application and when?
- Are sensitive files kept from appearing on screen directories?

Bibliography

Forgione, D. and Blankley, A. "Microcomputer Security and Control." *Journal of Accountancy* 85 (June 1990).

Highland, H.J. *Computer Virus Handbook.* Oxford UK: Elsevier Advanced Technology, 1990.

Kane, P. "An Epidemic of Antivirus Groups." *INFOSecurity Product News* 3 (July/August 1992).

Levin, R.B. *The Computer Virus Handbook.* Berkeley CA: Osborne McGraw-Hill, 1990.

Sobol, M. "Callback Security." *Information Systems Security* 1, no. 1(1992).

Author Biographies

Douglas B. Hoyt

Douglas B. Hoyt is a consultant and writer based in Hartsdale NY. He is a board member of the New York Metropolitan Chapter of the Information Systems Security Association, a founding member of the Institute of Management Consultants, and recipient of the Distinguished Service Award from the Association of Systems Management.

Protecting the Portable Computing Environment

Phillip Q. Maier

Today's portable computing environment can take on a variety of forms: from remote connectivity to the home office to remote computing on a standalone microcomputer with desktop capabilities and storage. Both of these portable computing methods have environment-specific threats as well as common threats that require specific protective measures. Remote connectivity can be as simple as standard dial-up access to a host mainframe or as sophisticated as remote node connectivity in which the remote user has all the functions of a workstation locally connected to the organization's local area network (LAN). Remote computing in a standalone mode also presents very specific security concerns, often not realized by most remote computing users.

PORTABLE COMPUTING THREATS

Portable computing is inherently risky. Just the fact that company data or remote access is being used outside the normal physical protections of the office introduces the risk of exposure, loss, theft, or data destruction more readily than if the data or access methods were always used in the office environment.

Data Disclosure

Such simple techniques as observing a user's remote access to the home office (referred to as shoulder surfing) can disclose a company's dial-up access phone number, user account, password, or log-on procedures; this can create a significant threat to any organization that allows remote dial-up access to its networks or systems from off-site. Even if this data or access method isn't disclosed through shoulder surfing, there is still the intermediate threat of data disclosure over the vast amount of remote-site

to central-site communication lines or methods (e.g., the public phone network). Dial-up access is becoming more vulnerable to data disclosure because remote users can now use cellular communications to perform dial-up access from laptop computers.

Also emerging in the remote access arena is a growing number of private metropolitan wireless networks, which present a similar, if not greater, threat of data disclosure. Most private wireless networks don't use any method of encryption during the free-space transmission of a user's remote access to the host computer or transmission of company data. Wireless networks can range in size from a single office space serving a few users to multiple clusters of wireless user groups with wireless transmissions linking them to different buildings. The concern in a wireless data communication link is the threat of unauthorized data interception, especially if the wireless connection is the user's sole method of communication to the organization's computing resources.

All of these remote connectivity methods introduce the threat of data exposure. An even greater concern is the threat of exposing a company's host access controls (i.e., a user's log-on account and static password), which when compromised may go undetected as the unauthorized user accesses a system under a valid user account and password.

Data Loss and Destruction

Security controls must also provide protection against the loss and destruction of data. Such loss can result from user error (e.g., laptop computers may be forgotten in a cab or restaurant) or other cause (e.g., lost baggage). This type of data loss can be devastating, given today's heavy reliance on the portable computer and the large amount of data a portable computer can contain. For this reason alone some security practitioners would prohibit use of portable computers, though increased popularity of portable computing makes this a losing proposition in most organizations.

Other forms of data loss include outright theft of disks, copying of hard disk data, or loss of the entire unit. In today's competitive business world, it is not uncommon to hear of rival businesses or governments using intelligence-gathering techniques to gain an edge over their rivals. More surreptitious methods of theft can take the form of copying a user's diskette from a computer left in a hotel room or at a conference booth during a break. This method is less likely to be noticed, so the data owner or company would probably not take any measures to recover from the theft.

Threats to Data Integrity

Data integrity in a portable computing environment can be affected by direct or indirect threats, such as virus attacks. Direct attacks can occur from an unauthorized user changing data while outside the main facility on

a portable user's system or disk. Data corruption or destruction due to a virus is far more likely in a portable environment because the user is operating outside the physical protection of the office. Any security-conscious organization should already have some form of virus control for on-site computing; however, less control is usually exercised on user-owned computers and laptops. While at a vendor site, the mobile user may use his or her data disk on a customer's computer, which exposes it to the level of virus control implemented by this customer's security measures and which may not be consistent with the user's company's policy.

Other Forms of Data Disclosure

The sharing of computers introduces not only threats of contracting viruses from unprotected computers, but also the distinct possibility of unintended data disclosure. The first instance of shared computer threats is the sharing of a single company-owned portable computer. Most firms don't enjoy the financial luxury of purchasing a portable computer for every employee who needs one. In order to enable widespread use of minimal resources, many companies purchase a limited number of portable computers that can be checked out for use during prolonged stays outside the company. In these cases, users most likely store their data on the hard disk while working on the portable and copy it to a diskette at the end of their use period. But they may not remove it from the hard disk, in which case the portable computer's hard disk becomes a potential source of proprietary information to the next user of the portable computer. And if this computer is lost or misplaced, such information may become public. Methods for protecting against this threat are not difficult to implement; they are discussed in more detail later in this chapter.

Shared company portables can be managed, but an employee's sharing of computers external to the company's control can lead to unauthorized data disclosure. Just as employees may share a single portable computer, an employee may personally own a portable that is also used by family members or it may be lent or even rented to other users. At a minimum, the organization should address these issues as a matter of policy by providing a best practices guideline to employees.

DECIDING TO SUPPORT PORTABLES

As is the case in all security decisions, a risk analysis needs to be performed when making the decision to support portable computers. The primary consideration in the decision to allow portable computing is to determine the type of data to be used by the mobile computing user. A decision matrix can help in this evaluation, as shown in Exhibit 1. The vertical axis of the decision matrix could contain three data types the company uses: confidential, sensitive, and public. Confidential data is competition-sensitive

	CONTROL STRATEGY			
DATA CLASSIFICATION	PORTABLE COMPUTING NOT PERMITTED	PORTABLE COMPUTING WITH STRINGENT SAFEGUARDS	PORTABLE COMPUTING WITH MINIMAL SAFEGUARDS	PORTABLE COMPUTING WITH FEW SAFEGUARDS
Company Confidential	Recommended Action	Not Permitted	Not Permitted	Not Permitted
Company Sensitive	Recommended Action	Recommended Action	Not Permitted	—
Public Data			Recommended Action	Recommended Action

Exhibit 1. Decision Matrix for Supporting Portable Computers

data which cannot be safely disclosed outside the company boundaries. Sensitive data is private, but of less concern if it were disclosed. Public data can be freely disclosed.

The horizontal axis of the matrix could be used to represent decisions regarding whether the data can be used for portable computer use and the level of computing control mechanisms that should be put in place for the type of data involved. (The data classifications in Exhibit 1 are very broad; a given company's may be more granular.) The matrix can be used by users to describe their needs for portable computing, and it can be used to communicate to them what data categories are allowed in a portable computing environment.

This type of decision matrix would indicate at least one data type that should never be allowed for use in a mobile computing environment (i.e., confidential data). This is done because it should be assumed that data used in a portable computing environment will eventually be compromised even with the most stringent controls. With respect to sensitive data, steps should be taken to guard against the potential loss of the data by implementing varying levels of protection mechanisms. There is little concern over use of public data. As noted, the matrix for a specific company may be more complex, specifying more data types unique to the company or possibly more levels of controls or decisions on which data types can and cannot be used.

PROTECTION STRATEGIES

After the decision has been made to allow portable computing with certain use restrictions, the challenge is to establish sound policies and protection strategies against the known threats of this computing environment. The policy and protection strategy may include all the ideas

	DATA DISCLOSURE		DATA LOSS/DESTRUCTION		DATA INTEGRITY	
THREATS	Authentication Disclosure	Transmission Disclosure	Direct Theft	Indirect Theft	Virus	Malicious Tampering
PROTECTIONS	One-Time Passwords	Encryption	Software Controls	Physical Controls	Antivirus Software	Software Access Controls
		Hardware Control	Encryption	Color-Coded Disks	Physical Control Procedures	
			Encryption			

Exhibit 2. Portable Computing Threats and Protection Measures

discussed in this chapter or only a subset, depending on the data type, budget, or resource capabilities.

The basic implementation tool for all security strategies is user education. Implementing a portable computing security strategy is no different; the strategy should call for a sound user education and awareness program for all portable computing users. This program should highlight the threats and vulnerabilities of portable computing and the protection strategies that must be implemented. Exhibit 2 depicts the threats and the potential protection strategies that can be employed to combat them.

User Validation Protection

The protection strategy should reflect the types of portable computing to be supported. If remote access to the company's host computers and networks is part of the portable computing capabilities, then strict attention should be paid to implementing a high-level remote access validation architecture. This may include use of random password generation devices, challenge/response authentication techniques, time-synchronized password generation, and biometric user identification methods. Challenge/response authentication relies on the user carrying some form of token that contains a simple encryption algorithm; the user would be required to enter a personal ID to activate it. Remote access users are registered with a specific device; when accessing the system, they are sent a random challenge number. Users must decrypt this challenge using the token's algorithm and

provide the proper response back to the host system to prove their identity. In this manner, each challenge is different and thus each response is unique. Although this type of validation is keystroke-intensive for users, it is generally more secure than one-time password methods; the PIN is entered only into the remote users' device, and it is not transmitted across the remote link.

Another one-time password method is the time-synchronized password. Remote users are given a token device resembling a calculator that displays an eight-digit numeric password. This device is programmed with an algorithm that changes the password every 60 seconds, with a similar algorithm running at the host computer. Whenever remote users access the central host, they merely provide the current password followed by their personal ID and access is granted. This method minimizes the number of keystrokes that must be entered, but the personal ID is transmitted across the remote link to the host computer, which can create a security exposure.

A third type of high-level validation is biometric identification, such as thumb print scanning on a hardware device at the remote user site, voice verification, and keyboard dynamics, in which the keystroke timing is figured into the algorithm for unique identification. The portable computer user validation from off-site should operate in conjunction with the network security firewall implementation. (A firewall is the logical separation between the company-owned and managed computers and public systems.) Remote users accessing central computing systems are required to cross the firewall after authenticating themselves in the approved manner. Most first-generation firewalls use router-based access control lists (ACLs) as a protection mechanism, but new versions of firewalls may use gateway hosts to provide detailed packet filtering and even authentication.

Data Disclosure Protection

If standalone computers are used in a portable or mobile mode outside of the company facility, consideration should be given to requiring some form of password user identification on the individual unit itself. Various software products can be used to provide workstation-level security.

The minimum requirements should include unique user ID and one-way password encryption so that no cleartext passwords are stored on the unit itself. On company-owned portables, there should be an administrative ID on all systems for central administration as necessary when the units return on-site. This can help ensure that only authorized personnel are using the portable system. Although workstation-based user authentication isn't as strong as host-based user authentication, it does provide a reasonable level of security. At the least, use of a commercial ID and password software products on all portables requires that all users register for access to the portable and the data contained on it.

Other techniques for controlling access to portables include physical security devices on portable computers. Though somewhat cumbersome, these can be quite effective. Physical security locks for portables are a common option. One workstation security software product includes a physical disk lock that inserts into the diskette drive and locks to prevent disk boot-ups that might attempt to override hard-disk-resident software protections.

In addition to user validation issues (either to the host site or the portable system itself), the threat of unauthorized data disclosure must also be addressed. In the remote access arena, the threats are greater because of the various transmission methods used: dial-up over the public switched telephone network, remote network access over such media as the Internet, or even microwave transmission. In all of these cases, the potential for unauthorized interception of transmitted data is real. Documented cases of data capture on the Internet are becoming more common. In the dial-up world, there haven't been as many reported cases of unauthorized data capture, though the threat still exists (e.g., with the use of free-space transmission of data signals over long-haul links).

In nearly all cases, the most comprehensive security mechanism to protect against data disclosure in these environments is full-session transmission encryption or file-level encryption. Simple Data Encryption Standard (DES) encryption programs are available in software applications or as standalone software. Other public domain encryption software such as Pretty Good Privacy (PGP) is available, as are stronger encryption methods using proprietary algorithms. The decision to use encryption depends on the amount of risk of data disclosure the company is willing to accept based on the data types allowed to be processed by portable computer users.

Implementing an encryption strategy doesn't need to be too costly or restrictive. If the primary objective is protection of data during remote transmission, then a strategy mandating encryption of the file before it is transmitted should be put in place. If the objective is to protect the file at all times when it is in a remote environment, file encryption may be considered, though its use may be seen as a burden by users, both because of the processing overhead and the potentially extra manual effort of performing the encryption and decryption for each access. (With some encryption schemes, users may have to decrypt the file before using it and encrypt it again before storing it on the portable computer. More sophisticated applications provide automatic file encryption and decryption, making this step nearly transparent to the user.) Portable computer hardware is also available that can provide complete encryption of all data and processes on a portable computer. The encryption technology is built into the system itself, though this adds to the expense of each unit.

A final point needs to be made on implementing encryption for portable users, and that is the issue of key management. Key management is the coordination of the encryption keys used by users. A site key management scheme must be established and followed to control the distribution and use of the encryption keys.

VIRUS PROTECTION IN A PORTABLE ENVIRONMENT

All portable or off-site computers targeted to process company data must have some consistent form of virus protection. This is a very important consideration when negotiating a site license for virus software. What should be negotiated is not a site license per se, but rather a use license for company's users, wherever they may process company data. The license should include employees' home computers and as well as company-owned portables. If this concept isn't acceptable to a virus software vendor, then procedures must be established in which all data that have left the company and may have been processed on a nonvirus-protected computer must be scanned before it can reenter the company's internal computing environment. This can be facilitated by issuing special color-coded diskettes for storing data that are used on portables or users' home computers. By providing the portable computer users with these disks for storage and transfer of their data and mandating the scanning of these disks and data on a regular basis on-site, the threat of externally contracted computer viruses can be greatly reduced.

CONTROLLING DATA DISSEMINATION

Accumulation of data on portable computers creates the potential for its disclosure. This is easily addressed by implementing a variety of procedures intended to provide checks against this accumulation of data on shared portable computers. A user procedure should be mandated to remove and delete all data files from the hard disk of the portable computer before returning it to the company loan pool. The hardware loaning organization should also be required to check disk contents for user files before reissuing the system.

THEFT PROTECTION

The threat of surreptitious theft can be in the form of illicit copying of files from a user's computer when unattended, such as checked baggage or when left in a hotel room. The simplest method is to never store data on the hard disk and to secure the data on physically secured diskettes. In the case of hotel room storage, it is common for hotels to provide in-room

safes, which can easily secure a supply of diskettes (though take care they aren't forgotten when checking out).

Another method is to never leave the portable in an operational mode when unattended. The batteries and power supply can be removed and locked up separately so that the system itself is not functional and thus information stored on the hard disk is protected from theft. (The battery or power cord could also easily fit in the room safe.) These measures can help protect against the loss of data, which might go unnoticed. (In the event of outright physical theft, the owner can at least institute recovery procedures.) To protect against physical theft, something as simple as a cable ski lock on the unit can be an effective protection mechanism.

USER EDUCATION

The selection of portable computing protection strategies must be clearly communicated to portable computer users by means of a thorough user education process. Education should be mandatory and recurring to assure the most current procedures, tools, and information are provided to portable users. In the area of remote access to on-site company resources, such contact should be initiated when remote users register in the remote access authentication system.

For the use of shared company portable computers, this should be incorporated with the computer check-out process; portable computer use procedures can be distributed when systems are checked out and agreed to by prospective users. With respect to the use of noncompany computers in a portable mode, the best method of accountability is a general user notice that security guidelines apply to this mode of computing. This notification could be referenced in an employee nondisclosure agreement, in which employees are notified of their responsibility to protect company data, on-site or off-site. In addition to registering all portable users, there should be a process to revalidate users in order to maintain their authorized use of portable computing resources on a regular basis. The registration process and procedures should be part of overall user education on the risks of portable computing, protection mechanisms, and user responsibilities for supporting these procedures.

Exhibit 3 provides a sample checklist that should be distributed to all registered users of portables. It should be attached to all of the company's portable computers as a reminder to users of their responsibilities. This sample policy statement includes nearly all the protection mechanisms addressed here, though the company's specific policy may not be as comprehensive depending on the nature of the data or access method used.

- Remove all data from hard disk of company-owned portables before returning them to the loan pool office.
- Leave virus-scanning software enabled on portable computers.
- If it is necessary to use company data on home computers, install and use virus-scanning software.
- Use company-supplied color-coded ("red") disks to store all data used outside the company.
- If no virus-scanning software is available on external computers, virus scan all red disks before using them on company internal computers.
- Physically protect all company computing resources and red disks outside of the facility. (Remember that the value of lost data could exceed that of lost hardware.)
- Be aware of persons watching your work or eavesdropping when you work at off-site locations.
- Report any suspicious activity involving data used in an off-site location. (These might involve data discrepancies, disappearances, or unauthorized modifications.)
- Remote Access (Dial-Up) Guidelines
- If dial-up facilities are to be used, register with the information security office and obtain a random password token to be used for obtaining dial-up access.
- Encrypt all company-sensitive data files before transferring them over dial-up connections in or out of the central facility.
- Report when you no longer require dial-up access and return your password-generating token to the security office.

Exhibit 3. Portable Computing Security Checklist

SUMMARY

The use of portable computing presents very specific data security threats. For every potential threat, some countermeasure should be implemented to ensure the company's proprietary information is protected. This involves identifying the potential threats and implementing the level of protection needed to minimize these threats. By providing a reasonably secure portable computing environment, users can enjoy the benefits of portable computing and the organization can remain competitive in the commercial marketplace.

Operations Security and Controls

Patricia A.P. Fisher

Operations security and controls safeguard information assets while the data is resident in the computer or otherwise directly associated with the computing environment. The controls address both software and hardware as well as such processes as change control and problem management. Physical controls are not included and may be required in addition to operations controls.

Operations security and controls can be considered the heart of information security because they control the way data is accessed and processed. No information security program is complete without a thoroughly considered set of controls designed to promote both adequate and reasonable levels of security. The operations controls should provide consistency across all applications and processes; however, the resulting program should be neither too excessive nor too repressive.

Resource protection, privileged-entity control, and hardware control are critical aspects of the operations controls. To understand this important security area, managers must first understand these three concepts. The following sections give a detailed description of them.

RESOURCE PROTECTION

Resource protection safeguards all of the organization's computing resources from loss or compromise, including main storage, storage media (e.g., tape, disk, and optical devices), communications software and hardware, processing equipment, standalone computers, and printers. The method of protection used should not make working within the organization's computing environment an onerous task, nor should it be so flexible that it cannot adequately control excesses. Ideally, it should obtain a balance between these extremes, as dictated by the organization's specific needs.

This balance depends on two items. One is the value of the data, which may be stated in terms of intrinsic value or monetary value. Intrinsic value is determined by the data's sensitivity — for example, health- and defense-related information have a high intrinsic value. The monetary value is the potential financial or physical losses that would occur should the data be violated.

The second item is the ongoing business need for the data, which is particularly relevant when continuous availability (i.e., round-the-clock processing) is required.

When a choice must be made between structuring communications to produce a user-friendly environment, in which it may be more difficult for the equipment to operate reliably, and ensuring that the equipment is better controlled but not as user friendly (emphasizing availability), control must take precedence. Ease of use serves no purpose if the more basic need for equipment availability is not considered.

Resource protection is designed to help reduce the possibility of damage that might result from unauthorized disclosure and alteration of data by limiting opportunities for misuse. Therefore, both the general user and the technician must meet the same basic standards against which all access to resources is applied.

A more recent aspect of the need for resource protection involves legal requirements to protect data. Laws surrounding the privacy and protection of data are rapidly becoming more restrictive. Increasingly, organizations that do not exercise due care in the handling and maintenance of data are likely to find themselves at risk of litigation. A consistent, well-understood user methodology for the protection of information resources is becoming more important to not only reduce information damage and limit opportunities for misuse but to reduce litigation risks.

Accountability

Access and use must be specific to an individual user at a particular moment in time; it must be possible to track access and use to that individual. Throughout the entire protection process, user access must be appropriately controlled and limited to prevent excess privileges and the opportunity for serious errors. Tracking must always be an important dimension of this control. At the conclusion of the entire cycle, violations occurring during access and data manipulation phases must be reported on a regular basis so that these security problems can be solved.

Activity must be tracked to specific individuals to determine accountability. Responsibility for all actions is an integral part of accountability; holding someone accountable without assigning responsibility is meaningless. Conversely, to assign responsibility without accountability makes it

impossible to enforce responsibility. Therefore, any method for protecting resources requires both responsibility and accountability for all of the parties involved in developing, maintaining, and using processing resources.

An example of providing accountability and responsibility can be found in the way some organizations handle passwords. Users are taught that their passwords are to be stored in a secure location and not disclosed to anyone. In some organizations, first-time violators are reprimanded; if they continue to expose organizational information, however, penalties may be imposed, including dismissal.

Violation Processing

To understand what has actually taken place during a computing session, it is often necessary to have a mechanism that captures the detail surrounding access, particularly accesses occurring outside the bounds of anticipated actions. Any activity beyond those designed into the system and specifically permitted by the generally established rules of the site should be considered a violation.

Capturing activity permits determination of whether a violation has occurred or whether elements of software and hardware implementation were merely omitted, therefore requiring modification. In this regard, tracking and analyzing violations are equally important. Violation tracking is necessary to satisfy the requirements for the due care of information. Without violation tracking, the ability to determine excesses or unauthorized use becomes extremely difficult, if not impossible. For example, a general user might discover that, because of an administrative error, he or she can access system control functions. Adequate, regular tracking highlights such inappropriate privileges before errors can occur.

An all-too-frequently overlooked component of violation processing is analysis. Violation analysis permits an organization to locate and understand specific trouble spots, both in security and usability. Violation analysis can be used to find:

- The types of violations occurring. For example:
  - — Are repetitive mistakes being made? This might be a sign of poor implementation or user training.
  - — Are individuals exceeding their system needs? This might be an indication of weak control implementation.
  - — Do too many people have too many update abilities? This might be a result of inadequate information security design.
- Where the violations are occurring, which might help identify program or design problems.
- Patterns that can provide an early warning of serious intrusions (e.g., hackers or disgruntled employees).

A specialized form of violation examination, intrusion analysis (i.e., attempting to provide analysis of intrusion patterns), is gaining increased attention. As expert systems gain in popularity and ability, their use in analyzing patterns and recognizing potential security violations will grow. The need for such automated methods is based on the fact that intrusions continue to increase rapidly in quantity and intensity and are related directly to the increasing number of personal computers connected to various networks. The need for automated methods is not likely to diminish in the near future, at least not until laws surrounding computer intrusion are much more clearly defined and enforced.

Currently, these laws are not widely enforced because damages and injuries are usually not reported and therefore cannot be proven. Overburdened law enforcement officials are hesitant to actively pursue these violations because they have more pressing cases (e.g., murder and assault). Although usually less damaging from a physical injury point of view, information security violations may be significantly damaging in monetary terms. In several well-publicized cases, financial damage has exceeded $10 million. Not only do violation tracking and analysis assist in proving violations by providing a means for determining user errors and the occasional misuse of data, they also provide assistance in preventing serious crimes from going unnoticed and therefore unchallenged.

Clipping Levels. Organizations usually forgive a particular type, number, or pattern of violations, thus permitting a predetermined number of user errors before gathering this data for analysis. An organization attempting to track all violations, without sophisticated statistical computing ability, would be unable to manage the sheer quantity of such data. To make a violation listing effective, a clipping level must be established.

The clipping level establishes a baseline for violation activities that may be normal user errors. Only after this baseline is exceeded is a violation record produced. This solution is particularly effective for small- to medium-sized installations. Organizations with large-scale computing facilities often track all violations and use statistical routines to cull out the minor infractions (e.g., forgetting a password or mistyping it several times).

If the number of violations being tracked becomes unmanageable, the first step in correcting the problems should be to analyze why the condition has occurred. Do users understand how they are to interact with the computer resource? Are the rules too difficult to follow? Violation tracking and analysis can be valuable tools in assisting an organization to develop thorough but useable controls. Once these are in place and records are produced that accurately reflect serious violations, tracking and analysis become the first line of defense. With this procedure, intrusions are discovered before major damage occurs and sometimes early enough to catch

the perpetrator. In addition, business protection and preservation are strengthened.

Transparency

Controls must be transparent to users within the resource protection schema. This applies to three groups of users. First, all authorized users doing authorized work, whether technical or not, need to feel that computer system protection requirements are reasonably flexible and are not counterproductive. Therefore, the protection process must not require users to perform extra steps; instead, the controls should be built into the computing functions, encapsulating the users' actions and producing the multiple commands expected by the system.

The second group of users consists of authorized users attempting unauthorized work. The resource protection process should capture any attempt to perform unauthorized activity without revealing that it is doing so. At the same time, the process must prevent the unauthorized activity. This type of process deters the user from learning too much about the protective mechanism yet controls permitted activities.

The third type of user consists of unauthorized users attempting unauthorized work. With unauthorized users, it is important to deny access transparently to prevent the intruder from learning anything more about the system than is already known.

User Access Authorities

Resource protection mechanisms may be either manual or automatic. The size of the installation must be evaluated when the security administrator is considering the use of a manual methodology because it can quickly be outgrown, becoming impossible to control and maintain. Automatic mechanisms are typically more costly to implement but may soon recoup their cost in productivity savings.

Regardless of the automation level of a particular mechanism, it is necessary to be able to separate types of access according to user needs. The most effective approach is one of least privilege; that is, users should not be allowed to undertake actions beyond what their specific job responsibilities warrant. With this method, it is useful to divide users into several groups. Each group is then assigned the most restrictive authority available while permitting users to carry out the functions of their jobs.

There are several options to which users may be assigned. The most restrictive authority and the one to which most users should be assigned is read only. Users assigned to read only are allowed to view data but are not allowed to add, delete, or make changes.

The next level is read/write access, which allows users to add or modify data within applications for which they have authority. This level permits individuals to access a particular application and read, add, and write over data in files copied from the original location.

A third access level is change. This option permits the holder not only to read a file and write data to another file location but to change the original data, thereby altering it permanently.

When analyzing user access authorities, the security practitioner must distinguish between access to discretionary information resources (which is regulated only by personal judgment) and access to nondiscretionary resources (which is strictly regulated on the basis of the predetermined transaction methodology). Discretionary user access is defined as the ability to manipulate data by using custom-developed programs or a general-purpose utility program. The only information logged for discretionary access in an information security control mechanism is the type of data accessed and at what level of authority. It is not possible to identify specific uses of the data.

Nondiscretionary user access, on the other hand, is performed while executing specific business transactions that affect information in a predefined way. For this type of access, users can perform only certain functions in carefully structured ways. For example, in a large accounting system, many people prepare transactions that affect the ledger. Typically, one group of accounting analysts is able to enter the original source data but not to review or access the overall results. Another group has access to the data for review but is not able to alter the results. In addition, with nondiscretionary access, the broad privileges assigned to a user for working with the system itself should be analyzed in conjunction with the user's existing authority to execute the specific transactions needed for the current job assignment. This type of access is important when a user can be authorized to both read and add information but not to delete or change it. For example, bank tellers need access to customer account information to add deposits but do not need the ability to change any existing information.

At times, even nondiscretionary access may not provide sufficient control. In such situations, special access controls can be invoked. Additional restrictions may be implemented in various combinations of add, change, delete, and read capabilities. The control and auditability requirements that have been designed into each application are used to control the management of the information assets involved in the process.

Special Classifications. A growing trend is to give users access to only resource subsets or perhaps to give them the ability to update information only when performing a specific task and following a specific procedure.

This has created the need for a different type of access control in which authorization can be granted on the basis of both the individual requesting resource access and the intended use of that resource. This type of control can be exercised by the base access control mechanism (i.e., the authorization list, including user ID and program combinations).

Another method sometimes used provides the required access authority along with the programs the user has authorization for; this information is provided only after the individual's authority has been verified by an authorization program. This program may incorporate additional constraints (e.g., scoped access control) and may include thorough access logging along with ensuring data integrity when updating information.

Scoped access control is necessary when users need access only to selected areas or records within a resource, thereby controlling the access granted to a small group on the basis of an established method for separating that group from the rest of the data. In general, the base access control mechanism is activated at the time of resource initialization (i.e., when a data set is prepared for access). Therefore, scoped access control should be provided by the data base management system or the application program. For example, in personnel systems, managers are given authority to access only the information related to their employees.

PRIVILEGED-ENTITY CONTROL

Levels of privileges provide users with the ability to invoke the commands needed to accomplish their work. Every user has some degree of privilege. The term, however, has come to be applied more to those individuals performing specialized tasks that require broad capabilities than to the general user. In this context, a privilege provides the authority necessary to modify control functions (e.g., access control, logging, and violation detection) or may provide access to specific system vulnerabilities. (Vulnerabilities are elements of the system's software or hardware that can be used to gain unauthorized access to system facilities or data.) Thus, individuals in such positions as systems programming, operations, and systems monitoring are authorized to do more than general users.

A privilege can be global when it is applicable to the entire system, function-oriented when it is restricted to resources grouped according to a specific criterion, or application specific when it is implemented within a particular piece of application code. It should be noted that when an access control mechanism is compromised, lower-level controls may also be compromised. If the system itself is compromised, all resources are exposed regardless of any lower-level controls that may be implemented.

Indirect authorization is a special type of privilege by which access granted for one resource may give control over another privilege. For example, a user with indirect privileges may obtain authority to modify the

Class	Job Assignment	Class Access Privileges
A	General User	A
B	Programmer	B, A
C	Manager	C, A (sometimes B)
D	Security Administrator	D, B, A
E	Operator	E, D, B, A
F	System Programmer	F, E, D, B, A
G	Auditor	G, B, A

Exhibit 1. Sample Privileged-Entity Access

password of a privileged user (e.g., the security administrator). In this case, the user does not have direct privileges but obtains them by signing on to the system as the privileged user (although this would be a misuse of the system). The activities of anyone with indirect privileges should be regularly monitored for abuse.

Extended or special access to computing resources is termed privileged-entity access. Extended access can be divided into various segments, called classes, with each succeeding class more powerful than those preceding it. The class into which general system users are grouped is the lowest, most restrictive class; a class that permits someone to change the computing operating system is the least restrictive, or most powerful. All other system support functions fall somewhere between these two.

Users must be specifically assigned to a class; users within one class should not be able to complete functions assigned to users in other classes. This can be accomplished by specifically defining class designations according to job functions and not permitting access ability to any lower classes except those specifically needed (e.g., all users need general user access to log on to the system). An example of this arrangement is shown in Exhibit 1.

System users should be assigned to a class on the basis of their job functions; staff members with similar computing access needs are grouped together with a class. One of the most typical problems uncovered by information security audits relates to the implementation of system assignments. Often, sites permit class members to access all lesser functions (i.e., toward A in Exhibit 1). Although it is much simpler to implement this plan than to assign access strictly according to need, such a plan provides little control over assets.

The more extensive the system privileges given within a class, the greater the need for control and monitoring to ensure that abuses do not occur. One method for providing control is to install an access control mechanism, which may be purchased from a vendor (e.g., RACF, CA-TOP,

SECRET, and CA-ACF2) or customized by the specific site or application group. To support an access control mechanism, the computer software provides a system control program. This program maintains control over several aspects of computer processing, including allowing use of the hardware, enforcing data storage conventions, and regulating the use of I/O devices.

The misuse of system control program privileges may give a user full control over the system, because altering control information or functions may allow any control mechanism to be compromised. Users who abuse these privileges can prevent the recording of their own unauthorized activities, erase any record of their previous activities from the audit log, and achieve uncontrolled access to system resources. Furthermore, they may insert a special code into the system control program that can allow them to become privileged at any time in the future.

The following sections discuss the way the system control program provides control over computer processing.

Restricting Hardware Instructions. The system control program can restrict the execution of certain computing functions, permitting them only when the processor is in a particular functional state (known as privileged or supervisor state) or when authorized by architecturally defined tables in control storage. Programs operate in various states, during which different commands are permitted. To be authorized to execute privileged hardware instructions, a program should be running in a restrictive state that allows these commands.

Instructions permitting changes in the program state are classified as privileged and are available only to the operating system and its extensions. Therefore, to ensure adequate protection of the system, only carefully selected individuals should be able to change the program state and execute these commands.

Controlling Main Storage. The use of address translation mechanisms can provide effective isolation between different users' storage locations. In addition, main storage protection mechanisms protect main storage control blocks against unauthorized access. One type of mechanism involves assignment of storage protection keys to portions of main storage to keep unauthorized users out.

The system control program can provide each user section of the system with a specific storage key to protect against read-only or update access. In this methodology, the system control program assigns a key to each task and manages all requests to change that key. To obtain access to a particular location in storage, the requesting routine must have an identical key or the master key.

Constraining I/O Operations. If desired, I/O instructions may be defined as privileged and issued only by the system control program after access authority has been verified. In this protection method, before the initiation of any I/O operations, a user's program must notify the system control program of both the specific data and the type of process requested. The system control program then obtains information about the data set location, boundaries, and characteristics that it uses to confirm authorization to execute the I/O instruction.

The system control program controls the operation of user programs and isolates storage control blocks to protect them from access or alteration by an unauthorized program. Authorization mechanisms for programs using restricted system functions should not be confused with the mechanisms invoked when a general user requests a computing function. In fact, almost every system function (e.g., the user of any I/O device, including a display station or printer) implies the execution of some privileged system functions that do not require an authorized user.

Privilege Definition

All levels of system privileges must be defined to the operating system when hardware is installed, brought online, and made available to the user community. As the operating system is implemented, each user ID, along with an associated level of system privileges, is assigned to a predefined class within the operating system. Each class is associated with a maximum level of activity.

For example, operators are assigned to the class that has been assigned those functions that must be performed by operations personnel. Likewise, systems auditors are assigned to a class reserved for audit functions. Auditors should be permitted to perform only those tasks that both general users and auditors are authorized to perform, not those permitted for operators. By following this technique, the operating system may be partitioned to provide no more access than is absolutely necessary for each class of user.

Particular attention must be given to password management privileges. Some administrators must have the ability and therefore the authorization to change another user's password, and this activity should always be properly logged. The display password feature, which permits all passwords to be seen by the password administrator, should be disabled or blocked. If not disabled, this feature can adversely affect accountability, because it allows some users to see other users' passwords.

Privilege Control and Recertification

Privileged-entity access must be carefully controlled, because the user IDs associated with some system levels are very powerful and can be used

inappropriately, causing damage to information stored within the computing resource. As with any other group of users, privileged users must be subject to periodic recertification to maintain the broad level of privileges that have been assigned to them. The basis for recertification should be substantiation of a continued need for the ID. Need, in this case, should be no greater than the regular, assigned duties of the support person and should never be allocated on the basis of organizational politics or backup.

A recertification process should be conducted on a regular basis, at least semi-annually, with the line management verifying each individual's need to retain privileges. The agreement should be formalized yet not bureaucratic, perhaps accomplished by initialing and dating a list of those IDs that are to be recertified. By structuring the recertification process to include authorization by managers of personnel empowered with the privileges, a natural separation of duties occurs. This separation is extremely important to ensure adequate control. By separating duties, overallocation of system privileges is minimized.

For example, a system programmer cannot receive auditor privileges unless the manager believes this function is required within the duties of the particular job. On the other hand, if a special project requires a temporary change in system privileges, the manager can institute such a change for the term of the project. These privileges can then be canceled after the project has been completed.

Emergency Procedures. Privileged-entity access is often granted to more personnel than is necessary to ensure that theoretical emergency situations are covered. This should be avoided and another process employed during emergencies — for example, an automated process in which support personnel can actually assign themselves increased levels of privileges. In such instances, an audit record is produced, which calls attention to the fact that new privileges have been assigned. Management can then decide after the emergency whether it is appropriate to revoke the assignment. However, management must be notified so the support person's subsequent actions can be tracked.

A much more basic emergency procedure might involve leaving a privileged ID password in a sealed envelope with the site security staff. When the password is needed, the employee must sign out the envelope, which establishes ownership of the expanded privileges and alerts management. Although this may be the least preferred method of control, it alerts management that someone has the ability to access powerful functions. Audit records can then be examined for details of what that ID has accessed. Although misuse of various privileged functions cannot be prevented with this technique, reasonable control can be accomplished without eliminating the ability to continue performing business functions in an efficient manner.

Activity Reporting. All activity connected with privileged IDs should be reported on logging audit records. These records should be reviewed periodically to ensure that privileged IDs are not being misused. Either a sample of the audit records should be reviewed using a predetermined methodology incorporating approved EDP auditing and review techniques or all accesses should be reviewed using expert system applications. Transactions that deviate from those normally conducted should be examined and, if necessary, fully investigated.

Under no circumstances should management skip the regular review of these activities. Many organizations have found that a regular review process deters curiosity and even mischief within the site and often produces the first evidence of attempted hacking by outsiders.

CHANGE MANAGEMENT CONTROLS

Additional control over activities by personnel using privileged access IDs can be provided by administrative techniques. For example, the most easily sidestepped control is change control. Therefore, every computing facility should have a policy regarding changes to operating systems, computing equipment, networks, environmental facilities (e.g., air-conditioning, water, heat, plumbing, electricity, and alarms), and applications. A policy is necessary if change is to be not only effective but orderly, because the purpose of the change control process is to manage changes to the computing environment.

The goals of the management process are to eliminate problems and errors and to ensure that the entire environment is stable. To achieve these goals, it is important to:

- *Ensure orderly change.* In a facility that requires a high level of systems availability, all changes must be managed in a process that can control any variables that may affect the environment. Because change can be a serious disruption, however, it must be carefully and consistently controlled.
- *Inform the computing community of the change.* Changes assumed to affect only a small subsection of a site or group may in fact affect a much broader cross-section of the computing community. Therefore, the entire computing community should receive adequate notification of impending changes. It is helpful to create a committee representing a broad cross-section of the user group to review proposed changes and their potential effect on users.
- *Analyze changes.* The presentation of an intended change to an oversight committee, with the corresponding documentation of the change, often effectively exposes the change to careful scrutiny. This analysis clarifies the originator's intent before the change is implemented and is helpful

in preventing erroneous or inadequately considered changes from entering the system.

- *Reduce the impact of changes on service.* Computing resources must be available when the organization needs them. Poor judgment, erroneous changes, and inadequate preparation must not be allowed in the change process. A well-structured change management process prevents problems and keeps computing services running smoothly.

General procedures should be in place to support the change control policy. These procedures must, at the least, include steps for instituting a major change to the site's physical facility or to any major elements of the system's software or hardware. The following steps should be included:

1. *Applying to introduce a change.* A method must be established for applying to introduce a change that will affect the computing environment in areas covered by the change control policy. Change control requests must be presented to the individual who will manage the change through all of its subsequent steps.
2. *Cataloging the change.* The change request should be entered into a change log, which provides documentation for the change itself (e.g., the timing and testing of the change). This log should be updated as the change moves through the process, providing a thorough audit trail of all changes.
3. *Scheduling the change.* After thorough preparation and testing by the sponsor, the change should be scheduled for review by a change control committee and for implementation. The implementation date should be set far enough in advance to provide the committee with sufficient review time. At the meeting with the change control committee, all known ramifications of the change should be discussed. If the committee members agree that the change has been thoroughly tested, it should be entered on the implementation schedule and noted as approved. All approvals and denials should be in writing, with appropriate reasons given for denials.
4. *Implementing the change.* The final step in the change process is application of the change to the hardware and software environment. If the change works correctly, this should be noted on the change control form. When the change does not perform as expected, the corresponding information should be gathered, analyzed, and entered on the change control form, as a reference to help avoid a recurrence of the same problem in the future.
5. *Reporting changes to management.* Periodically, a full report summarizing change activity should be submitted to management. This helps ensure that management is aware of any quality problems that may have developed and enables management to address any service problems.

These steps should be documented and made known to all involved in the change process. Once a change process has been established, someone must be assigned the responsibility for managing all changes throughout the process.

HARDWARE CONTROL

Security and control issues often revolve around software and physical needs. In addition, the hardware itself can have security vulnerabilities and exposures that need to be controlled. The hardware access control mechanism is supported by operating system software. However, hardware capabilities can be used to obtain access to system resources. Software-based control mechanisms, including audit trail maintenance, are ineffective against hardware-related access. Manual control procedures should be implemented to ensure that any hardware vulnerability is adequately protected.

When the system control program is initialized, the installation personnel select the desired operating system and other software code. However, by selecting a different operating system or merely a different setup of the operating system (i.e., changing the way the hardware mechanisms are used), software access control mechanisms can be defeated.

Some equipment provides hardware maintenance functions that allow main storage display and modification in addition to the ability to trace all program instructions while the system is running. These capabilities enable someone to update system control block information and obtain system privileges for use in compromising information. Although it is possible to access business information directly from main storage, the information may be encrypted. It is simpler to obtain privileges and run programs that can turn encrypted data into understandable information.

Another hardware-related exposure is the unauthorized connection of a device or communications line to a processor that can access information without interfacing with the required controls. Hardware manufacturers often maintain information on their hardware's vulnerabilities and exposures. Discussions with specific vendors should provide data that will help control these vulnerabilities.

Problem Management

Although problem management can affect different areas within computer services, it is most often encountered in dealing with hardware. This control process reports, tracks, and resolves problems affecting computer services. Management should be structured to measure the number and types of problems against predetermined service levels for the area in which the problem occurs. This area of management has three major objectives:

1. Reducing failures to an acceptable level.
2. Preventing recurrences of problems.
3. Reducing impact on service.

Problems can be organized according to the types of problems that occur, enabling management to better focus on and control problems and thereby providing more meaningful measurement. Examples of the problem types include:

- Performance and availability.
- Hardware.
- Software.
- Environment (e.g., air-conditioning, plumbing, and heating).
- Procedures and operations (e.g., manual transactions).
- Network.
- Safety and security.

All functions in the organization that are affected by these problems should be included in the control process (e.g., operations, system planning, network control, and systems programming).

Problem management should investigate any deviations from standards, unusual or unexplained occurrences, unscheduled initial program loads, or other abnormal conditions. Each is examined in the following sections.

Deviations from Standards. Every organization should have standards against which computing service levels are measured. These may be as simple as the number of hours a specific CPU is available during a fixed period of time. Any problem that affects the availability of this CPU should be quantified into time and deducted from the available service time. The resulting total provides a new, lower service level. This can be compared with the desired service level to determine the deviation.

Unusual or Unexplained Occurrences. Occasionally, problems cannot be readily understood or explained. They may be sporadic or appear to be random; whatever the specifics, they must be investigated and carefully analyzed for clues to their source. In addition, they must be quantified and grouped, even if in an Unexplained category. Frequently, these types of problems recur over a period of time or in similar circumstances, and patterns begin to develop that eventually lead to solutions.

Unscheduled Initial Program Loads. The primary reason a site undergoes an unscheduled initial program load (IPL) is that a problem has occurred. Some portion of the hardware may be malfunctioning and therefore slowing down, or software may be in an error condition from which it cannot recover. Whatever the reason, an occasional system queue must be

cleared, hardware and software cleansed and an IPL undertaken. This should be reported in the problem management system and tracked.

Other Abnormal Conditions. In addition to the preceding problems, such events as performance degradation, intermittent or unusual software failures, and incorrect systems software problems may occur. All should be tracked.

Problem Resolution

Problems should always be categorized and ranked in terms of their severity. This enables responsible personnel to concentrate their energies on solving those problems that are considered most severe, leaving those of lesser importance for a more convenient time.

When a problem can be solved, a test may be conducted to confirm problem resolution. Often, however, problems cannot be easily solved or tested. In these instances, a more subjective approach may be appropriate. For example, management may decide that if the problem does not recur within a predetermined number of days, the problem can be considered closed. Another way to close such problems is to reach a major milestone (e.g., completing the organization's year-end processing) without a recurrence of the problem.

SUMMARY

Operations security and control is an extremely important aspect of an organization's total information security program. The security program must continuously protect the organization's information resources within data center constraints. However, information security is only one aspect of the organization's overall functions. Therefore, it is imperative that control remain in balance with the organization's business, allowing the business to function as productively as possible. This balance is attained by focusing on the various aspects that make information security not only effective but as simple and transparent as possible.

Some elements of the security program are basic requirements. For example, general controls must be formulated, types of system use must be tracked, and violations must be tracked in any system. In addition, use of adequate control processes for manual procedures must be in place and monitored to ensure that availability and security needs are met for software, hardware, and personnel. Most important, whether the organization is designing and installing a new program or controlling an ongoing system, information security must always remain an integral part of the business and be addressed as such, thus affording an adequate and reasonable level of control based on the needs of the business.

DATA CENTER SECURITY: USEFUL INTRANET SECURITY METHODS AND TOOLS

John R. Vacca

INSIDE

INTRODUCTION

Information technology (IT) is now used universally to support critical enterprise business decisions. It has evolved beyond basic applications such as billing and inventory, to the point where it directly supports customers and the manufacturing process. This sophisticated enterprise business information technology is entirely dependent on the diverse (and often incompatible) operating systems and hardware data center environments needed for their execution. All these systems must be managed and maintained if they are to continue to provide support for the ever-increasing applications on which the enterprise's data center depends. Information technology has a significant impact on the effectiveness of the intranet as a whole. Consequently, competitive performance of the enterprise is now directly affected by the management and control of data center computer resources.

The measure of IT management's success or failure to support the enterprise is based on its ability to establish and meet required levels of performance, reliability, and availability — while staying within budgetary constraints.

PAYOFF IDEA

This article provides IT managers with a set of data center management and intranet security software tools and methods. The information presented in this article will enable IT management to better meet the challenges inherent in managing data center services, costs, and security as the use of distributed systems becomes evermore critical to the enterprise.

The management and control of complex intranets and data centers are, however, daunting challenges to be met by IT professionals in the next decade. Some key areas that must be achieved are to:

- establish and consistently meet service-level agreements with end users
- control costs to meet service levels at the lowest possible level of investment
- protect the wealth of enterprise information and key resources that often span multiple operating systems and hardware platforms

In addition, IT management must not only achieve and maintain all these goals, but it must do this while ensuring complete system integrity at all times. An increasingly large role in enterprise computing is being played by intranet and client/server configurations of midrange and desktop computing environments, and open systems such as UNIX-based data center environments. Downsizing and decentralizing of processing resources is a result of the evolution toward a more global view — one that is replacing the traditional mainframe view of IT management.

In other words, IT management recognizes more and more that it needs both diverse and complementary information processing technologies if it is to meet the needs of the enterprise. Consequently, there has been substantial growth in complex heterogeneous systems that span multiple computing platforms. This leaves systems and data center intranet security management to contemplate some new concerns: it must now be recognized that midrange and desktop computing environments collectively represent a significant investment in information processing power. It must bring to each computing platform the standard systems management functionality that has been required on large mainframe systems: the need for the same level of automation, resource management, intranet security, and data integrity. Data center intranet security management is essential today for distributed systems, and this provides IT management with new challenges.

INTRANET SECURITY SOFTWARE AND SYSTEMS MANAGEMENT

In many ways, the challenges of distributed systems and data center intranet security management are the same as those of distributed applications. End-user applications, such as manufacturing and general ledger systems, are built on known database structures and application objectives. Distributed systems and data center intranet security management solutions, however, must address a very diverse and often perplexing variety of environments.

There are vast differences in the various operating systems of each platform, and management for distributed systems and data center intra-

net security must adjust accordingly. In order to present a unified whole, the goal is to insulate the administrator from the specific vagaries of each system.

Having the software solutions needed to provide comprehensive systems and data center intranet security management on each platform in the intranet is a significant part of mastering distributed systems and intranet security management. These solutions alone are only effective, however, if they can be tied together into a single point of management. Systems and intranet security administrators must be able to manage all or any desired part of the enterprise from any location within the intranet. Single point of management allows administrators to implement and enforce overall enterprise policies while continuing to provide the local controls necessary in a constantly changing environment to ensure responsiveness.

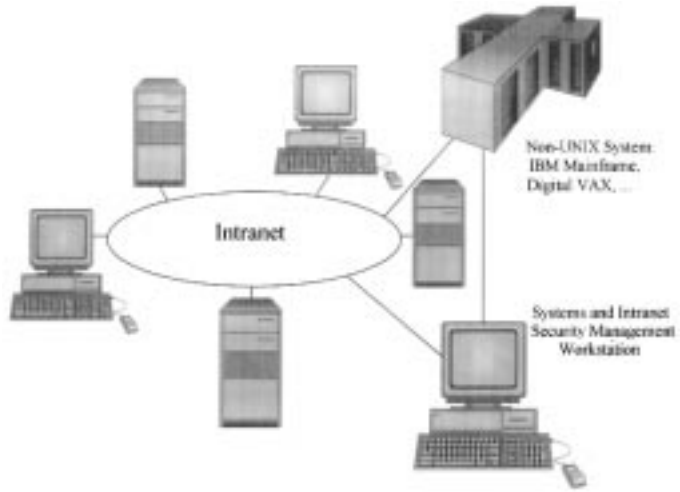
IT management must be able to provide the consistent, high service levels that are required by enterprises. Systems management and intranet security software should deliver integrated, total data center and intranet automation capabilities. At the same time, it should make possible the controlling of costs and ensuring protection of valuable data and resources. For example, enterprises provide distributed systems and intranet security management across multiple platforms. They also provide the ability to endorse and extend industry standards. This allows data center management software to be compliant with, fully support, and extend the capabilities of the Open Software Foundation's (OSF) initiatives for DCE (distributed computing environment) and DME (distributed management environment).

Systems and data center intranet security management software should provide a single point of management for complex heterogeneous systems. For example, data center management software should provide a single point of management through three strategic elements.

1. It meets the specific needs of the platform and exploits the special characteristics and benefits of each by providing robust solutions that are appropriate to each platform.
2. It has a flexible manager–agent architecture that enables each solution to work cooperatively with other solutions in the intranet; managing the flow of work; or performing work on behalf of other solutions in the intranet.
3. It provides a flexible user interface that enables each and every solution within the data center environment to be managed from a single location or multiple locations if desired.

This third element is known as the systems and data center intranet security management workstation, as shown in Exhibit 1, it provides sys-

EXHIBIT 1 — The Systems Management and Intranet Security Workstation



tems and data center intranet security management for a homogeneous or heterogeneous intranet.

INTRANET SECURITY AND SYSTEMS MANAGEMENT WORKSTATION

When managing all of the solutions in the data center environment, a systems and intranet security management workstation should provide the systems and intranet security administrator with a GUI (graphical user interface)-based user interface. Such a workstation could be used to administer a single UNIX system, a remote IBM mainframe, an OS/2 or Novell-based LAN, or even a heterogeneous intranet containing all of these components and others such as the IBM AS/400 and Digital VAX/VMS. The systems and intranet security management workstation should operate on a lower cost X-terminal.

For example, in an enterprise's systems management workstation, the modern interface reduces the complexity of systems and intranet security management. It also provides an intuitive and logical view of otherwise complex issues. The GUI adjusts for the particular aspects of each environment, where necessary. For example, the user interface is identical when managing a multi-node job scheduler for defining jobs within job sets; or, schedules and their relationships to one another. However, when opening up a job detail window for actually setting up individual jobs, the GUI would present a job from an IBM MVS system as a collection of JCL (Job Control Language) statements. A job on UNIX would display as

a shell script, while a job on Digital VAX/VMS would contain Digital Control Language (DCL) in place of JCL or shell statements.

Control and flexibility of this type is essential for distributed systems and data center intranet security management. Being able to manage each platform from a common location with a common interface, and preserving the concepts and terminology across the intranet, are extremely useful — along with having the tools needed on each platform.

Investments in the training of systems and data center intranet security administration personnel can be leveraged into new platforms by maintaining the model for each solution across the system. For example, users familiar with ACF2 or TOP SECRET security systems on IBM MVS would find the workload management and intranet security provided through UNIX familiar and easily understood.

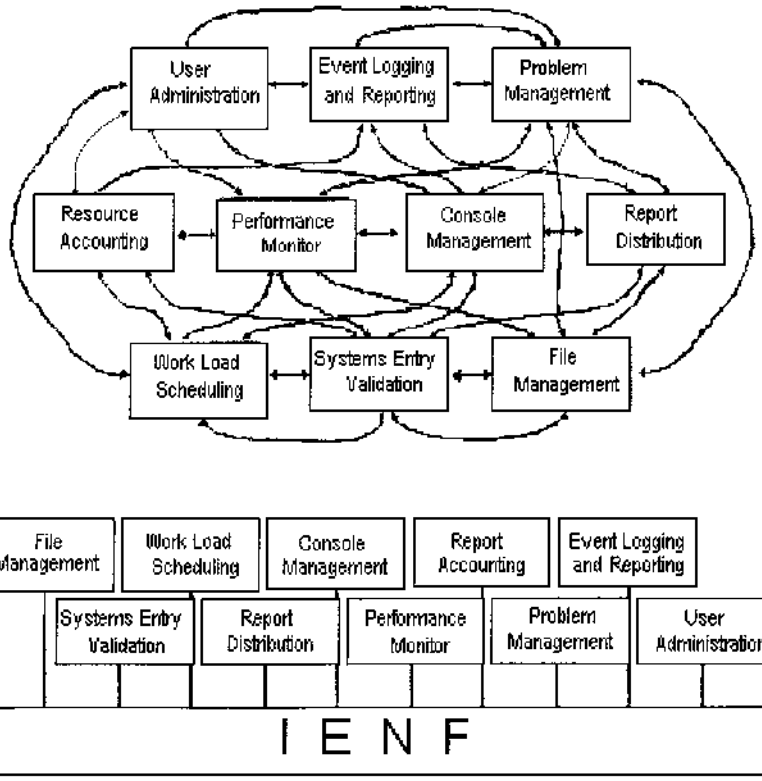
THE CONNECTIVITY FACTOR

An enterprise's systems and intranet security management software should cover a broad range of interrelated functions required to manage and control data center and intranet activity. In addition, it should utilize service layers to interact with each other, adding value to the intranet's information technology systems as a whole.

Furthermore, an enterprise's blueprint for a software architecture and its underlying guiding principles should try to provide a comprehensive strategy for software development for the IT community. This approach, where all components work together across multiple platforms, is essential in maintaining sufficient responsiveness to the continually evolving priorities of enterprise-driven information processing requirements.

Automating each area of systems and data center intranet security management does not in itself result in a complete and successful approach to systems management. Each component of the systems and intranet security management solution must support and communicate with every other component in order to achieve total data center and intranet automation. For example, intranet-wide information security cannot be ensured if each systems management solution uses its own security tables. Problem and change management cannot keep pace with the dynamic activities of large and complex data center environments if software that addresses functions such as scheduling, report distribution, storage management, and security cannot automatically open and update problem incidents. Clearly, global workload management is unattainable unless each platform provides workload scheduling and resource balancing capabilities that interface with all other platforms. Therefore, complete and effective integration requires both a design for integration and an architecture that support development and enhancement of the completely integrated solution. As shown in Exhibit 2 for an IENF, an enterprise's services enable the integration of systems and data center intranet

EXHIBIT 2 — An Integration Event Notification Facility (IENF)



security management functions without an uncontrollable and unmanageable explosion of interfaces.

POWERFUL SOLUTIONS

The rule-based policies that govern data center procedures are particularly well-suited to automation and can be handled by systems and intranet security management software. This automation capability is essential in addressing the complex activities of multi-vendor and multi-operating system intranet environments where a variety of procedures are generally followed due to the variation in capabilities provided by the native platforms. An enterprise's systems management and intranet security software should simplify the management of these complex data center environments by extending native platform capabilities wherever necessary to ensure operational consistency, regardless of platform.

EXHIBIT 3 — Systems and Intranet Security Management

The otherwise numerous and complex procedures are significantly reduced, enabling operations staff to become familiar with all aspects of a streamlined, unified system. This dramatically reduces the training required of operational staff, particularly in complicated subsystem procedures, and eliminates the need for multiple subsystem specialists to closely monitor data center and intranet activities. An enterprise's systems management and intranet security software should provide a robust, fully integrated, distributed solution that covers the essential disciplines of automated production control, automated storage management, performance management and accounting, data center administration, and security, control, and audit (see Exhibit 3).

To completely automate information technology systems and derive maximum benefit, all of these components must be present for all platforms. Vendors that offer only a part of this functionality and limit the functionality to specific platforms cannot effectively assist IT management in meeting its service-level objectives.

AUTOMATED PRODUCTION CONTROL SOLUTIONS

A data center should provide an integrated set of solutions for automated production control. These solutions cover all areas of functionality, including workload management; rerun, restart, and recovery; console management; report distribution; control language validation; report balancing; and production documentation.

Automated Workload Management

Workload management is concerned with complete automated management of production workloads, including workload balancing, automatic submission and tracking of work based on user-defined scheduling criteria, priority, and system resource availability. This ensures that work is completed correctly and that critical deadlines are met.

Automated Rerun, Restart, and Recovery

Rerun, restart, and recovery automates the often complex, manual-intensive, and error-prone rerun and recovery process, thus enabling processing to restart at the optimum recovery point. In addition, it automatically handles the otherwise time-consuming manual procedures such as job setup, data set recovery, and backout.

Automated Console Management

Console management improves operating efficiency and reduces errors by automating the handling of console messages. It provides an advanced message/action capability that can alter, suppress, or reply to messages or initiate other actions, such as automatically issuing IPL/IMLs, alerts (through voice and pager notification capabilities), commands or invoking programs based on the content, and frequency and other characteristics of the message traffic. In addition, selective action based on specific console and terminal IDs allows the assignment of consoles to specialized applications, such as intranet monitoring, system monitoring, or tape mount processing. A simulation capability is also provided to assist in the development and verification of these event/action criteria.

When used in conjunction with a programmable workstation, this software provides a single focal point for all console operation activities in a multi-CPU, multi-operating system, and intranet environment. In addition, remote access to perform console management is available through remote dial-ups (PC, remote TSO, CICS, session) or through the telephone using the latest voice and touch-tone technology.

Automated Report Distribution Function

Report distribution provides extensive capabilities for the flexible and efficient production, tracking, and distribution of reports. This results in speeding the delivery time, increasing the accuracy, and improving the tracking of reports. Facilities are provided that automatically identify pages from existing reports, place them into bundles, and sort them by delivery location prior to actual printing.

These capabilities provide end users with the information they want, when and where they need it, while reducing or eliminating redundant information and the materials and efforts that are wasted in its distribu-

tion. Automated report distribution software also provides online viewing capabilities that can reduce the need for a hard copy of reports as well as the option to select all or parts of reports for printing. Report archiving capabilities enable the storing of reports offline for auditing purposes and for future viewing or reprinting.

A data center's report management software on intelligent workstations should extend report management capabilities by enabling end users to receive reports as files on their local computer. The ability to merge, annotate, or change reports using a familiar computer should be automated by the workstation-based software, and redistribution of these new reports should be provided through interaction with the software on the host system.

Advanced Control Language Validation

Virtually all systems in use today provide an interpretive control language for defining the execution of batch and online processing. Examples of these languages include IBM Job Control Language (JCL) and the UNIX Shell Script language. A data center's design for systems and intranet security management should include complete advanced control language validation capabilities that reduce or eliminate errors that can cause failures during production execution, and that aid the end user in diagnosing problems with control language programs. In addition, this software should enforce site-specific standards while providing the reports and cross-reference information that are needed for future maintenance.

Automated Report and File Balancing

Report balancing consists of extensive, automated report and file balancing capabilities that enable a quality level to be achieved that is unavailable through manual efforts. In addition to automatically ensuring the accuracy of reports after they are printed, this software can uniquely catch errors during the execution of production work cycles (both enabling fast and accurate resolution of problems), and prevent the completion of in-error production runs and the distribution of incorrect report output.

Production Control Documentation

Production documentation provides complete and consistent centralized online documentation system for the production control environment. Integration with other production control software enables documentation efforts to be automated and centralized, ensuring accessible and accurate information essential to data center operations, and particularly for contingency planning, disaster recovery, and future maintenance.

AUTOMATED STORAGE MANAGEMENT SOFTWARE

Automated storage management (ASM) software significantly extends the native operating system's capabilities of storage and resource management. This software optimizes performance and access to information. It ensures availability, integrity, and reliability — regardless of the various media device types and differing configurations of mainframes, midrange computers, PCs, and LANs that define the IT processing environment.

Backup Management

Backup management provides the ability to back up files based on creation date and version, as well as supporting backups of multiple versions of the same file. It keeps track of which volume each file has been backed up to.

It also enables users, system, and intranet security administrators to view the media (tape versus disk) and version of each backed-up file — and easily initiate a restore if needed. Backup management eliminates the problem of keeping track of where backed-up files are kept, and how to find them when a restore is needed.

Archive Management

Archive management makes sure that files have been removed from the online disk system and stored on other media that are based on storage management policies and are available when needed. It ensures that enough storage is always available on the file system to keep users working.

Archive Transparent Restore Processing

Through an automated storage management (ASM) common file catalog, the archive management function can locate and initiate a restore of an archived file without user intervention. This function is known as automatic transparent restore, or IXR. With IXR processing, the user request, process, or program attempting to access an archived file is automatically suspended, the file is restored by ASM, and the process is allowed to continue without failure. IXR helps to ensure a successful file management plan by removing the greatest fear that users have of any storage management system — not having their data when they need it.

Threshold-Based Archiving

A specialized capability of an archive management function is threshold-based archiving. Regardless of how careful systems and intranet security administrators are in defining archive policies, inevitably there will come a time when a process unexpectedly demands more storage from the file system than was anticipated. Sudden and unexpected file shortages can

be disastrous to planned work and the users of the system — as the file system becomes exhausted and work halts. Up until now, the only answer was to run another backup as quickly as possible, and try to guess which files could be deleted without causing too much other disruption. This process was slow, disruptive, and error-prone at best. Threshold-based archiving utilizes a common file catalog to determine which of the next set of files eligible for archive are currently backed up. The catalog is then updated to indicate that the files have been archived, and deletes it from the disk file system. No additional backups are taken, no best guesses are made, and no costly disruption of work in progress results. IXR, of course, stands ready to bring back any file needed.

MULTIMEDIA STORAGE MANAGEMENT

Multimedia management uses a rule-based, policy-oriented design to provide comprehensive storage resource functions for a wide variety of media, both permanently mounted file storage devices and removable tape, WORM, and erasable optical technologies. These functions include space management, allocation control and management, I/O optimization, volume defragmentation, and mount management. Each of these capabilities is designed to provide the best possible utilization of the storage devices available, while maintaining the service levels defined by the enterprise's storage management policies.

Extended Data Storage Management

Extended data management enhances and fully automates all data management functions, regardless of platform. It includes reformatting, sorting, compression, and optimization of data seamlessly and independent of physical file organization and data format.

Performance and Error Management

Through data integrity and device failure recovery facilities, system throughput can be optimized and disruptions caused by failures can be minimized. In addition, high-cost, high-performance options of disk devices can be exploited to their full potential.

Finally, take a look at yet another useful data center intranet security method and tool — xswatch.

WHAT IS XSWATCH?

Xswatch, like its predecessors, was built to watch log files for interesting information. Most log files that grow at a reasonably fast rate have a lot of data, but little useful information. There are several extant implementations that either scan an entire log file or use the equivalent of *tail -f* to monitor the file as new lines are added to it. They then cull log mes-

sages that are deemed important or interesting to the implementor. These implementations can be as simple as *tail -f | grep pattern* to ones that are as complex as a full-blown C program complete with its own macro language.

All of these programs have the same basic structure: match a line of text against a pattern/action pair. If the pattern matches, execute the associated action. The application is up to the end user. For example, watching /var/adm/messages for *file system full,* then executing a job to remove all core files older than one day from the file system. Another example is monitoring for authentication failures. A third might be to send a page to an operator in case of an intranet fault.

Xswatch extends this idea by creating a very general architecture that allows the end user to execute almost any arbitrary pattern/action pair. It takes advantage of the PERL programming language's ability to create code on-the-fly instead of imposing a specialized syntax. The result is a workbench for monitoring almost anything that uses a log file.

What Is the Motivation for Writing It?

The motivation for writing (yet another) log watching program was something that did not have a limited subset of the functionality of the language the engine was written in (PERL). Something was also needed that would monitor multiple files simultaneously. Also, an application was needed that had at least the ability to scale. Xswatch's architecture depends on forwarding syslog entries to a central server. On a large intranet, the number of syslog datagrams could consume a significant amount of intranet bandwidth. It seems better to have xswatch running on many machines, forward only the important data to a central server, and leave the uninteresting data on the local machine. Finally, writing xswatch is an experiment in code reuse and software integration. A primary design goal was to avoid reinventing the wheel wherever possible. For xswatch, there was at least partial success achieving these goals.

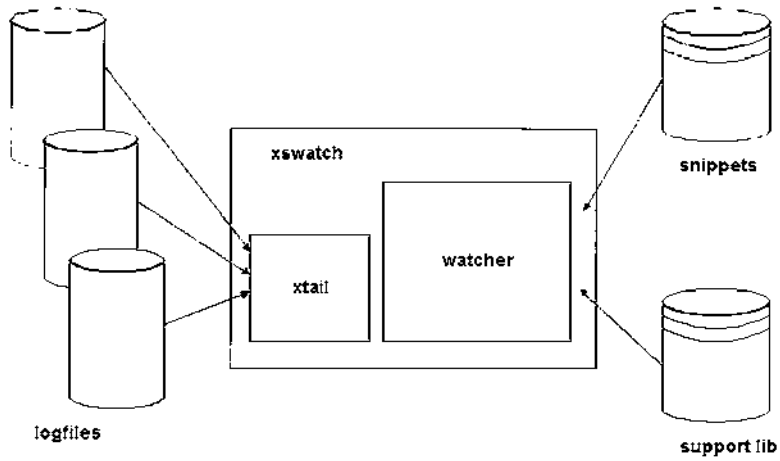
The Xswatch Components

Xswatch has four main components: an engine, snippets, support libraries, and log files, as shown in Exhibit 4. The engine is a small PERL program that essentially manages system resources for the user. The engine consists of four subparts: xtail, a signal handler, an event server, and a watcher function.

The Engine Itself

Xtail is a C program that tails off multiple log files at once. Xtail is the most operating system-dependent part of xswatch. It tracks each file for conditions such as rollovers, truncations, even appearance and disappearance of files.

EXHIBIT 4 — Xswatch Architecture



The signal handler is mostly for managing xswatch in the background. Sending a SIGHUP signal will tell xswatch to kill the current xtail process, reset internal variables, and reload all of the snippets and resume operation. In this way, one can add or remove snippets without halting a running xswatch process. Sending a SIGQUIT signal will tell xswatch to write out configuration information to /tmp. This is useful for debugging snippets. Sending a SIGINT or SIGTERM will tell xswatch to shut down gracefully.

The event server is a PERL module that handles the general housekeeping chores of managing system resources such as I/O, child processes, signals, and timers. The watcher function is a PERL function call (like any other PERL function), except that the function itself is created on-the-fly, based on snippets. Watcher provides the structure that holds the snippets together. Watcher will be called once by the event server for every line of input from xtail.

Snippets

Just what are snippets anyway? The Webster dictionary definition of snippet is a small part, piece, or thing; specifically, a brief quotable passage. For xswatch, a snippet is a small piece of PERL code that handles pattern/action pairs. They have a very simple structure. There is an initialization section that uses the standard PERL *BEGIN* block and then some code, the simplest of which can be:

```
/... some regular expression ... / && do {
... some action ...;
}
```

Snippets are catenated together and compiled into the watcher function. Watcher will be called each time there is a new line of input data written to a log file. The watcher function provides just two pieces of data: an input buffer in *$_*, and the name of the log file where the data came from in *$logfile*. Anything else is up to the snippets.

The snippets are organized in a directory with file names that mimic the *System V rc.d* directories, (*nn.Description*, where *nn* is a two-digit string ranging from 00 to 99). The description can be anything (hopefully informative), with its length determined by the limits of the underlying operating system. This provides a simple way to order execution of the snippets without cutting and pasting. It is also an easy way to temporarily disable snippets. By changing the name from *10.Snippet to stopl0.Snippet*, or anything that does not begin with *nn.*, xswatch will not load the code and compile it into the watcher function. There are no restrictions as to what goes into a snippet. So, if a programmer wanted to put all his code in a single file, xswatch would work just as well as with many smaller files.

The *BEGIN* block is where each snippet should register which log files it wants to monitor. This is done with the PERL *push* call:

```
BEGIN {
push @watchfiles, '/var/adm/messages';
}
```

Registering the same file more than once (that is, in more than one snippet), is allowed. Xswatch will reduce the log files to a unique list. This is because so many people can add or subtract snippets from a common directory and not worry about interfering with other snippets. Notice in the following code that there are a lot of *BEGIN* blocks, and they are actually inside the function call definition of *watcher.* Not to worry though; when PERL compiles the function, the *BEGIN* blocks are executed immediately, once, and never again. When the watcher function is actually called by the event server, the *BEGIN* blocks are not touched.

Support Libraries

Since the snippets are simply PERL scripts, support libraries can be anything one needs or wants. For example, one can use the *Term::Cap* module for setting screen attributes, or one of the date and time parsers to convert the log file time stamp into seconds. Or one can include one's own modules to send a message to one's pager or e-mail.

Useful Features

Xswatch has a number of useful features. First, it is simple. Second, it uses no new grammars. Third, the use of snippets is widely accepted.

Fourth, it can be extended using standard PERL libraries and contributed code. Finally, it is scalable.

A Simple Engine

The engine is simple. The main program is seven lines of code. It knows nothing about its inputs and does nothing with them except to pass them to the snippets in the form of the watcher function. This minimizes code maintenance:

```
$result = GetOptions(qw(help snippet-dir=s debug=i version));
usage("invalid parameters") unless $result;
usage("$0: v$version") if $opt_version;
parse_command_line;
init;
start_server;
exit(0);
```

No New Grammars

Instead of worrying about learning yet another macro language, parsing, and the accompanying errors, xswatch uses PERL itself. As previously mentioned, PERL is well-suited for creating code on-the-fly. It also cuts down on training time. If one knows how to code in PERL, one can write snippets for xswatch. One does not have to look around in the documentation for a syntax guide. Finally, PERL does a much better job at parsing and compiling code than one would ever want to write for a specialized tool like xswatch.

Shortening the Development Cycle

Prototyping and testing are simplified because snippets are written in PERL. One can use *perl -cw* to check the syntax just like any other PERL script. If what one is looking for in a log file has a time domain component, one can do several things: save a time stamp in a variable and set an alarm, or create a new event using the event server. The source code shows a more elaborate sample that checks that the number of authentication failures from log-in does not exceed a certain threshold. This is to handle cases where a user dials in on a noisy phone line. The log-in program will record an authentication failure, but this is not really something to worry about. On the other hand, if xswatch saw repeated login failures over a short period of time, one could guess that someone is trying to penetrate the intranet.

Finally, snippets are readable (well, as readable as PERL code gets). One can document snippets, put them under source code management, and essentially treat them like any other body of code:

```
BEGIN {
    push @watchfiles, '/var/log/authlog';
    require PagerTools;
}
if ($logfile eq '/var/log/authlog') {
TIMER_RESET:
    if (/INVALID|REPEATED|INCOMPLETE/) {
    if (!defined $auth-trigger) {
        $auth_trigger =
        register_timed_client([],600,sub ($hit-count=0});
    } else {
        cancel_registration($auth_trigger);
        if ($hit_count++ > 5) {
            Pager::call_pager oncall, 'Authentication alert';
        }
            undef $auth_trigger;
            goto TIMER_RESET;
            }
        }
    }
```

Scaling

As the number of systems to monitor increases, syslog packets flying back and forth might become a significant load on the intranet resources. By using xswatch to filter out information at the local machine level, one could create an intranet of xswatches that (using the standard PERL Sys::Syslog module) connects to a parent syslog daemon and forwards messages to a centralized server.

Distilling Data into Information

There are a number of applications for xswatch, and one of the most important is distilling data into information. One of a system and intranet security administrator's key duties is to reduce, or at least maintain, entropy. Systems and intranets have a natural tendency toward increasing entropy. System and intranet security administrators are constantly bombarded with log files, e-mail messages, pages, even voice mail. Anyone who has ever had to plow through */var/log/syslog* looking for some clue as to what went wrong knows that it would be far better if, somehow, the system knew what was important and notified a human being in real-time. Xswatch allows an administrator, or even a group of administrators, to consolidate empirical knowledge (about log files) into a single directory. Snippets contain bits of experience learned over time about what to look for in a log file, and what to do about it. Another approach is discard the expected; everything else must be a fault or of some interest. In reference to firewalls, one really does not care when the system resists

an attack, or someone tries to access a blocked port; the firewall has done its job. What one really wants to log is when something goes wrong.

Downsides of Xswatch

Xswatch has its flaws. During the development process, a few issues arose that should not come as a surprise — but did anyway. PERL-centrism is seductive. During the early development of xswatch, and after several ineffective attempts, it was clear that although PERL may be portable, there are some things that really are better implemented in C or some other compiled 3GL. PERL is not cheap. A long-running PERL process will consume about 2 MB of virtual memory on a SPARCstation 5 running SunOS. The resident set size hovered around 1200 KB. Even so, the good news is that there need be only one of these processes running on any given machine. Also, if one considers that an xterm process consumes almost 500 KB, one xswatch process does not hurt so much.

The holy grail of code reuse also has its problems. The most recent version of the EventServer module was incompatible with PERL. Contributed modules tend to lag behind releases of the main body of code, so it is important to track each revision carefully. There are also many PERL modules that, although useful and publicly available, would not work without some effort. Hopefully, as the body of available code matures, these library modules will become more stable.

Xswatch shows promise as an incremental refinement in the publicly available collection of log-watching programs. Note that there are now commercial products in the field that offer more integrated and scalable services. They might even be more affordable! Xswatch makes an effort to encourage code reuse and integration. The maintenance costs for tracking admittedly diverse software packages seem to be less than the cost of maintaining an equivalent amount of homegrown code. Xswatch was also an experiment in toolsmithing — to the extent that the main engine is seven lines of code long. It is modest in scope. The entire xswatch program is six heavily commented pages of PERL code, including the 15 pages of online documentation. Using xswatch is as simple as creating a few lines of PERL script with only two assumptions and installing the file into a directory. Finally, because xswatch does not have its own macro language, it is only limited by the functionality of PERL.

CONCLUSION

Distributed systems and intranet security management solutions must address a very diverse and often perplexing variety of data center environments, each with its own needs and challenges. Distributed systems and intranet security management must also adjust for the vast differences in the various operating systems of each platform, insulating the administra-

tor from the specific vagaries of each system in order to present a unified whole. Through the single point of IT management, overall enterprise policies can be implemented and enforced while continuing to provide the local controls necessary to ensure responsiveness to the constantly changing data center environment.

Finally, xswatch makes an effort to encourage code reuse and integration. Xswatch shows promise as an incremental refinement in the publicly available collection of log-watching programs.

John Vacca is an information technology consultant and internationally known author based in Pomeroy, OH. Since 1982, John has authored 25 books and more than 330 articles in the areas of Internet and intranet security, programming, systems development, rapid application development, multimedia, and the Internet. John was also a configuration management specialist, computer specialist, and the computer security official for the NASA space station program (Freedom) and the International Space Station Program, from 1988 until his early retirement from NASA in 1995. John can be reached at jvacca@hti.net.

129

Physical Access Control

Dan M. Bowers, CISSP

The objective of physical access control is not to restrict access but to control it. That is, the data center manager should know who is granted access, when access is granted, and why. This chapter provides overview of the function of access control systems, the physical elements they can use, and the basic techniques they employ. It also describes two popular access control technologies, keypad access control and portable-key access control, and discusses their advantages and disadvantages. The chapter also examines two other technologies, proximity access control and physical-attribute access control, as well as several developing technologies.

Problems Addressed

Access control devices and systems are an important part of every security system. In a large-scale security system there may be intrusion alarms, motion detectors, exit alarms, closed-circuit television surveillance, guards and patrols, physical barriers and turnstiles, and a variety of other devices and systems. The combined advantages of these elements characterize an effective physical security system. This chapter provides a guide for the data center manager who must determine the optimal combination for an IS installation and networks.

Types of Access Control

This section discusses access control systems and devices and briefly describes the other elements that make up the total security system.

Portal Hardware

Portal hardware includes some simple and obvious devices. The simplest single-door access control system includes at least an electric strike to automatically unlock the door, a timer to make sure that the door does not stay open all day, and a bell or light to indicate when the door is opened or that it has not reclosed properly. There may also be sensors to ensure that bolts are fully engaged and exit switches or sensors to allow people to exit without activating an alarm.

Physical Barriers

To make certain that all persons entering a facility are scrutinized by the access control equipment, they must be prevented from entering areas in which there is no access control equipment. The design of such physical barriers as walls, fences, windows, air vents, and moats is an important part of a security system.

Turnstiles

These can be incorporated to ensure that only one person enters through a controlled portal at a time.

Guards

Many of the most effective security systems use guards and automated systems rather than relying wholly on one or the other.

Other Sensors and Annunciators

In addition to devices used in portal hardware, sensors are frequently useful and can usually be monitored directly by the access control system. These sensors can include intrusion detectors, motion detectors, object protection alarms, smoke detectors, and tamper alarms.

Multiple Systems

Usually, access control systems are provided in conjunction with other security and safety systems. Frequently, there are closed-circuit television cameras and monitors and object surveillance systems. There may be an extensive alarm-monitoring system. Access control is sometimes combined with a time-and-attendance or job-cost monitoring system, because the data required for these systems frequently can be collected at the access control point. Energy management and other forms of facility automation are increasingly being provided along with the security system. Clearly, the more functions that are provided, the more complex the total system design task becomes and the more vital it is that all of the systems efficiently mesh together.

Processors and Controllers

In a simple one-portal access control device, the controller can consist of a single-circuit board containing circuitry that can verify entry codes and energize a door strike. At the other end of the spectrum, a system encompassing access control, fire detection, alarm handling, time-and-attendance monitoring, and energy management will require a sizable computer and an extensive communications controller, along with a substantial software and maintenance investment. Between these extremes, there are a nearly-infinite number of ways in which the required control intelligence can be distributed within the system.

Central Alarm Station

For monitoring and controlling an electronic security system, one alternative to employing a dedicated in-house processor and response staff is to locate this function in a central alarm station.

Electrical Power System

Any security system relies on an electrical power system. Such systems, however, are subject to numerous aberrations, including blackouts (local or widespread) that must be accounted for in a complete system design.

People

Frequently, one of the last factors to be considered in the design of a system is that people are the reason for the existence of data security systems. There are people who must be admitted to the facility without delay, and there may be different sets of people who must be admitted to different areas of the facility, and perhaps only during certain times. There are people who must not be admitted to the facility at all. Consequently, a data security force is necessary to monitor admission activities, respond to alarms, and handle unusual situations.

Designing the Total Security System

In the design of the total security system, it is essential that the user begins with an analysis of risks and threats. However, it is not within the scope of this chapter to provide instruction in risk analysis. Some of the more important studies that should be conducted during this process are:

- *Identifying the most serious risks.* The lesser risks can frequently be resolved as by-products of the basic security provided.
- *Determining the requirements for authorized entrants.* Who is granted entry, how often, and at what times?
- *Examining the geography of the facility.* The physical layout is an important determinant of the required security measures and equipment.
- *Will the various security systems be independent or combined?* Access control, alarms, closed-circuit television, and all other systems should be taken into consideration.
- *Should the security system be combined with other functions?* Energy management, time-and-attendance monitoring, and other functions that may be integrated should be considered.
- *Local control or a commercial central station?* The control center should be located in a secure area for monitoring, management, and response of the security system.

Principles of Access Control

A complete access control system performs three essential functions within the security system:

1. Limiting access through a portal to a defined list of authorized persons
2. Creating an alarm if illegitimate access or activity is detected
3. Providing a record of all accesses for use in postincident investigation

Not all systems provide all of these functions.

To identify authorized persons, all access control systems use one or more of three basic techniques, which have been described as involving something a person knows, something a person has, and something a person is or does. Physically, examples of these three security methods are the combination lock, the portable key, and the physical attribute.

The combination lock is also called a stored-code system; the code is a series of numbers that is stored both in the user's memory and in the lock mechanism, and entry of the correct code by the user with a rotary dial or a set of push buttons allows access. Access control systems universally use a set of push buttons for entry of the code in a combination lock system, and they are usually known as keypad access control systems.

The portable key operates on the principle that if the prospective admittee possesses an object that itself contains the proper access code, that person is qualified to be admitted; the ordinary metal key and lock is the simplest example of such a system. Although ordinary metal keys are easily duplicated and ordinary locks are easily picked, there are key-and-lock systems that are the equal of many modern card-access systems in both security and price; both post office boxes and bank safe-deposit boxes are opened with metal keys (and in both cases the portable-key system is combined with other elements to make up an effective total security system).

The most common form of portable-key access control uses a plastic card with a magnetic stripe as the key, but there are also a variety of sizes and shapes of tokens, metal and plastic keys, and even pens and rings. The code is embedded in these devices by various means, and the key is recognized by a mechanism that automatically reads the code when the key is inserted in a slot, groove, or hole.

Another method of portable-key access control (which is discussed later) uses proximity cards that emit a signal that can be picked up by a badge reader to open doors for authorized persons. Often, card access devices are combined with employee badges to minimize the temptation to allow someone else to use the access control card or to prevent an intruder from using a lost or stolen card.

The physical-attribute system, which is also examined later, is based on recognizing a unique physical or behavioral characteristic of the person to be allowed admittance. In the past, this characteristic has been the human face, and the access control system consisted of a guard who compared the actual face with a picture badge or ID card; this is still the most widespread physical-attribute system in use today. There are also automatic and semiautomatic systems using faces, fingerprints, hand geometry, voiceprints, signatures, and the pattern of blood vessels on the wrist and the retina of the eye.

An access control system is not necessarily a personal identification system, and not all personal identification systems are used for access control. The following categorizations of access control systems may be useful:

- *Universal code or card:* All persons who may be admitted know the same code or carry a card containing the same code, and the access control system opens the portal when it recognizes the code.
- *Group coding:* Persons have a code or card-code that identifies them as part of a group to be admitted to a particular area or at a particular time.

- *Personal identification systems:* A unique code number or set of physical attributes is assigned to each person, and the access decision is based upon whether that particular individual is to be admitted to that place at that time. Personal identification systems have other applications as well, including time-and-attendance monitoring and job-cost accounting data collection.

Weaknesses, Combinations, and Features

There are fundamental weaknesses in all of these basic techniques that automation cannot change. A code can be divulged to an accomplice or observed during entry. A key can be stolen, lost, copied, or given to an accomplice. These situations can occur whether the code and key are meant to open $1.98 locks or are recognized by $100,000 computer systems. Physical-attribute systems have inherent false-acceptance and false-rejection errors, and the two kinds of errors are usually balanced against each other.

Combinations of techniques can greatly increase the security of a system. For example, a code-plus-key system requires that the prospective admittee inserts the key into a reader and enters the proper code using a keypad. This removes many of the weaknesses of the two simpler systems; it also costs more than either of the simpler systems alone.

Other features that can improve the security of an access control system are:

- *Tamper alarms:* If a perpetrator can gain access by smashing or opening the electronic controller, the security level obviously has been diminished. The controls should not be accessible from the unprotected side of the portal, and a sensor should be provided that can detect an attack on the unit and create an alarm.
- *Power-fail protection:* Some units have internal batteries so that an access control device continues to perform its function even if power fails.
- *Fail-safe or fail-soft protection:* The equipment must be expected to fail, however infrequently. There should be a mechanical-key bypass to allow access under failure conditions. When failure occurs, the portal defaults to either permanently open or permanently closed.
- *Code changes:* An effective element of the security system can be the periodic changing of the access codes. Both the code that the person has or knows and the code within the access control equipment itself must be changed.

Keypad Access Control

Keypad access control devices use a combination-lock technique for access control; they require that a correct sequence of numbers is depressed on a set of push buttons or selected from a displayed sequence of numbers using a single push button to gain access. The mechanism may be mechanically operated, in which case the positions of the push buttons operate a mechanism similar to the tumblers in an ordinary lock, allowing the bolt to be manually operated or closing a switch that may operate an electric door strike. Most keypad devices are electronically operated, with the sequence of push buttons being decoded by logic circuits and the door being electrically unlocked.

As in all combination-lock devices, the level of security that is provided depends on the number of combinations available. The number of combinations provided depends on the following factors:

- The number of keys or code numbers provided
- The number of key depressions required to enter the code
- Whether a key may be repeated in the code sequence
- Whether multiple keys may be depressed at one time

Most keypad access control systems use a ten-key pad and a four-digit repeating, nonmultiple code. However, there are systems that use from 5 to 16 keys and from 2- to 10-digit codes, and the number of code combinations ranges from 720 to more than 4 million.

The simplest method of attacking a keypad control system is to try all possible numerical combinations. The defenses against such attack are:

- *Number of combinations:* The greater the number of combinations, the longer the time needed to try them all.

- *Frequent code changes:* A large number of combinations require the perpetrator to try them over a period of days or weeks; changing the code during the period requires the attacker to begin all over again.
- *Time penalty (error lockout):* This is a feature available with many keypad systems. It deactivates the system for a selected period of time after entry of an incorrect number, so unauthorized persons cannot quickly try a large number of combinations.
- *Combination time:* This option is available with most keypad systems. The system controls the amount of time allowed to enter the combination. Because authorized persons can readily enter their numbers, anyone taking excessive time is likely to be unauthorized.
- *Error alarms:* After an incorrect number has been entered (or in some cases, two or three incorrect numbers), these alarms are activated. This option prevents unauthorized persons from trying a large number of incorrect combinations.

Keypad Options and Features

The most significant options and features found in keypad access control systems are:

- *Master keying:* This option allows supervisory persons access using a code that overrides any restrictions (e.g., time-of-day restrictions) on the code provided to end users, and it usually allows the changing of the ordinary code using the keypad itself.
- *Key override:* Sometimes a metal-key override capability is provided for emergency and supervisory use. If this feature is chosen, it must be recognized that the system has been weakened by allowing both keypad and metal-key access.
- *Door delay:* The length of time that the door is unlocked and can be held open without alarm is controlled and usually is adjustable.
- *Remote indication:* There is usually an electrical means of providing a remote indication (at a guard station or central monitoring facility) that a portal is open.
- *Visitors' call:* A special button may be designated so that persons not possessing the combination may request entry.
- *Hostage or duress alarm:* In the event that an authorized entrant is physically coerced into opening a portal, a hidden alarm can be sounded by depressing an extra or alternative digit.
- *Personal identification:* A few keypad systems provide individual access codes for each authorized person.
- *Weatherproof units:* These are provided by many manufacturers for use on outdoor portals. There are also many forms of indoor units, some with attractive decor, and glow-in-the-dark and lighted keypads.

Most keypad access control devices are self-contained, stand-alone devices intended to operate a single portal using a common code. There are also those that obtain their intelligence from a central control unit that can control multiple portals and may also provide logging, space-and-time zone control, and other relatively sophisticated features. In addition, most manufacturers of card-access systems now offer the option of adding keypad access, thus providing a card-plus-keypad system, as discussed in a later section.

Strengths and Weaknesses of Keypad Systems

The cost of a simple, single-door keypad access control device with simple electronic keypads begins in the $100 range. The keypad alone, with rudimentary electronics, can be bought for as low as $20, but the organization must then add door strikes and battery or power supplies. Mainstream commercial-grade protection begins in the $100 range for mechanical and electrical keypads, and the electrical versions require an equal additional expenditure for a reliable electric strike and other necessary equipment. Installed costs can range from $200 to $500; for pure combination-lock-level access control, without penalties or gadgetry, these units are worth the expense.

Therefore, the first positive attribute of a keypad access system is that it is the least expensive means of providing electronic access control in place of — or in addition to — the conventional metal lock and key. Some other positive attributes are:

- Keypad access control can be made very secure if it provides many possible combinations and is installed as part of a system of secure, frequent code changes.

- Changing the code in a keypad system is a quick and simple process, unlike rekeying a lock-and-key system.
- Keypads are especially effective in combination with other forms of access control (e.g., cards or personal attribute systems).

On the negative side, some characteristics of keypad access systems that should be considered before the security of an operation is entrusted to these devices are:

- The code can be divulged without penalty. An insider can reveal the code to an accomplice, who then can gain illicit entry.
- Longer codes provide better security but also encourage authorized persons to write them down rather than memorize them. Therefore, they can be stolen more easily.
- The code can be determined by trying many combinations, if the precautions described are not implemented.
- The code can be observed as it is entered. Some manufacturers offer privacy panels to prevent such observation. One manufacturer provides a random and always-changing placement of the digits on the keypad, using an LED display, so that the numbers cannot be deduced by observing the positions of the depressed keys; another has a rolling single-digit display that is selected by a single push button, preventing an observer from determining what digit was selected.

The two most serious defects in the keypad access system are being able to divulge the code without penalty and the observability of the code numbers; for these reasons, keypad access should never be used alone except in minimum-security applications.

Portable-Key Access Control

A portable-key access control system admits the holder of a device (which may be a plastic card or other device) that contains a prerecorded code. The device is inserted into a reader, and if it contains the code that the reader requires, the portal is unlocked. This process is no different in concept from the operation of ordinary metal keys and locks. Modern systems, however, use keys that are more difficult to duplicate, and these systems can provide complex logic, identification, control, and logging functions that a simple key cannot. It should be recognized, however, that some versions of the metal lock-and-key system provide at least as much security as the simplest versions of card-access, at comparable cost.

The plastic, wallet-size card is overwhelmingly the most popular device used for portable-key access control systems. It is offered by 97 percent of the vendors, though 10 percent of these vendors offer other forms of portable keys as well. The second most popular device is a key-shaped token, usually plastic but sometimes metal; Medeco offers a standard metal key that contains an integrated computer circuit. Some versions are small enough to fit on an ordinary key ring. There are also metal cards of various sizes and several other kinds of metal-and-plastic tokens, strips, pens, and even finger rings. There is some merit in selecting a standard system to avoid dependence on a single vendor. On the other hand, there is some additional security conferred by using a relatively unique device.

Coding Methods

Various techniques and technologies are used to store the access code on or in the key device. Many of the early automated systems used simple visible bar codes that were read by photocells. Others used Hollerith-coded cards with punched holes identical to those in conventional computer cards, which were read by a punched-card reader. Some of these systems are still available. Other cards contained an electrical diode matrix reader, and the card made an electrical connection with the reader. These may be viewed as an ancestor of the modern smart card; they functioned with as much intelligence as they could, using the available technology.

Currently, most devices are magnetically encoded, and there are three basic types. The bank-card type has a magnetic stripe. The code is recorded magnetically onto the stripe and can be read, erased, and altered using conventional magnetic tape technology. Because this technology is well-known and readily available, the cards are easily corruptible, and several additional safeguards have therefore been developed for situations requiring

high-level security. Some vendors encrypt the data on the card so that even if it is read, it is not useful to the perpetrator. Many users, including banks, use a keypad in conjunction with the card reader, so a code must be entered in addition to an acceptable card. Malco Systems has invented a technique called watermark magnetics, which embeds a code during the card manufacturing process; the code cannot be altered and can be read only by a special reader.

The second type of magnetic encoding uses bits of magnetic material — magnetic slugs — embedded into the card during manufacturing. It is read by an array of magnetic-sensing heads to determine whether there is a slug at each of the possible positions. Wiegand-effect coding is currently the only popular magnetic-slug method in use. Each Wiegand slug incorporates a small bit of wire that is heat-treated under torsion, resulting in a magnetic snap-action. This creates a consistent signal over a wide range of reading speeds, unlike conventional magnetics, in which the read signal is proportionate to the speed of the card past the reader. Wiegand-effect coding yields superior performance in swipe readers, for example, in which the user manually moves the card past the read heads.

The third type of magnetic encoding is a descendent of the magnetic slug. It has a sandwich construction with a sheet of magnetic material in the center of the card; spots can be magnetized at various positions on the sheet, thus creating coding to be read by a magnetic-sensing head. These are usually called barium ferrite cards (named for their magnetic material).

There are several nonmagnetic coding techniques, many that are unique to a specific vendor who has developed the technique for a particular purpose, to be used only in its product line. There have been embedded-slug systems using capacitive and conductive particles that were sensed capacitively; none are known to be currently available. There was once a card using radioactive slugs that were read by a Geiger-counter type of apparatus (it was not enthusiastically received). There are embedded-slug devices using nonmagnetic metal slugs, which are read by eddy-current sensors similar to airport metal-detecting equipment. There are several devices coded by tuned circuits and read using radio waves; because these do not require the insertion of the card or token into a reading mechanism, they are categorized as proximity access control devices (discussed later in this chapter). In addition, there are several devices that use bar codes (frequently infrared-encoded so as not to be visible). There are also holographically encoded devices; several of these have come and gone since the first one was introduced by RCA in 1973.

The smart card is the latest manifestation of a portable key, though it has been highly touted and widely tested for a decade. The smart card comes in various grades of intelligence; it contains one or more integrated circuit chips, varying amounts of memory, sometimes a battery, and even a keyboard and display. Access codes are stored using various forms of encryption and manipulation algorithms and are communicated electronically to the access control system when requested.

The number of possible combinations of cards, personal identifiers, different companies and facilities within companies, time zones, and other factors that can be controlled by an access control system is determined by the number of binary digits that can be encoded on or in the access control device. Ten to forty binary digits will inherently provide 10^3 to 10^{12} combinations respectively, and the digits beyond those needed for pure access control can be used for such purposes as personal information.

In systems that have more than the number of codes required to merely open a portal (and nearly all do), the extra digits can be used to store the employee's number, shift of work, or other useful information. Encoding this information allows control over employee access by time of day and by area of the facility. It can also provide a unique identifying number for each person, which is automatically entered into a log showing who passed through which portal at what time, thus allowing the system to be used as a time clock. With individual identification, cards can be easily deauthorized when an employee is terminated or the card is lost or stolen. Other features such as antipass-back and in-out readers (discussed later) are also made possible when individual identification is provided.

The ease of counterfeiting the credential in a portable-key system is largely determined by the encoding mechanism. Optical bar codes and Hollerith punches are clearly visible, recognizable, decodable, and duplicatable. Magnetic stripes require more expertise and equipment, but do not pose a problem for the professional with some equipment and resources; the specifications are published by the American National Standards Institute, and anyone can purchase an encoder for $2000. Although embedded materials provide another step in security, analytic equipment is capable of detecting and cracking the code. Smart cards are merely very portable computers, and they are vulnerable to most hackers of respectable skill. Organized crime, competitive corporations, and foreign governments all have sufficient resources to breach such security measures.

Portable-Key Options and Features

Options and features available with portable-key access control systems include:

- *Access device:* This can be a card, plastic key, metal token, or other device.
- *Coding means:* Available technologies include magnetic stripes, Wiegand-effect codes, bar codes, Hollerith punches, barium ferrite, and integrated circuits.
- *Individual identification:* This is the ability to identify particular people at access.
- *Maximum number of portals:* Until recently, manufacturers created systems that were designed for niches of a particular size (e.g., one door, a dozen doors, or hundreds of doors), and the user could select a system well suited to the organization's needs. With the advances in computer and communications systems technology, most systems are physically capable of being connected to a virtually unlimited number of doors. This does not necessarily mean that the manufacturer's software or understanding extends to a system with a large number of portals.
- *Space and time zones and access levels:* These are means of controlling access to particular areas by particular persons at particular times.
- *Keypad:* Most systems allow key-plus-keypad access control to be implemented.
- *Alarm handling:* Most access control equipment provides the ability to recognize and report or act on a specified number of electrical contact closures (e.g., alarm points). These points could be door-open contacts associated with the access control function, or they could be unrelated points (e.g., smoke detectors or intrusion alarms).
- *Degraded-mode capability:* This defines the level of control that survives under failure conditions (i.e., the local controller may provide a less-intelligent form of control if the central computer fails).
- *Code changes:* This defines whether the user can recode cards or tokens or whether new ones can be purchased if code changing becomes necessary.
- *Time-and-attendance monitoring:* Data collection capability is available with many systems.
- *Antipass-back:* This is a feature whereby after a person's card has been used to pass through a portal, the card must exit before it can again be used to enter; this requires that readers are provided both for entrance and exit. Some vendors offer timed antipass-back, a version in which a certain amount of time must elapse before the card can again be used to enter.
- *Individual lockout:* This provides the ability to invalidate a single individual card.
- *Computer interface:* If a standard form of communications interface is provided, the access control equipment can be easily linked to other security or facility management or central database systems.
- *Limited-use cards:* These are useful for visitors or contractors. The sundown card expires on a particular date. The one-time card can be used only once; the limited-use card can be used only a certain number of times.
- *Dual-key access (two-person rule):* Two valid users must insert their cards for the portal to open.
- *Guard tour:* This provides a means of recording that patrolling guards make their appointed rounds at the appointed times.
- *Duress or hostage alarm:* This option is less easy to provide in a pure portable-key system than in a system with a keypad. Methods include running the card through backwards or pushing the card past an over-travel stop on an insertion reader.

Strengths and Weaknesses of Portable-Key Systems

The cost of a simple card reader begins in the $65 range and can go as high as $300. Intelligent single-portal systems with electric strike, power supply, and door contacts may provide some time-period control, individual lockout, and ability to be upgraded by being attached to a central computer; these are in the $500 to $1000 range, and another $2000 can add a logging capability.

Centrally controlled systems begin in the $2000 to $5000 range for mainstream, medium-scale access control and cost about $15,000 for relatively sophisticated features and a large number of terminals. These systems can cost hundreds of thousands of dollars when facility management capabilities are added. To this must be added the cost of the portal equipment. In most cases, costs of about $2000 per portal procure a satisfactory system, including the cost of installation and wiring.

The cost of the access control card or token must be considered during selection of a system. Most of the conventional plastic cards can be obtained for $1 to $2 each in reasonable quantities; the addition of logos, employee pictures, or pocket clips can drive this into the $4 to $6 range. Smart cards are three to four times higher.

The positive attributes of portable-key systems are sufficiently strong to make this method of access control by far the most widely used. The most important assets of portable-key systems are:

- They are pickproof. There is no means of operating the locking mechanism without having an access card that contains the proper code.
- They provide identification of the owner of the card. This is the most important feature. Individuals can be controlled as to when and where they are allowed to enter doors, a log can be kept of what person opened what door at what time, and the access privileges of a particular person can be changed or eliminated at any time.
- Many valuable features can be provided if needed. The two-person rule, sundown cards, antipass-back, timekeeping, and other options are available.
- They can be installed at reasonable cost for the performance they provide.

There are, of course, negative aspects of portable-key systems, namely:

- Cards can be lost, stolen, or given to an accomplice, and the possessor of the card will be granted all of the access privileges of the owner.
- Cards can be copied. This is true regardless of what manner of coding they employ; higher-technology encoding merely requires higher-technology counterfeiting.
- A duress alarm is more difficult to implement in a card system than in a keypad, and few card-access systems have duress alarms.
- The cost per portal is four times that of a keypad and thirty times that of an effective metal-key system, and in many applications it may not be warranted. In addition, if some of the more sophisticated features are not used, the card system may not provide higher security.
- The cost of the card or other forms of portable-key security can be a significant expense if there needs to be a large number of cardholders.

Combinations of individual access control techniques can give the user the best of both worlds, minimizing the defects and maximizing the positive attributes of the individual systems. For example, push-button access control devices are simple, reliable, and inexpensive, and their keys cannot be lost or stolen. However, the keys can be given away without penalty, and there is usually no personal identification capability. All persons possessing the correct code will be accepted by the code recognition unit. Card and other portable-key access control systems can have personal identification capabilities and can be made virtually pickproof; however, cards can be used by nonauthorized persons.

Key-plus-keypad systems combine the positive attributes of both these simpler systems. The person requesting admittance must possess the portable key and must know the numbers to use on the keypad. The numbers may be the same for every entrant, or each may have a different code to remember, or the code can be derived from information on the coded key or be related to the date on the calendar. Other combinations are also in common use; for example, card-plus-face, as on the picture badge, or keypad-plus-fingerprint, using automatic fingerprint recognition equipment.

Portable-key systems are indeed the mainstream in electronic access control, and they are used in every kind of application. When combined with keypad or personal-attribute systems, they provide sufficient security for such demanding applications as automated teller machines and high-security installations of the U.S. government.

Recommended Course of Action

Every security decision requires the balancing of risk and expenditure, and in choosing an access control system for a facility, the data center manger must decide what expenditure is warranted for the solution to the security problem. A total security and life safety system encompasses perimeter control, internal surveillance, access control, fire detection, walls and barriers, guards, employee screening, and audit trails. In many installations, measures are in place for many or all of these aspects, and the data center manager must weigh the costs of new or additional security measures.

The keypad access control system is simply a combination lock that is quicker to operate and more difficult to defeat and that has more features and options than does the version sold at the corner hardware store. Such features as hostage alarms, error alarms, and remote sensors can be valuable in many cases. Push-button systems cannot be employed alone in situations in which there is a large risk of collusion (because the combination can be divulged without penalty) unless one of the few systems with individual identification is employed. Keypad systems can cost ten times what common locks cost, and the increased security and extra features are well justified in many cases.

The card-only system is equivalent to a conventional lock and key, but it is more difficult to duplicate and can have many additional features. When equipped with personal identification, individual control, and access logs, these systems are virtually undefeatable by an amateur. The risk of lost and stolen cards is still present, and entry may be gained before the card's loss is known and its access privileges canceled. Card-only systems can cost 50 times as much as common locks and can provide sufficient additional security to justify that cost when the security needs require it; additional features and side benefits, such as collecting time-clock information, can also help justify costs.

Because no amount of ultra-high technology can create a card that is immune to loss or theft, it does not make much sense to pay a great deal of money for exotic coding techniques. Although sophisticated codes require more effort and resources to crack and duplicate, it will be done if the stakes are worth it. In addition, the security of card systems is not highly dependent on the code or its embodiment.

Card-plus-keypad systems plug the loss and theft loopholes in card-only systems and the collusion loophole in keypad systems; they cost little more than card-only systems and provide substantially increased security. The increased security provided by adding a keypad to a card system may well allow the use of a simple stand-alone system rather than a much more expensive, centrally controlled system requiring options and expensive wiring. Card-plus-keypad systems can therefore be less expensive than sophisticated card-only systems.

Proximity Access Control

Proximity access control defied all logic a decade ago by becoming well entrenched and then boosting its primary — and for a while only — promoter, Schlage Electronics, to the top in sales of access control equipment. The technology was more cumbersome than conventional card or keypad access, the cards and readers were more expensive, the reliability was (perhaps marginally) lower, and proximity still meant that in most cases a user had to extract the card from wallet or purse and place it against a reader instead of passing it through a slot.

Proximity access control continues to capture a significant and increasing market share, which supports half a dozen principal vendors. In addition, nearly all significant access control system vendors now feel compelled to offer proximity readers, though most vendors purchase the equipment from the six primary manufacturers and then affix their own brand or label on the equipment.

Proximity access control systems perform the usual functions of unlocking a portal, powering up a computer terminal, or disarming an alarm system by using a device that is in the possession of the person desiring admittance, but there is no necessity for physical or electrical contact between the coded device and the reading and controlling mechanism or system. Some proximity systems operate as card-access systems do, without requiring the card to be inserted into a reader; others are actually keypad systems without wiring between the keypad and the access control system. Some are automatically sensed when they come into the vicinity of a reader; some require an intentional action by the person possessing them.

In every access control system, a code must be communicated from the user-carried device to a reading mechanism; in keypad or card systems, this communication takes place electrically over physical wiring. In a proximity system, it is accomplished with electromagnetic (including radio and other derivative forms), optical (including infrared), or sonic (including ultrasound) transmissions.

Principles of Operation

There are two basic classes of proximity access control systems: those in which the user initiates transmission of the code to the system (e.g., the garage door opener) and those in which the system senses the presence of a coded device without the user's performing any action at all. These two classes are called the user-activated and system-sensing proximity systems, respectively.

The user-activated systems must incorporate a power source in the device carried by the user. This is a battery in all of the current units, but devices having other power sources are known to be in development. The types of user-activated systems are:

- *Wireless keypads:* The user depresses a sequence of keys on an ordinary keypad, and the coded representation of the keys is transmitted by radio (in one case by infrared light); the system detects the transmission and decodes it.
- *Preset code:* The code is set into the device by means of jumpers or switches (the garage door opener is the most common preset-code system), and the user depresses a single key that causes the code to be transmitted — by radio, ultrasound, or infrared — for the system to detect and decode.

The system-sensing systems implement a variety of technologies, range in cost, and operate at widely differing distances. Some require power from a battery inside the portable device, and some use power absorbed from the interrogating system. The several types are listed in the following sections.

Passive Devices

These devices contain no power source and communicate the code to their interrogator by reradiating the interrogating radio frequency (RF) signal at a frequency (or frequencies) different from the original. The most common technique incorporates tuned circuits in printed wiring on the card. This is similar to the operation of most electronic article-surveillance antishoplifting systems. One system uses a crystalline structure on the surface of the card.

Field-Powered Devices

These devices contain an active electronic circuit, including code storage electronics and an RF transmitter, along with a power supply circuit capable of extracting sufficient electrical power from the RF interrogating field to accomplish a transmission of the code in response to the interrogating signal.

Transponders

These devices are automatically operated two-way radio sets. The device, which contains a radio receiver, a radio transmitter, and code storage electronics, is battery powered. The system transmits a coded interrogating signal that is received by the device, and then the device transmits a return signal containing the access code. This operation is a wireless form of the poll-response process through which a computer communicates with its network of terminals, similar to the method used in air traffic control to identify airplanes to ground controllers.

Continuous Transmission

The device is battery powered and contains a radio transmitter that continuously transmits the entry code. When the device is a certain distance from a protected portal, the transmission is detected and the code is received by the system. Continuous transmission requires more battery power than the other battery-operated methods do; the batteries must be recharged every night.

Proximity Access Control Features and Functions

Proximity systems vary widely in performance, cost, and convenience. No single choice is best for all applications. Some parameters to be considered are:

- *Activation distance:* The distance at which a proximity system can be triggered varies from two inches to nearly fifty feet, with the battery-powered tokens providing the greatest distance.
- *Hands-off vs. triggered devices:* Some devices require the user to push buttons or keys; others require no action and thus need not be removed from pocket, wallet, or purse.
- *Concealment:* Because there is no need for accessible and visible keypads or card readers, most proximity systems can be installed so that the presence of an access control system is not obvious. This precaution in itself can add to the security of an installation.
- *Physical protection:* Because radio and optical waves can pass through such materials as cement, wood, brick, and bulletproof glass, most proximity access control systems can be easily protected from assault and vandalism by placing the interrogating unit behind a barrier.

- *Form and size of device:* Proximity tokens come in a range of sizes — from one that could fit into an empty medicine capsule to cigarette-pack size.
- *Code changes:* Passive cards and most field-powered devices have codes that are embedded and cannot be changed. All of the other devices (which are more expensive) allow the code to be changed by means of internal switches, jumpers, or an external programming unit.
- *Cost of token:* The system cost for proximity access control differs little from the cost of a conventional card-access system. The cost of the tokens varies widely from the high end of standard cards ($4 to $7) for the passive card versions, to the $10 vicinity for field-powered devices, to $15 to $75 for active tokens, and $100 or more for the rugged, sophisticated tags used in manufacturing applications.

Strengths and Weaknesses of Proximity Access Control Systems

Proximity access control systems offer several unique features:

- The user is not required to remove a card from the wallet and pass it through a reader, but must be within the prescribed range of the reader.
- Because the readers can, in most cases, read through such materials as wood or plastic, the reader can be concealed, both to hide its presence from intruders and to protect it from vandalism.
- Because the reader can be placed within a wall, for many products it can be made to read on either side of the wall, thus providing both card entry and card exit using a single reader.

The disadvantages of proximity access control systems are:

- The more popular systems have a range of only a few inches; this requires that the user hold the wallet or purse very close to the reader, which somewhat reduces convenience.
- Because the proximity systems are wireless, they are susceptible to errors caused by transmissions and reradiations from sources exterior to the security system.
- Systems that have substantial reader range can have problems discriminating when more than one token-holder is within their field, because they can receive multiple transmissions.
- The cost of proximity access control systems is, in general, higher than that of card-access systems with equivalent features.
- Some proximity systems have a relatively low code capacity, though there is no inherent technical limitation for most kinds of systems.

There are many applications for which proximity access control is quite beneficial, such as those in which persons must open portals while burdened with packages or driving a vehicle. The ability to hide the reader within a wall is also important to applications in which vandalism can be expected and adds to the security of the system. The long-range systems are also used in personnel-locator and personnel-tracking systems, because they can detect a token-holder within the space under surveillance, without any action on the part of the token-holder. Most systems, however, are installed in conventional access control applications, in which card access would have done as well, and these system-sensing, passive-card systems must be considered part of the established mainstream of access control products.

Physical-Attribute Access Control Systems

The ultimate in reliable access control would uniquely identify a person and admit that person and only that person, regardless of whether the person possessed a particular coded token or knew a particular code. This ultimate system would be based on recognition of one or more physical attributes of the person. Automated systems for performing such a function have been available since the early 1970s; they are variously called physical-attribute systems, personal-characteristics systems, and biometric systems.

For two decades, access control industry experts have predicted widespread use of these systems, saying that only the cost problem stood in the way. For the past five years, these predictions have come almost entirely from those who have a vested interest in the technology, as the market share of physical-attribute systems has dwindled from insignificant to miniscule and the vendors have struggled, disappeared, or sold out. Although these systems eventually may predominate, the immediate prospects seem less promising than they did a decade ago.

Physical-attribute identification systems of the nonautomated variety have been in use for centuries (i.e., recognition of the human face by guards). In this century, picture-badge systems were introduced, allowing the guard to compare the face on the card with the face of the person; such systems use the human face as the unique physical attribute and are still in use in high-security installations of the U.S. government, on passports, and on the drivers' licenses of many states (which have become the most commonly accepted form of identification for banking and credit transactions). Two other physical attributes are also well-accepted means of personal identification: the signature and the fingerprint.

Many automated and semiautomated identification systems using these three basic physical attributes have been developed. Some are still available and are in common use. Three additional physical attributes have been added to most recent systems: the geometry of the hand, the characteristics of the voice, and the pattern of the blood vessels on the wrist and the retina.

Facial Recognition Systems

Access control using recognition of the human face is the most venerable form of access control. There is no fully automatic system using the face as the physical attribute. There are, however, semiautomatic (or machine-assisted) facial recognition systems that are really improvements on the concept of the picture badge; instead of the picture being carried on a card outside the system's control (and therefore subject to counterfeiting), the reference picture is stored internally (on microfilm, video tape, or disk) and presented to the guard for comparison with the actual face. An employee number is used to retrieve the reference picture from the file, thus making this a sort of face-plus-keypad system. Such systems cost several thousand dollars per portal. This kind of stored-face system has been offered by various vendors over the past two decades, beginning with Ampex in 1972.

A new form of machine-assisted facial recognition system has achieved considerable popularity during the past few years. Begun on the seemingly unpromising premise that users would be willing to pay $30,000 or more for a computer and video ID badge-making machine — rather than a $5000 film-based setup — video ID systems have burgeoned into full-fledged access control systems that present the photo of any person stored in the system at any remote station so that a guard can make the comparison with the real person.

There are also face-based access control systems that present a side-by-side display of a prospective entrant's face along with the picture ID that the person presents. These systems are remote picture-badge inspection systems.

A simple form of face-based access control is becoming commonplace in multiunit housing and is also offered for single-family homes. This is the video intercom, which allows the occupant to both speak with a visitor and see the visitor's face before opening the door.

Signature Comparison

The signature is the basis for personal identification in millions of financial transactions every day. When a signature comparison is made — usually at the bank teller's window — it is done by a teller who has no training in the subject, but is aided with the use of a personal identification number (PIN). There are a number of machine-assisted methods for facilitating signature verification by automating the presentation of the signature to the teller; these are not typically used for access control.

There is no fully automated system offered for signature comparison — for example, pattern recognition of a previously written signature against a file signature. All fully automated systems use the manner in which the person writes the signature as the physical attribute — pressure, acceleration, and speed — not the appearance of the finished signature. This technology was developed by the Stanford Research Institute (SRI) during the 1970s, and several companies, including IBM, have promoted it.

Fingerprint Comparison

Fully automatic fingerprint-comparison systems have been available for 20 years from a continually changing cast of vendors. There is, in fact, a substantial and very productive automated fingerprint search operation in place at the FBI, making 14,000 searches a day through a file of 23 million prints, and from which stems the technology of the commercially offered access control systems.

Two fundamental approaches have been taken to the problem of automatic recognition of fingerprints. The first is through pattern recognition — comparison of the form, whorls, loops, and tilts. The second and most

accurate is the recognition of the singular points that are the endings and splittings of ridges and valleys, called minutiae. There is also a semiautomatic system that presents the reference print and the actual print of the person in a form convenient to make the recognition decision. The fully automatic systems generally cost in the range of $5000 per portal.

Hand Geometry Systems

Hand geometry as a physical attribute on which to base an access control system stems from a 1971 study by SRI in which glove measurements for U.S. Air Force pilots were statistically measured, with the aim of reducing manufacturing variability and increasing inventory efficiency. SRI concluded that human hand geometry is a distinct, measurable characteristic that can be related to individuals. In addition, SRI concluded that standards can be established that greatly reduce the probability of cross-identifying a particular individual.

On this premise, Identimation Corp. introduced an access control system in 1972 during a time when interest in physical-attribute identification systems was at its peak. Most of the efforts were concentrated on the more conventional attributes of face, fingerprint, and voice, and the professional pattern-recognition community skeptically viewed handprint recognition. Yet the Identimation system survived in the market until it was abandoned by Stellar Systems, Inc. in 1988. Other introductions of hand-geometry products have been made, without great success.

Prices of hand-geometry systems are comparable to those of sophisticated card-access systems.

Retinal Pattern Recognition

In 1983, a personal-attribute access control system was introduced that was based on the premise that the pattern of the blood vessels on the retina of the human eye is a unique identifier, following research presented in a 1935 medical paper. Blood-vessel pattern systems have been introduced from time to time, but none has endured. These mechanisms are best suited for controlling physical access to secure areas with a low volume of traffic because:

- They are too slow to avoid unacceptable backups during significant traffic times (e.g., shift changes).
- Hygiene problems may arise from placing the eye against the eyepiece.

Voice Recognition

Despite considerable research and development work over 20 years, there was no offering of a voice-based access control system product until 1985, when there were two introductions. Voice recognition may prove to have certain significant advantages over other physical-attribute systems: the input device can be an ordinary telephone handset, and the internal workings are entirely electronic and should continue to decrease in cost. Other systems require mechanics, optics, and other relatively expensive technologies. Successful technology has proved elusive, however, and the voice-access companies are either defunct or dormant.

An Assessment of Physical-Attribute Access Control

Although industry experts predicted for a decade that physical-attribute systems were the future of access control, that future has continued to be much further away than was anticipated. A large part of the problem is cost: the per-portal cost can be more than twice that of a sophisticated card-access system. The second problem is the absolute unavoidability of false-acceptance and false-rejection errors. Even though the physical attribute itself may be unique, the measurement of it may be imprecise. The questions that a designer of a security system must resolve when considering physical-attribute systems are:

Is the system really more secure than the alternatives?
If it is more secure, is it worth the added cost?
Can the attribute be faked, resulting in potential penetration risk?
Is any one attribute more reliable than the others?

As always, there is no standard or universal answer. Each security situation must be analyzed and choices made that are appropriate for that system.

The error rate of a personal-attribute system depends primarily on how it is used within the total system. If the prospective entrant presents a finger (or face, voice, hand, eye, or signature) to the system and the system is required to determine whether this fingerprint exists among a (possibly huge) file of acceptable persons, a relatively high error rate can be expected. If, however, an identifying card or PIN is also presented, the system is required to determine only if the fingerprint does or does not match the fingerprint that is on file for that person; very low error rates, in the tenths to thousandths of a percent, can be achieved with a personal-attribute system that uses this technique. Of course, such a system is really a combination system — attribute-plus-card or attribute-plus-keypad — which always results in increased security.

In addition, there is some concern that the digitized signal of a biometric reader could be captured and played back to bypass the reader and thus defeat the system, though this concern is related more to computer system access than to physical access to a restricted area. Another biometric access control system currently being marketed involves keyboard dynamics, which records the key strokes used to type in a password or passphrase and compares them with the actions of a person trying to gain access. This is similar to the signature comparison process. This system appears to be quite accurate but also is probably more appropriate for computer access control use.

The bottom line on personal-attribute access control systems is that when combined with card or keypad, they are accurate and reliable and provide excellent security; whether they provide sufficient additional security over a card-plus-keypad system to justify the substantial increase in cost must be determined by the buyer.

As to which personal attribute is the most effective identifier, all of the attributes currently used are roughly equivalent in accuracy. High technology does not by itself provide high security; satisfactory security is provided by a well-designed total security system.

Recommended Course of Action

Physical-attribute systems will one day be the ultimate in access control, but they have yet to achieve any important acceptance or to stand the test of time in the mainstream of access control applications. Still, the data center manager must keep abreast of developments in this and other physical access technologies. To keep their new security systems from becoming obsolete in the near future, they should consider:

- *Smart cards:* Massive investments by major credit card companies have not yet resulted in widespread use of these cards. In security applications, smart cards, like biometrics, are too expensive for what they deliver. Marketing pressure will inevitably result in some penetration of these cards into access control applications; currently, however, they have limited popularity and use.
- *Universal cards:* There are already systems that can use almost any coded card as an access control card rather than requiring the procurement of new and special cards. Despite some yet-to-be-resolved legal questions over how universal cards may be used, their use could be an interesting and cost-reducing trend.
- *Wireless systems:* These can reduce costs by eliminating a great deal of expensive installation and wiring. Such systems will continue to become more popular, including some of the simpler proximity access devices (e.g., wireless tokens and keypads).
- *Physical-attribute systems:* Although these systems have achieved credibility as an access control means, they have yet to solve the cost-justification problem, and they have achieved no user following. There will be a continuing trend toward reduced prices, but these systems will be viewed as top-of-the-line and justifiable only in particular situations for most of the next decade.
- *Proximity access systems:* These systems will continue to capture a significant share of the card-access market, using the new capabilities conferred by increasingly intelligent devices at increasingly lower costs. Proximity access may well exceed ordinary card access in popularity in the future, but biometrics will ultimately dominate the market.

SOFTWARE PIRACY: ISSUES AND PREVENTION

Roxanne E. Burkey

INSIDE

User Ignorance, Software Licensing Methods, Effect of Changing Technology, Costing of Staying Up with Technology, Legal Issues and Enforceability

INTRODUCTION

Software piracy comes in many forms. The definition of software piracy is using a software without paying for the rights to that use. With just this basic definition, the issue is stealing. The problem that begins the controversy includes defining when someone knows they have stolen the software, versus whether that is a defensible stance for an individual to take. After all, stealing is still stealing. Therefore, if stealing is wrong for ethical and legal reasons, why is there an issue and what is the best way to decide a solution?

The difficulty in dealing with software piracy is understanding the issues that make it a problem. Issues included and worthy of discussion to understand all sides of the problem are:

1. User ignorance, which plays a large role in individual software stealing
2. Vendors' lack of standardization of software licensing methods
3. Swiftly changing technology, which reduces monitoring abilities
4. The cost to stay up with technology
5. The legal issues and enforceability

Clarifying the point of the problem is the first step to solving the problem. Education of the user/busi-

PAYOFF IDEA

Software piracy represents an ethical as well as business challenge to individuals and businesses alike. The impact on vendors who supply software and the user community at large could limit the growth of the industry for many years if these issues are not addressed. Awareness of the problem and the steps needed to permanently raise the conscientiousness level is necessary for our societal growth. This issue is not one that can be swept under the rug and forgotten about. For vendors to take the entire responsibility to imitate changes to prevent piracy will cost businesses money as well as the trust of the software vendor community.

ness community and oversight by groups like Software Publishers Association (SPA) provide an industry method of problem solution. The ethical issues then become part of the moral fabric of both the individual and business organizations. The following discussion details the issues and provides an ethical foundation for solution.

USER IGNORANCE

Blaming the issue of piracy on someone else is much easier. For that matter, blaming most things on someone else is much easier. Individuals should take responsibility for their actions. When they find any problems that need changes, they should voice those requests to the proper audience. Users blame the piracy issues on the software vendors. The licensing agreements contained within the software package are confusing and potentially misleading. There are no industry standards for licensing agreements. To complicate matters further, the technology changes have also altered ways multiple users in the network type of environment access software.

Privately used software is installed by a variety of users. Some users have little understanding of the information systems they are using. They follow the directions from the vendor for the software installation. Frequently, the first step to software installation states, "Make a copy of the enclosed diskettes using a standard DOS command." The purpose for this activity is to have a set of archival diskettes in case a problem develops with the original set. It is not to provide copies to friends and relatives. They then use this archival set only if reinstallation is necessary.

Purchasing software is somewhat misleading. The purchase does not mean one owns the software lock, stock, and barrel, but rather that the vendor is allowing the use of their information. They provide documentation for gaining the most benefit from using the software and outline what application best suits the software. They do, however, retain all rights to the code contained within that software. The vendor holds the copyright on the software, not the buyer. A lack of understanding of this agreement is the main reason piracy as a crime is overlooked by the ignorant user.

The buyer (user) must read all of the fine print contained in the software package. This provides the acceptable guidelines for copying the software, copying the documentation, exporting the software, and the agreement as viewed by the vendor. When purchasing software, an agreement exists between the purchaser and the software vendor. That agreement essentially states that the vendor is responsible for the performance of the software as they designed it. Users frequently do not realize that the breaking of the seal on the package often constitutes their acceptance of the software and all the agreements that act demands.

In the business environment, the user may not have knowledge of whether the copy of software being accessed is a legal copy or not. Information technology experts within the organization should ensure that copies available for the users are legitimate copies. Users should not bring software from home for their work PCs, nor bring software from work for their home PCs. Users often believe they have rights to the use of the software despite their PC's location. They design single-user software for loading onto one workstation or PC. If one is fortunate enough to have multiple PCs in the home, each usually requires a separate software copy.

Keeping users from accessing or using illegal software in the work environment is the job for the information technology personnel. IS personnel often monitor the work areas to insure no illegal software is present on the company equipment. When illegal software is found, they may erase the program and then limit that user's access to the system. Outside auditors could construe the simple activity of replacing a PC in the work environment and transferring the software on the hard drive to the new PC without erasing it from the original PC, as software theft. Therefore, having an understanding of the licensing agreements of the vendors used by the business is necessary for the information technology staff. This should include the number of users or workstations that access the software. They must inform management when they reach the license limits of users for a software, and acquire additional software or licenses depending upon the vendor's agreement.

Management often lacks a clear understanding of the liability associated with using illegal software. The penalties for illegal usage, once explained by the information technology staff, should become the foundation for the development part of the policy of the organization regarding software. Once these policies are in place, companies need to adhere to the guidelines and enforce appropriate disciplinary action.

Piracy and Internet Shareware

Many individuals are on the Internet. This communication method provides for access to many types of shareware software. Much of this software is distributed to aid communications between the Internet user community. These programs are typically not copyright protected. The only issue regarding this type of software distribution is the possibility of virus transfers to a user. If they overcome this issue through virus-checking processes, then users can readily share this type of software without fear of penalty.

Individuals, businesses, and learning institutions must adhere to the same set of rules regarding copyrighted software. A clear understanding of the agreement between the buyer and the vendor is essential to an effective method of loading, handling, and sharing software. Despite how indi-

viduals perceive the sharing of software, the federal law protects software that is copyrighted. The user is required to read and understand the agreement from each individual vendor, including the rules established for use of the vendor's software. Clearly, there is no defense for user ignorance regarding this form of stealing. Each user has the obligation to question his/her employer regarding the legitimacy of the software being used to avoid being a party to illegal software usage. Stealing is wrong. Ignoring someone else who is stealing is also wrong. Individuals should not steal software, nor allow themselves to be a party to this illegal activity.

VENDOR STANDARDIZATION

The multiple ways in which vendors issue software licenses — either by machine, user blocks, simultaneous use, file servers, and/or sites — are extremely confusing to the user and/or technical support staff. Some vendors allow the same diskettes for installation on a single machine with archival copies generated by the user. Some software will automatically prohibit access when it determines that the maximum allowed simultaneous usage has been reached. Some software can be readily copied, while some cannot. There is, however, very little standardization.

Microsoft™ has developed special encryption coding into its newer software to prohibit copying the software. This is done to prevent a backup or copy routine from working on another machine. If, however, the software requires reinstallation, the original media must be utilized for this purpose. Most vendors do not have the vast resources required to take this extra step in the software development process and seemingly trust the user community to do what is right. Documenting the licensing agreements with each software package places the burden of responsibility firmly on the user, regardless of how the vendor's software allows for copying.

Each vendor's package will contain the written copyright notice. This outlines the items covered under the copyright. They may include the documentation, software (regardless of media), time frame, and company location. They then provide the licensing agreement terms. This details the rights the vendor is granting to the user. These generally include:

1. The acceptable use of the software specified by number of computers, users, etc.
2. Transfer agreement of the Software and Documentation
3. Backup specifically for archival purposes
4. Problems with unlawful copying
5. The removal or alteration of the proprietary notices
6. Unlawful decompiling or reverse engineering of the software
7. Warranty information

8. Specifying the designated use of the software for fee services or personal use
9. Government licensing agreements
10. Export law assurances
11. Other special restrictions

Software vendors are slowly recognizing that standards are necessary to help crack down on piracy. They are using organizations like the SPA and National Computer Ethics and Responsibility Campaign association (NCERC) to find out the problems from the user community vantage point. They recognize that revenue losses are not recoverable and are taking steps to make it easier for the user. Software is too easily transferred from one area to another with very little effort. The technology that creates the need for the software also creates the ability to do it very quickly and blind to all but the most sophisticated users. Software in the PC environment is more complex in distribution than the mainframe environment. The processes and procedures to successfully keep up with the software are not always available in the business organization, and the private user would not normally consider incorporating this activity into their environment.

In a large business environment, which is technology oriented, this can pose a major inventory monitoring problem. It is extremely difficult in a fluid technology environment to keep up with what is running on which machine without the proper controls in place and functioning.

Whether standardization of licensing agreements is present or not, it is still wrong to take or use something without permission. In the case of software, the vendor issues permission on the guidelines for use of the software to the buyer (user). Using a copy of software is illegal. Taking something that does not belong to you despite your understanding is also wrong.

TECHNOLOGY EXPLOSION

The technology explosion in recent years has influenced most of the nations of the world. Businesses are fully aware of the need to use technology to help maintain a competitive edge in the world marketplace. Knowing this is happening and controlling it are two distinctively different exercises. The very methods of data transmission, media availability, and increased user capabilities, and the power of personnel computing, have created a technology monster in many aspects. Ever more users can reach more information in a single day than ever imagined. To meet this information explosion, developmental efforts to safeguard information has become an evolutionary process based on which problem requires addressing. So it is with protection from piracy.

Software developers are developing more effective ways of safeguarding their copyrighted information. The code is increasingly complex. Encryption is a method currently used by more and more software developers to protect their information. The problem with exporting software from the U.S. in an encrypted format is that it crosses government agencies. Rules are vastly different between the Department of Commerce, which handles most export issues, and the State Department's Office of Defense Trade Controls, which views the encrypted software in a different manner. This causes delays in exporting the products because of restriction guidelines in place within the State Department. It is difficult to export something like software when viewed in the same way as a jet fighter. Efforts to change the restrictions, separating the products types, and still protect U.S. companies competing in the global market by allowing encryption capabilities are being addressed by Congress. Until they alter these rules, the capability of other countries to trade encrypted products will hurt U.S. companies.

Many encryption algorithms are offered via the Internet to help reach the source codes for various copyrighted materials. More complex algorithms are required to stay ahead of the competition and the pirates waiting for the newest and best software products. In addition, Internet access typically opens a system to access by anyone else who has the ability to break into a system. Preventative measures, especially on large wide area networks, include firewalls, virus protectors, encryption of secure data areas, and access tracking programs area available to reduce access and stealing. These measures are needed because data access can provide a competitive edge through stealing someone else's information.

Most businesses have a respect for technology and want the benefits it offers to their competitive edge. They frown on businesses acquiring an edge in unethical manners. Business will help lobby for changes to laws and develop safeguards to protect their business information and trade secrets, as well as protect the rights of software developers to protect the products used by businesses. Business professionals, for the most part, expect the rights of their companies and respect the rights of others. Once they are informed of a problem and presented with the money and ethical issues, they take steps (as with piracy) to change accordingly. It would not be surprising to learn that businesses would band together, much like countries do, to eliminate doing business with those companies they find not adhering to the copyrights of others.

ECONOMIC ISSUES

Technology is a very expensive commodity. Many individuals and companies do not feel they can afford the cost to be competitive. The black market over the past 7 years for software has provided individuals with good copies of the software at less than half the retail price. In some cas-

es, the registration numbers have been in line with the vendors' actual number sequences, causing vendor technical support staff to also support these illegal copies. In these cases, the software vendor is taking a double hit for pirated software. The estimated losses, in gross revenues, to software vendors is in the billions of dollars. The cost of running their software support without the revenues is a hidden cost not part of the generally accepted loss figures.

The economic impact to the software vendors is significant. It costs both time and money to develop and support software applications. The loss of revenues is certainly the primary issue. Close on the heels of this issue is the one of extra development costs to foil would-be thieves. Developing encryption methods and gaining the required approvals is a significant investment. To move forward with technology and keep ahead of foreign competition, especially from Japan, requires all the resources of a software developer to be focused on the newest media available and the next generation of business requirements. The long-term effect of doing battle to protect rights already theirs is to increase costs to the paying user or limit the development dollars. In either of these scenarios, the business and individual user lose.

The issue remains constant. It is wrong to steal. Others end up having to pay the price for the thieves. It hurts short term and long term when someone steals from another. The impact of the stealing is potentially on the competitiveness of U.S. vendors versus foreign vendors. If this activity is not stopped by both businesses and individuals, the long-term effect could be a limitation on continued development by U.S. software vendors. Businesses are in business for profits and the technology industry is no different.

LEGAL CONCERNS

Software piracy is a federal crime. The thief is liable to the software vendor as well as penalties for breaking a federal law. The penalties are substantial under tort laws: Electronic Communications Privacy Act, Computer Security Act, Computer Matching and Privacy Protection Act, and Uniform Trade Secrets Act. Individuals and/or businesses found guilty of stealing software are liable for compensatory and statutory damages up to $100,000 for each illegal copy found on site. On top of these costs, the federal conviction could include up to 5 years imprisonment, defense attorney costs, court costs, and statutory damages of up to $100,000. In one case, an insurance company was found guilty of illegal software from three major software vendors on their computers. The settlement costs included $266,436, plus the costs to replace the system software.

More and more companies are being investigated. The reasons for this are varied. The SPA relies on its watchdogs in the field. Larger companies

have been the primary targets of investigation into software piracy. These companies typically have the wider range of technology usage. The software utilized by these companies is more standardized throughout the organization. They often have IS groups that are in charge of setting up equipment for remote office locations. Without the clear policy defined within the organization regarding software licensing, the tendency to save money on departmental budgets and save installation efforts can be easily overlooked. Most IS professionals are aware of licensing agreements. Many of them advise the organization of the legal ramifications of not adhering to these agreements. Unless the business organization is committed to ethical purchasing and usage of software, the IS professional does not have the required backing he/she needs to perform the responsibilities of his/her job. If this is the case within an organization, the cost/benefit analysis never made the impression it should have with management.

Businesses are making policies regarding software copying within their office environment. Many include the policy as part of the new-hire paperwork. It often includes prohibiting the making or accepting of unlicensed copies of software, providing manuals to the employees for the software they are to utilize, and centralizing the purchase, installation, and license registration for all company software. This not only helps ensure that the employees are aware of the policies regarding software, but also provides a mechanism to track the software used within the company. Many companies also perform periodic audits of the personal computers and the licenses on file.

There are steps that organizations can perform to comply with federal laws. These include the education of management regarding copyright infringement, standardized software for company use, central points to gather purchase documentation, scheduled review for registration with software vendors, destruction of any illegal copies found on systems with appropriate disciplinary action toward the offending employee, standards for installation/registration of new software, and scheduled audit reviews by responsible IS staff. In following these guidelines, most organizations can avoid serious problems from an outside audit of their systems. Most IS professionals will advise management of known problems. If the policies are in place for management to listen, then expensive and embarrassing consequences can be avoided.

The legal issues of software piracy are straightforward and to the point. Prosecution of holders of illegal software will be swift and expensive to the extent the laws will provide. Companies found to frequently use illegal software in this country will find the legal system fully functional. By law, software piracy is illegal. The laws are enacted to protect the rights of persons or businesses. Therefore, individuals and businesses have no rights to illegal copies of software. For an individual or business to continue this practice, with the laws currently in place, is wrong.

CONCLUSION

The issue raised by software piracy is an ethical one. The solution includes:

1. More user education
2. Increased standardization among vendors
3. Stronger methods for preventing software replication
4. Improved monitoring capabilities to keep pace with changing technology
5. A clearer understanding of the economic impact to the end user
6. Stiffer legal ramifications for thieves
7. Improved ethical awareness of the very act of stealing

Organizations and individuals suffer from the outcome of software piracy. By raising awareness of the seriousness and consequences of this crime, IS managers can help thwart this type of theft.

Roxanne E. Burkey, Senior Consultant Designer for Nortel's Symposium Professional Services, has provided client analysis and design support for information systems for 20 years. She is a certified Novell Systems Administrator with a master's in Information Systems.

130

Auditing the Electronic Commerce Environment

Chris Hare, CISSP, CISA

With the proliferation of Internet access and the shift to performing some brick-and-mortar transactions online, the need for stability and reliability in the E-commerce arena is becoming increasingly apparent. E*Trade, one of the many successful E-commerce sites, depends completely on its online presence to stay in business. An outage, regardless of cause, can potentially cost millions of dollars. For example, consider the distributed denial-of-service (DDoS) attacks against Yahoo! and CNN. Once a way to stop the attack had been found, thousands of dollars were spent to facilitate the system cleanup, in addition to the lost revenue. This chapter describes a methodology to assess the security and reliability of E-commerce. Based on this author's previous experiences with risk assessment, security, reliability, and Web "touch and feel – ease of use" can be identified as critical to the ongoing success of E-commerce. The approach described in this chapter can assist any E-commerce Web site owner, manager, or auditor in identifying and securing some of these key risk areas.

It Is Possible to Get Your E-Commerce Infrastructure under Control

The most significant challenge in the development and implementation of one's E-commerce environment will be gluing it all together. Success is dependent on a careful marriage of process, technology, and implementation to achieve the end result. Achieving the final goal depends on a comprehensive strategy, understanding legal and export issues, the processes in use, as well as the technology available to perform the work. Design the environment with confidentiality, integrity, and availability as priorities — not as after-thoughts.

Strategy

Do not get caught up in the waves of technology and methods of doing things. Technology is only one part of the entire puzzle. One uses technology to implement already-operational manual processes to reach a larger market. The operational aspect drives the technological requirements, which in turn affect the overall development of the required systems. The implementation of the project is often affected by changing business and legal needs rather than by changes in technology.

Strategy is the key to the development of an effective E-commerce implementation. The people within an organization must have a vision they can use to drive their planning and development activities. This vision determines the goals senior management has and lays the groundwork for how to measure success. Without a strategy, it will be impossible for you, your employees, your shareholders, and customers to determine if you have achieved anything.

Strategy must also be based on the business decisions that an organization will make. The existing corporate policies must be reviewed and implemented to provide consistency in dealing with the public, regardless of the medium the customer uses to access one's services.

Technology Is Only the Method of Implementing Desire

One's team will use the strategy to establish goals they can translate into project plans and then into manageable activities to meet the strategy. When developing an E-commerce strategy, one must consider:

- What are you trying to achieve by moving to E-commerce?
- How closely is your electronic commerce strategy aligned with your existing corporate strategy?
- What existing corporate business processes must be integrated?
- Who is going to use the service? Is it business-to-business, business-to-consumer, or both?
- Who is going to use the services being offered?
- What do our customers want us to offer?

Armed with the answers to these questions, it becomes possible to start addressing the technology solutions that may provide the implementation. As illustrated in Exhibit 130.1, the technology solution is complex and involves many components. Before choosing the individual components to achieve the technology implementation, one must understand how each component in the business process interacts with the others.

Legal

It is a challenge for most companies to ensure compliance with the legislation of the country where they are located or the countries in which they do business. There are local, state, national, and international laws. There are additional regulations, depending on the industry and whether you are a publicly traded company. However, doing business electronically poses new challenges.

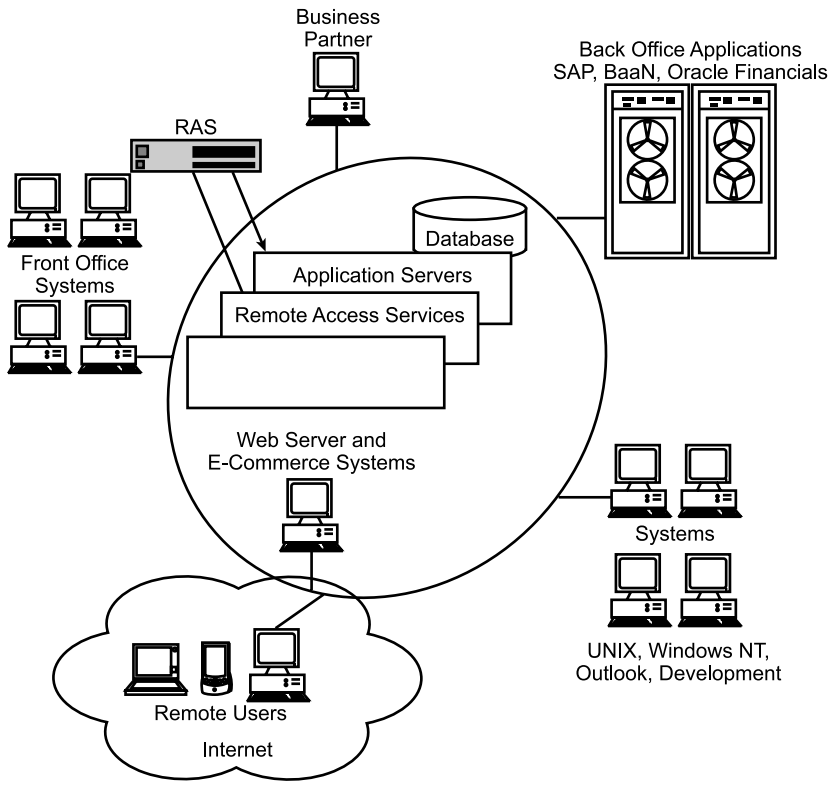


EXHIBIT 130.1 E-commerce system infrastructure.

Privacy

Consumers are concerned about the privacy of their information, while you are concerned about the privacy of information they provide to you or you share with them. Aside from legal requirements in various parts of the world regarding the privacy of information, it would not be good business not to provide privacy controls. If consumers are aware that you do not take this into consideration, they will not do business with you electronically.

The privacy issue can mean some real challenges for an organization. For example, during 1999, the European Union (EU) enacted standards surrounding privacy and the protections of information. The EU stated it might choose not to do business with companies or countries that do not implement similar privacy standards. Consequently, one should specifically state what the organization's privacy policy is. This demonstrates a commitment on the organization's part to the protection of its consumer's information.

Solving the privacy issue means that technical implementers will use words like encryption, digital signatures, and digital certificates. These are technologies used to provide the privacy components to help increase the protection of information sent and received while users interact with an electronic business site.

It is the privacy issue regarding consumer purchasing habit information that led to the development of Secure Electronic Transaction (SET) protocols by Mastercard and Visa, as illustrated in Exhibit 130.2.

All transactions must be properly secured to prevent the loss, through transmission or unauthorized access, of important business information. This must be calculated into the strategy. Doing so will mitigate the risk of information loss and poor performance or reliability from improperly implemented processes or technology.

Export Controls

Export controls are established by governments to regulate export of materials to countries considered dangerous or not in support of the national interest. Most countries do this and in some situations, such as encryption technologies, there are countries that prevent the import of the material.

Compliance with relevant export control legislation is strongly advised. The punishments for noncompliance can be significant, depending on the country and the material exported. Recent years have seen changes in some export rules, again specifically surrounding encryption. Countries have been adopting changes in encryption import/export rules in an effort to allow their producers to compete in the global marketplace.

It is important to review import/export legislation when developing an E-commerce infrastructure. There may be information or technology affected by these rules and they may impact to whom one can deliver the service and resulting products.

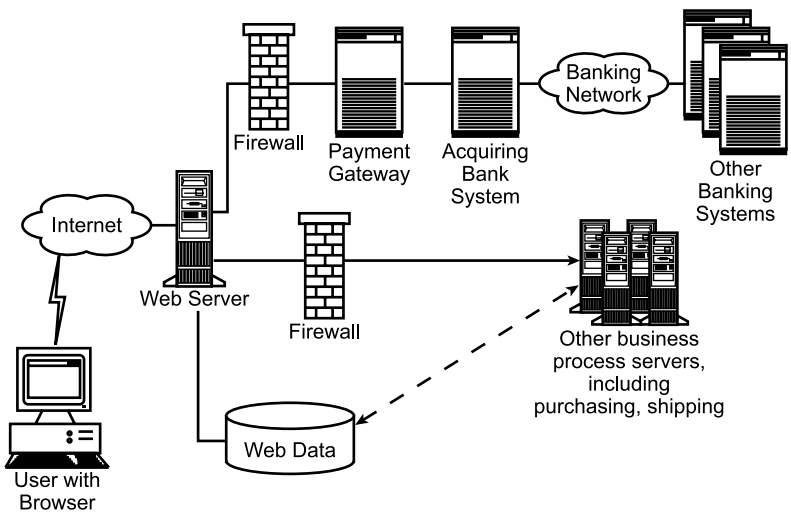


EXHIBIT 130.2 Sample SET transaction environment.

Legislation

Legislation is a major area for many companies. There is a variety of legislation controlling how privacy issues are handled and how business is conducted in general. Much of this legislation is not limited to electronic business. Internet laws and regulations pertain to everything from intellectual copyright to cyber squatting (registering URLs for profit).

The use of a qualified attorney is highly recommended due to the diverse issues and laws involved. With the assistance of an attorney, one should carefully consider the impact of law on the ability to get one's electronic business into full gear.

Considering the vast nature of the law, some areas of concern include, but certainly are not limited to:

- What national and international laws are applicable to E-commerce?
- How is legislative compliance ensured?
- What countries is the business prohibited from selling to through E-commerce?
- Are there distribution agreements and contracts that can be held in force electronically?
- Do the businesses support digital signatures, and are they considered legally binding within the business' jurisdiction?
- How are domestic and international disputes resolved?
- Is there technology or information requiring export permits before it can be available through the E-commerce infrastructure?

Project Management

With the strategy defined, the team can proceed to define the manageable activities resulting in the actual development and implementation of the infrastructure. However, project management is geared more toward ensuring that everyone understands what work must be done, the timeline in which to do it, and how much to budget.

There are a lot of pitfalls in allowing the team to implement electronic commerce services without project management. It will be difficult to gauge where the project is, and even more difficult to determine when it is finished and how much it will cost.

Project management provides the needed controls to define the project, and ensure it meets the business requirements and is completed on time and within budget. A project management strategy is critical to define the tasks required to complete the project. The project plan defines who owns the project and related sub-projects, and how users will be involved in the definition, development, and testing of the E-commerce implementation.

The project manager defines the work breakdown structure and establishes the milestones to measure progress on the project. The project manager allocates responsibilities and manages cost and resource budgets.

Without effective project management, the E-commerce project can become an expensive, never-ending endeavor that fails to meet the business needs.

The ability to plan a project and then properly implement it allows for accurate cost control and planning decisions. Things to consider:

- Does the project plan accurately define the end objectives in a measurable fashion?
- Are there adequate people and other resources to deliver the project on time and without unplanned resource costs?
- Has a standard project management review been conducted?
- How are project costs captured?
- Is the project on track from both a work and a financial perspective?

Reliability

The E-commerce infrastructure must be available whenever a customer wants to use it (availability), and it must operate as the customer expects it to (integrity). Most people do not realize it but reliability is a major component of security. Consumers want to have confidence that when they go shopping online, the merchant

they want to deal with will have all of its systems operating so that they can browse the catalog, enter their order, have any payment transactions properly completed, and then see the order arrive in a reasonable timeframe.

But what happens when things go wrong? Customers need to have a method of contacting the merchant so they can advise that merchant of the problem and seek an acceptable resolution. However, reliability reaches beyond getting problems fixed. It includes the ability of an organization to know there may be a problem now or in the future. How will the performance of the system be measured? How does one resolve a problem for which one of the service providers is responsible?

Performance

The ability of the systems to provide a reliable, friendly, and valuable experience is essential. Users have high expectations about content, access to the services, and quickly finding what they are looking for. Performance, in the eye of the user, is measured by how long it takes to get the information displayed on their screen. A fancy Web site with numerous animations and pretty graphics may be eye-appealing once fully downloaded, but most users get frustrated and are not likely to revisit if the merchant's home page takes forever to load on their system. Develop for the smallest system, and it will work on all others that need to access it.

The customer's view of performance is affected by the capacity planning of the merchant's Internet access and the servers used to offer the customer services. Failure on the part of the merchant to contemplate the actual level of performance one wants people to have will impact that merchant in the end. Capacity planning surrounding the network and server performance must be tempered by how many users one expects to have access to the site.

Having a plan to quickly respond to performance issues regardless of their cause is essential to stay ahead of customer demand. This translates into having capacity planning expertise on the team. These experts monitor performance on a daily basis to maximize the number of customers who can use the site and ensure there is adequate capacity to handle the increased number of users tomorrow.

Architecture

The second component in addressing reliability has to do with the overall system and network architecture. What systems are involved in delivering the service to customers? It is important to understand how they interact with each other in providing the service. Just as capacity planners are important, E-commerce architects who understand the market are critical. Security professionals who understand security architectures to protect the overall corporation and how to implement them are also essential.

Measuring Performance

The collection of metrics for capacity planning, customer satisfaction, and usage is imperative. Operational statistics are collected as part of operating the business and include such items as technology outages and usage. These operational statistics are generally used to provide information regarding problems and assist in determining where efforts should be focused to correct operational problems. Help desks or customer service areas can be invaluable for recording these kind of metrics.

As all of the operational statistics are collected, they must be analyzed and collated into metrics to report the state of the operation. How is the E-commerce environment working? How many customers have used the site? How much was spent and what was bought? However, metrics must be combined from across the organization to establish the strategic indicators used by top management to determine how the organization is doing and what they should be concerned about. This relationship is illustrated in Exhibit 130.3.

Some things to consider surrounding operational statistics and metrics include:

- What efforts are being made to collect, report, and validate the available metrics?
- What metrics are available from the internal and external service providers?
- Determine the reporting structure for these metrics.
- Determine how these metrics are used.
- What process is in place to use the metrics to create feedback to improve the system or correct problems?

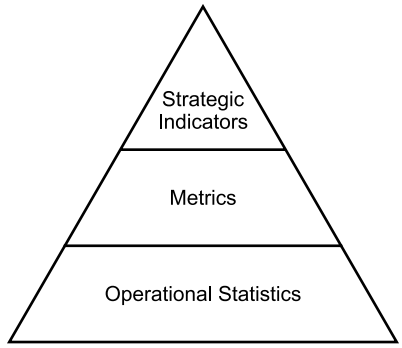


EXHIBIT 130.3 Operational statistics to indicators.

Problem Resolution

The primary users of an E-commerce site are its customers. However, sometimes things go wrong, or customers have questions arise during their visit and would prefer to talk with someone regarding the issue. Consequently, they need to have a place to report these problems or ask their questions.

This requires the implementation of a customer call center where problem reports regarding the Web site can be taken and directed to the correct support groups for resolution, or product questions asked and answers provided. Effectively operating this customer call center requires the use of a call tracking system capable of tracking the customer's issue and a history of what was done to provide resolution.

If operating a global company — and face it, if you are running an E-commerce site, your consumer audience will be global — you will need to establish a method for people to reach you in real-time from anywhere in the world.

The customer call center must be able to respond quickly to customer needs and provide the information they are requesting in a timely fashion. Doing so establishes confidence in the mind of the consumer about your abilities and enhances their buying experience.

When considering the call center, the following questions should be considered:

- How do both you and the customer evaluate satisfaction level?
- How long does it take to solve a problem once reported? Is the customer satisfied with the resolution? Is follow-up necessary?
- What are the common problems reported and what has been done to rectify them?
- What problem tracking and resolution system is in use?
- Are problems recorded so that metrics can be obtained and trending reasonably retrieved?

Service Level Agreements (SLAs)

Service level agreements (SLAs) establish the terms of service, including expected operational performance and problem escalation and resolution. Both issues are important in E-commerce activities. The operational performance of the service provided is critical because poor performance means the E-commerce services will be unavailable to the customer. This in turn can negatively impact both the bottom line and the image of the company on the Internet.

Timely resolution of problems is also important for the same reasons. Customers expect service level timelines for issues to be met. What SLAs are there with service providers, and are there penalties if they do not meet their commitments?

SLAs are also used to assist in measuring the capabilities of your service providers and are useful to have when renewing contracts. Having collected and maintained good information regarding performance and issue resolutions, one will have more success negotiating changes in the contract and price due to good or bad performance in the service delivery.

Things to remember when reviewing the SLAs in place for an E-commerce environment include:

- Obtain SLAs from suppliers such as ISPs and network providers.

- What quality-of-service provisions are in the SLAs? Are the service providers meeting these agreements?
- Do the service providers and your own organization maintain records on their performance?

Maintaining the Business

The ability of the infrastructure to recover from a systems failure, connectivity loss, or other issue is essential. Order entry for product sales is a critical activity that must be maintained. How will the organization handle the partial or complete loss of its E-commerce infrastructure? Are appropriate plans in place to maintain the E-commerce business?

Business continuity and disaster recovery planning form important elements in any business, but are not centered solely on the E-commerce services being offered. Business continuity is centered on maintaining the business operations after a fatal systems failure. For example, can E-commerce operations be maintained if several systems suddenly fail?

These are important questions to ask support organizations. If the organization is heavily dependent on the ongoing operation of the E-commerce environment, then a failure for even a short period of several hours can have disastrous effects on the business. If operating an enterprise based more on "foot traffic," one may be able to afford the downtime.

However, in today's information age, when an online business is offline, everyone hears about it — very quickly.

Areas of concern surrounding business continuity include:

- Has a business impact analysis been conducted to determine how important E-commerce is to the survival of the organization?
- Are the Web servers and other systems involved in the E-commerce delivery part of a contingency plan?
- Are there backup procedures, dependable backups, and regular data and system recovery testing?
- Is the status of systems monitored to maintain integrity and operation?

Development

As mentioned previously, customers will remember their experience with an E-commerce system based on how it worked for them. Consequently, the development of a consistent interface is required and can only be achieved through good development practices.

Standards and Practices

The key method of ensuring that consumers have a positive experience with an E-commerce site is to establish development standards and practices. These are independent of the "look and feel" established as their interactive experience.

The site developers use standards and practices to provide information and methods on how the applications will be developed. This includes things such as code standards, security, and how information submitted from the consumer will be validated and protected. Accordingly, security needs to be designed into the application from the start and not included as an after-thought.

Developers will make decisions regarding how they will develop and write their particular part of the system based on their previous experience or education. These differences make it difficult for ongoing maintenance and subsequent troubleshooting and issue resolution.

Change Control and Management

Change control is a critical part of the overall development/production cycle. Proper change control reduces the risk of improperly tested application code being placed into production, causing problems with data integrity, confidentiality, or reliability. It is also used to identify the changes that are made from day to day to the application code and allows for proper issue resolution and developer education.

A major issue with the development of application code is the fact that it is often put into production systems and "debugged" while customers are using it. This type of activity not only impacts the development of the system, but also affects the user's perception of the E-commerce site and the online presence of your enterprise.

Proper change control ensures that development code is tested in a development environment and is able to process not only the accurate information that the consumer provides, but also handling errors in the input, made either deliberately or accidentally.

Proper processing of information that is collected on the Web site affects business operations. Failure to process it correctly may result in improper or incorrect charges to the consumer, or delivery errors resulting in lost merchandise and increased costs.

When assessing the configuration and change control environment, one must consider:

- Software release change and version control, including both the application code and operating system changes.
- Is it possible to maintain a stable operating environment in today's fast-paced world? Is it possible to automate the change process?
- Development, implementation, and migration standards.

Connectivity

Connectivity is specifically concerned with the technologies used to establish network connectivity to public and private networks, how available bandwidth is calculated, and how the network is designed. E-commerce is very dependent on a successful network design and adequate capacity to ensure that consumers can get to a Web site, especially during the winter holiday season.

This means adequate Internet connectivity speed and capacity, and similar connectivity into your corporate network if applicable to your E-commerce design. Many network design people are leaders in their field, but adequate network capacity can be easily overlooked.

A network can also be overbuilt, having too much capacity and other resources built into it that ties up an enterprise's resources unnecessarily. It is necessary for the enterprise to have good technical management and network design staff to take the marketing and sales plans and build a network that will handle expected traffic and scale appropriately as demand increases.

The network staff must understand that an E-commerce site must be located in an appropriate place. This means that if one intends to operate on a global scale, one may want to consider having multiple locations to ensure the best connectivity and performance for the consumer. This can increase the complexity of one's environment in the process and in turn increase one's dependency on good planning.

Part of this planning includes redundancy, which in turn forms part of one's contingency and business continuity planning. If one component or location becomes unavailable for any reason, one is able to maintain presence and continue operation of E-commerce enterprises.

Consumers are looking for a positive, encouraging experience when interacting with an E-commerce environment. Failing to provide this experience reflects negatively on your online presence. This may result in a perception that the company is not prepared to handle E-commerce and consumers will be reluctant to conduct business with your site.

In reviewing network connectivity, remember to consider:

- Location(s) of E-commerce sites
- Network capacity
- Maintaining and monitoring of network availability
- Network topology
- Redundancy of the network
- Security
- How secure are transmission links
- Do you use a switched network
- Is any form of virtual private network (VPN) used in E-commerce delivery

Security

There are four major components that make up the security area:

1. Client or user side of the connection

2. Network transmission system
3. Protection of the network information during transmission
4. User identification and authentication

Protection of the network security elements and the computer systems that reside in the E-commerce infrastructure is a major portion of protecting the data integrity and satisfying legal and best practices considerations. This level of protection is addressed through various means, all of which must be working cooperatively to establish defense-in-depth.

As seen in Exhibit 130.4, the layering is visualized as a series of concentric circles, with the level of protection increasing to the center. Layer 1, or the network perimeter, guards against unauthorized access to the network itself. This includes firewalls, remote access servers, etc. Layer 2 is the network. Some information is handled on the network without any thought. As such, layer 2 addresses the protection of the data as it moves across the network. This technology includes link encryptors, VPN, and IPSec.

Layer 3 considers access to the server systems themselves. Many users do not need access to the server but to an application residing there. However, a user who has access to the server may have access to more information than is appropriate for that user. Consequently, layer 3 addresses access and controls on the server itself.

Finally, layer 4 considers application-level security. Many security problems exist due to inconsistencies in how each application handles or does not handle security. This includes access and authorization for specific functions within that application.

There are occasions where organizations implement good technology in bad ways, which results in a poor implementation. For example, the best firewall poorly configured by the user will not stop undesirable traffic to a site, or a database security system that has all of the data tables granted for "public" access does not protect the data they contain. This generally can lead to a false sense of security and lull the organization into complacency.

Consequently, by linking each layer (see Exhibit 130.5), it becomes possible to provide security that the user does not see in some cases, and will have minimal interaction with to provide access to the desired services. Integration between each layer makes this possible.

The same is true when implementing security within the E-commerce environment. It must be considered at all layers: the client, the network, the perimeter, and the associated servers. The Web interface has four primary layers: the operating system, the CGI programs, the Web content, and the Web server. Each layer is dependent on the components of the other layers working correctly.

Client Side (User)

Clients interact with the E-commerce infrastructure through their Web browser. The users, however, have certain expectations about how the interaction will look, act, and perform at their computer. For the experience to be a positive one, certain programming considerations must be addressed during design, development, and implementation.

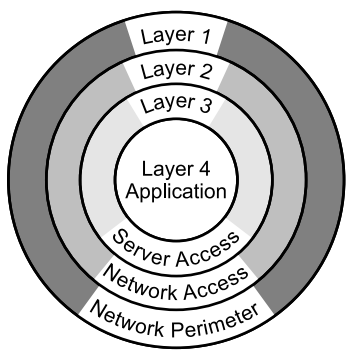


EXHIBIT 130.4 Levels of protection.

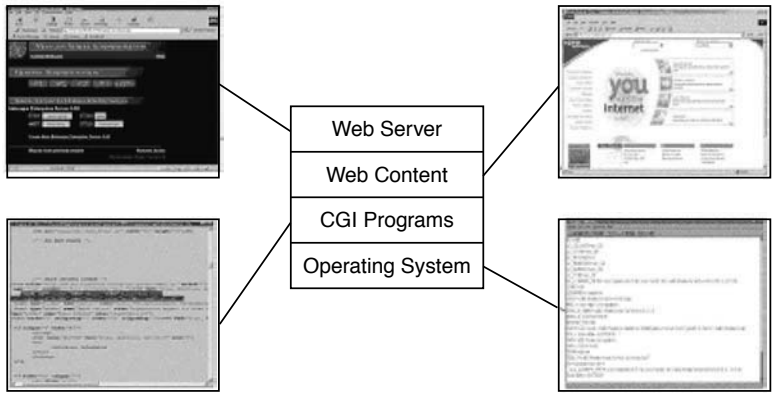


EXHIBIT 130.5 Linking layers.

The experience the user has will be different across the different browser implementations, and choosing to support browser extensions that are not supported by other browsers is not a good business decision. The HTML, dynamic, and graphic content must be compatible with the different Web browsers available. E-commerce applications must consider this requirement. Not all users will want to enable extended features in their browser, such as cookies, Java, and JavaScript. This greatly affects the functionality that can be offered in the design of the application.

The users and businesses that will use a service may not be connected directly to the Internet. They may be using a proxy server to provide security or cache network requests. They may also be using a slow-speed network link. These factors must be included in the design to maintain a positive experience.

When considering client-side issues:

- Examine what types of Web browsers and proxy servers are in use and in what operating environments.
- Determine how a customer registers for E-commerce access.
- Determine the ease of use of the E-commerce interface.
- Decide what applications will be used to develop the interface.

Firewalls

The firewall is an integral part of an E-business architecture. It is accepted that any computer directly on the Internet with no protection is a sacrificial host. One can expect it will be compromised at some point. Although it is not reasonable to hide everything behind the firewall, every system not needing to be directly visible to the Internet should be protected by a firewall. Additionally, no connections from any unprotected systems should pass directly through the firewall to the corporate network.

However, a firewall can be bolstered by the network design through the use of demilitarized zones (DMZs) and service networks (see Exhibit 130.6). The DMZ protects its systems through filters and access control lists in the routers. The service network is a separate network connected to the firewall. Any system that does not need direct Internet connectivity and does not need to be on the corporate network is put in the service network.

The customer interacts with the systems in the DMZ. Additional services required to provide the customer with their experience are obtained by systems in the services network. Any additional information that must be retrieved from systems on the corporate network is retrieved by the intermediate servers. Although this seems to be an overly complex arrangement, there is a high degree of security inherent in the design. The systems outside the firewall have no ability to connect to the corporate network. The firewall is configured to only allow connections from the DMZ to the service network, and then only to specific IP addresses and network services. The systems in the service network are then authorized to connect with systems in the corporate network for the required information.

The use of intrusion detection systems and periodic evaluation using vulnerability assessment tools is also highly recommended as part of an E-commerce security architecture due to the nature of the service and likelihood of attack.

When considering the firewall and network security implementation, examine:

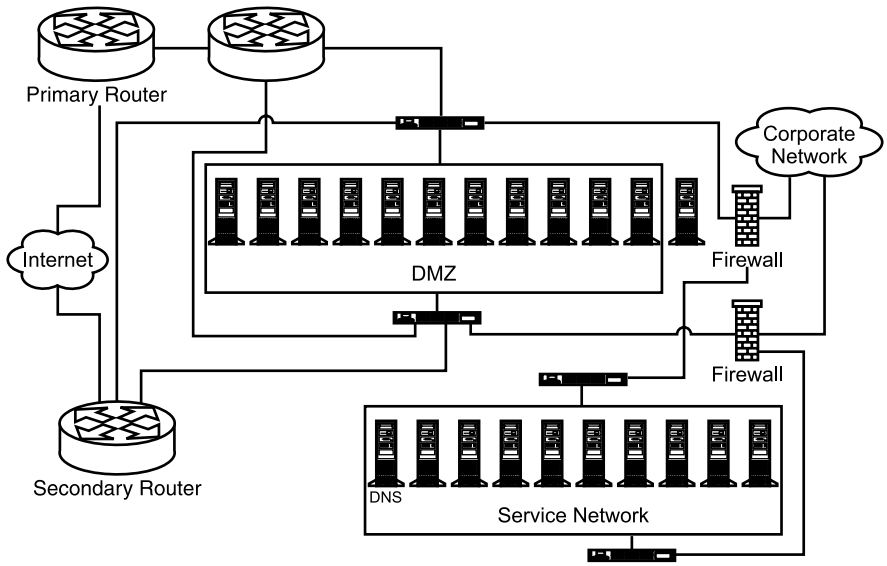


EXHIBIT 130.6 Demilitarized zones (DMZ) and service networks.

- Vulnerability reports of all network elements using a network vulnerability tool such as Cybercop or ISS
- The DMZ systems to determine if they are "hardened" to reduce the potential attack points
- How the Web client and server negotiate SSL encryption and what encryption strengths are offered
- Non-HTTP ports opened through the firewall(s) for browsing and analyze security implications
- The firewall topology
- Firewall configuration files
- Access control lists of network devices
- Network communication protocols
- Configuration management on the network security elements

Securing the E-Commerce Server

The E-commerce server consists of a variety of components all connected together to provide the business service. Multiple systems are used to reduce the complexity of any single system in an effort to improve the chances of properly securing each system. These services include the HTTP or Web server itself, personalization systems, directory systems, e-mail gateways, and authentication systems.

Directory Services

Directory services provide a mechanism for maintaining an online repository of registered users and their related information. By using a central repository for this information, any of the systems requiring authentication data or information regarding the user can access it. Additionally, applications can query information regarding the user, including their mailing information when ordering or requesting hardcopy information or when products are shipped to them.

Several directory systems are available, but those based on X.500 and Lightweight Directory Access Protocol (LDAP) technology provide the highest level of integration and availability.

Because all of the information regarding the users is stored in a central repository, special care must be taken to protect the information on those systems and provide authenticated and secure transmission channels for the data. The repository must have high availability, as many systems will be dependent on its ability to provide the information when requested. As previously stated, the consolidation of the data makes it easier for the administrators to provide confidentiality and maintain integrity while the information is stored and during transmission across the network. One can argue that the consolidation of the data also makes the system

a target for attack. However, the centralization also provides network security personnel with the opportunity to protect the system.

When evaluating the directory services provided, consider:

- How much data will be stored
- How quickly must the directory provide the response
- How many queries can the directory handle at a single time
- What security functionality is integrated into the directory
- Does the directory support authenticated connections
- Does the customer understand that this data is being stored

Mail Server

Electronic mail is a key component in any E-commerce infrastructure. It allows for the delivery of information from the E-commerce infrastructure systems to a user or business. Customers depend on e-mail to request information and to interact with customer service or support people when questions or problems arise. It can also be used by customers to report things they like or dislike about the experience. E-mail, which is used for many things, should not be used as a transport method for information requiring special protection. Information sent via e-mail is as public as a postcard. Consequently, the distribution of credit card or purchase information, as well as user name and passwords, must not be distributed through e-mail. This can be made possible and secure through encryption technologies such as S/MIME.

The operation of the mail server is critical to the infrastructure. E-mail servers are also regularly used by hackers to access other systems or send unsolicited bulk e-mail, or spam, as they are often not considered to be a major security risk. Many of the available commercial mail servers have idiosyncrasies related to their configuration that both can protect and expose information. Consider the incorrectly configured mail server that allows external users to send e-mail as if they were employees of the company, or using the mail server to relay spam to other mail servers.

Such examples are written and documented on a daily basis in the security industry and are usually related to simple misconfigurations, the use of out-dated software implementations, or not remaining current with software patches.

When addressing e-mail security and availability, consider:

- Which mail transport agents and mail user agents are being used
- Access permissions for the mail transport agent's (MTA) configuration files
- Periodic review of the mail server's delivery and error logs to determine the possibility of misuse
- Probing the MTA for common "exploits" to test vulnerabilities to various attacks
- Evaluating the use of virus protection technologies
- Content management and encryption technologies

Web Server

The Web server can be considered the most critical component in the E-commerce infrastructure. It is required to deliver Web-viewable content to the user, run programs to retrieve or send information to the user or other systems, and perform specific checks to determine the validity of requests. It is expected to be available all the time and to provide responses to the user within an acceptable time period. If users have to wait due to poor network or Web server performance, they will quickly leave your site. Once again they will form a negative perception of the business and not be likely to return.

There are a number of Web servers available, both as commercial and freeware software implementations. If one can afford it, buy a commercial implementation to have quick support when issues arise and gain vendor maintenance for the software. Although the initial expense for freeware implementations may be low, and they are quite robust, the post-installation maintenance and support expenses can be quite high. Consider company turnover and retention of experts to maintain the freeware implementation. It is likely to be much easier to find trained experts on commercial software than someone who is familiar with a tailored freeware implementation.

While configuring the Web server itself, development standards are needed for the design of applications and Web content. The Web server software must not execute on the system with any special or administrative permissions. This reduces the risk of an attacker gaining administrative privileges to compromise the server.

The operation of the server is also dependent on the availability of Common Gateway Interface (CGI) scripts to provide access to applications and forms. CGI programs require careful scrutiny during development and before final production to validate that there are no exposures to poorly written code resulting in security issues. Confidentiality and data integrity have been presented several times. The Web server should be capable of providing encrypted sessions through Secure Sockets Layer (SSL) or Transport Layer Security (TLS). Both SSL and TLS require no additional hardware and both use a server-side certificate. The issuance of a certificate for a site is beyond the scope of this chapter. Several reputable firms can issue certificates for Web servers.

Using SSL or TLS, the organization and customer can be confident that the information being displayed or sent is protected while in transit across the network.

When reviewing the Web server, consider the following:

- Review the user ID and account permissions the Web server runs under (i.e., root, administrator).
- Determine which Web sites are public and which are controlled access.
- Analyze access permissions for HTML documents, ASP and CGI, directories and scripts.
- Examine Microsoft IIS or other Web server application configurations and log files.
- Determine how requests received by the Web server from the browser are verified.
- Determine how requests sent to a back-end processor are verified as completed.
- Examine Web-based applications and database connectivity, including Java, JavaScript, and XML.
- Check for the existence of well-known ASP and CGI scripts and utilities that pose a security risk.
- Examine Web and proxy server configuration files.
- Check the Web server configuration files and certificates to enable SSL communications.
- Analyze high-availability components in the E-commerce service.
- Evaluate operating system and Web software patch levels and configuration files on critical servers.
- Evaluate application patch levels and configuration files.
- Determine how external E-commerce systems authenticate to internal systems.
- Consider the certificate authority that issued the server certificate and if there is a method for the customer to validate the authenticity of the certificate.
- Evaluate the requirements of non-repudiation features.
- Evaluate CGI scripts and review the program code.
- Consider Web content management.

Operating System Security

All of the components previously described rely on the foundation services provided by the operating system. Although each of the individual application components can be made more secure, without a strong, secure foundation, other efforts are affected. Today, the vast majority of E-commerce systems run on either Windows NT or UNIX operating systems. Each of these environments has its own advantages and disadvantages and system vulnerabilities.

Windows NT Operating System

Windows NT is a popular operating system used to perform specific computing tasks in any infrastructure. Proper configuration of the operating system is essential. If not properly configured and security is not properly implemented, it can be trivial to compromise.

Windows NT relies heavily on the registry to provide both operating system and application configuration settings. Several key services in Windows NT operate at the same network service port. This can provide a remote user with the ability to probe the system and collect important registry information. With this information in hand, such as disk sharing information, user names, and system configuration details, a successful attack can be launched against the system.

When using Windows NT as an E-commerce operating system platform:

- Conduct a scan of all Windows NT systems providing E-commerce services using both host- and network-based vulnerability scanners. Analyze the results and attempt to exploit them on the operating system to gain unauthorized access.

- Review unnecessary services and ports.
- Review registry settings and operating system patch levels and configuration files on critical servers.
- Evaluate configuration and change management on the operating system components.
- Implement virus protection technologies.

UNIX Operating System

The UNIX operating system provides a multi-user, multi-processing environment used for many different tasks. Like Windows NT, however, improper configuration of the security modules and operating system can make it trivial to compromise. UNIX is a much more popular E-commerce environment than Windows NT. Despite the relative maturity of the operating system, new problems with UNIX implementations are discovered on a weekly basis. The visibility of some of the new security issues even makes it to the news media due to the dependence in the computing world upon this operating system.

Like Windows NT, UNIX is not intended to be a secure operating environment. Any security expert can provide a multitude of ways to defeat the security systems on either operating system. Considerable effort is required to "harden" the operating system and reduce the vulnerabilities in the E-commerce environment. As a multi-user operating system, UNIX has a large number of network-based services providing major parts of the system's functionality. Many of these services and ports are not necessary in order to provide E-commerce functionality. These services are often exploited to initiate confidentiality, data integrity, or system availability attacks.

When using UNIX as an E-commerce operating system, be sure to:

- Conduct a scan of all UNIX systems providing E-commerce services using host- and network-based vulnerability scanners. Analyze the results and attempt to exploit them on the operating system to gain unauthorized access.
- Review unnecessary services and ports.
- Evaluate operating system patch levels and configuration files on critical servers.
- Evaluate configuration and change management on the operating system components.

Back Office Applications

The E-commerce infrastructure has communications paths to various back office applications, including search engines, Oracle, BaaN, and SAP, to facilitate the ordering of products from the catalog. These systems are sufficiently protected, as well as the data sent across the network, to restrict protected information access. In addition, there are specific performance and security considerations for these applications.

Search Engine

The search engine is used to find specific documents or Web pages within the E-commerce environment. The quality of the search engine responses depends on how fast this "crawler" can traverse the Web links and pages to produce an index for the location of relevant material. Most search engines perform this work in two stages. First, the search engine "crawls" through the Web pages and collects information. Second, it builds a searchable index for use later when the user requests the search.

Different search engines offer different levels of performance in the collection of this information. This affects the validity of the search results when the user requests the search. If pages that exist cannot be found when the search is requested, the user will think the information does not exist. Consider the negative perception this can have on the user's experience at the Web site. If pages no longer exist or contain irrelevant information appear, the user will become frustrated.

For example, consider the graphs in Exhibit 130.7. Both graphs illustrate basic system activity for two different search engines running on exactly the same hardware. The system on the top makes much better use of the system's resources during the crawling and indexing phases. This improved use of system resources suggests the engine is working effectively. The graph on the bottom shows much lower resource utilization, suggesting the engine may not be capable of handling the workload despite the hardware resources.

User interaction with the search engine is also critical. If the search engine itself has not been properly implemented, it is possible for performance, including the search, to be slow, due either to the software or the

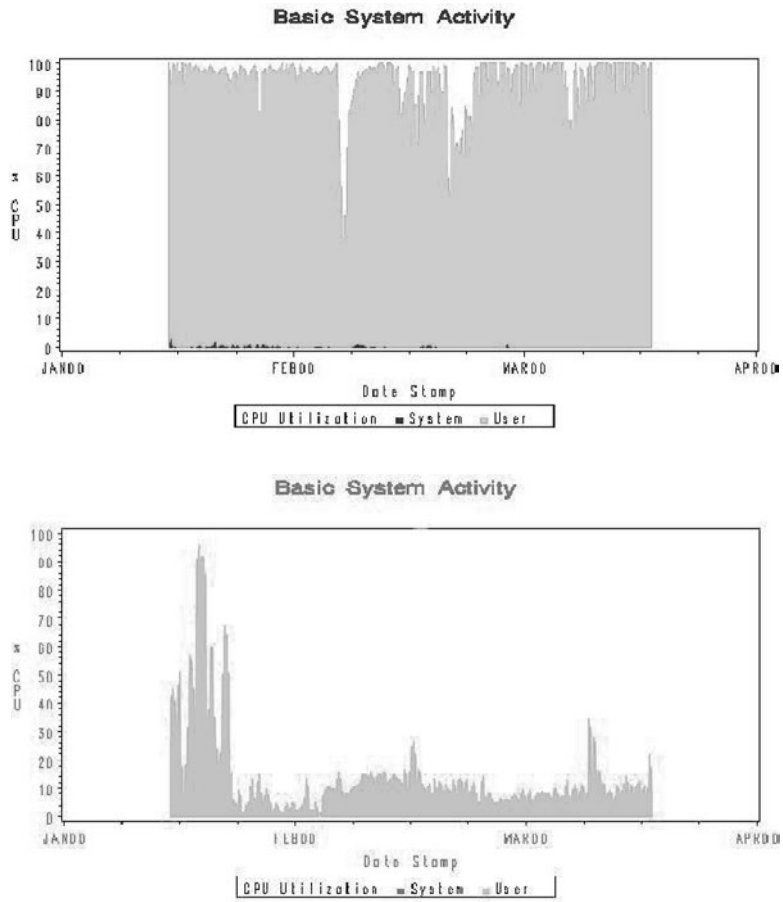


EXHIBIT 130.7 Basic system activity for two different search engines.

hardware on which it is running. Some search engine implementations do not handle simultaneous searches well. Careful review of the product, combined with simulated load testing, is required prior to implementation.

When evaluating the search engine, review:

- How well the crawling and indexing features work
- The success rate and relevance of the returned documents
- The CPU and LAN utilization
- How quickly search responses returned to the user
- The vendor's reputation

The back office systems provide information to the E-commerce user over which the organization wants to maintain strict control. In general, these same systems will be used to provide the day-to-day operations for the rest of the company. Because they are generally within the protection of the corporate network, they can be considered protected. The "hard and crunchy" network perimeter is becoming less and less practical as more and more users and customers are demanding services and access technologies. However, the issues previously presented regarding development, application, and operating system configuration must all be applied here as well.

Communication to these systems from the external E-commerce system is controlled by the firewall. The firewall will only allow specific external systems to communicate with specific internal systems to minimize the risk of total compromise in the event of an attack.

Being successful in implementing connectivity and protecting these back office systems is dependent on a thorough understanding of how data is moved from one system to another, what protocols and transport

methods are used, who creates the data, who processes it on the receiving computer, and the sensitivity of the information itself.

When evaluating and implementing connectivity to back office systems, one must:

- Evaluate protection of sensitive organizational data
- Evaluate configuration management on the back office components
- Evaluate the use of virus protection technologies
- Evaluate database configuration and administration practices
- Evaluate order transmission from the Web site to the order management system
- Evaluate the order fulfillment process

E-Nough!

This chapter has discussed the components of E-commerce architecture and identified what the organization should focus on when developing its environment or preparing to perform an audit. This chapter is by no means an all-encompassing examination of each of the technology areas, but is intended to show the reader the relationship and dependencies of various components that make up an E-commerce environment.

The implementation of an E-commerce environment allows any corporation to economically achieve global presence and enter the global marketplace successfully. In fact, some retailers have no or few storefront (bricks-and-mortar) premises due to E-commerce.

This is a challenging and fast-paced world where it is so important to be first, be visible, and be remembered. Do it fast, be quick, and do it right; if you do not, you blow it.

This is the nature of E-business. If one does not get it right the first time, one will not have enough time to fix it later. This is our E-dilemma!

Acknowledgments

Very special thanks to my colleague and close friend, Mignona Cote. Her insight into many areas in technology, business, and risk areas have taught me many things. Without her assistance, this work would not have been completed.

131

Improving Network-Level Security through Real-Time Monitoring and Intrusion Detection

Chris Hare, CISSP, CISA

Corporations are seeking perimeter defenses without impeding business. They have to contend with a mix of employees and non-employees on the corporate network. They must be able to address issues in a short time period due to the small window of opportunity to detect inappropriate behavior.

Today's Security Perimeter: How to Protect the Network

Many companies protect their networks from unauthorized access by implementing a security program using perimeter protection devices, including the screening router and the secure gateway. A screening router is a network device that offers the standard network routing services, and incorporates filters or access control lists to limit the type of traffic that can pass through the router. A firewall or secure gateway is a computer that runs specialized software to limit the traffic that can pass through the gateway. (The term "secure gateway" is used here rather than the more generic term "firewall.")

Although on the surface they seem like they are doing the same thing, and in some respects they are, the router and the secure gateway operate at different levels. The screening router and the secure gateway both offer services that protect entry into the protected network. Their combined operation establishes the firewall as shown in Exhibit 131.1.

Establishing firewalls at the entry points to the corporate network creates a moat-like effect. That means that there is a "moat" around the corporate network that separates it from other external networks.

The Moat

Although the moat provides good protection, it reduces the ability of the organization to respond quickly to changes in network design, traffic patterns, and connectivity requirements (see Exhibit 131.2). This lack of adaptability to new requirements has been evident throughout the deployment of the secure gateways within numerous organizations.

One of the major complaints surrounds the limited application access that is available to authorized business partner users on the external side of the firewall. In some situations, this access has been limited not by the authorizations allowed to those users, but to the secure gateway itself. These same limitations have prevented the deployment of firewalls to protect specific network segments within the corporate network.

Many organizations are only connected to the Internet and only have a need to protect themselves at that point of entry. However, many others connect to business partners, who are in turn connected to other networks. None of these points of entry can be ignored.

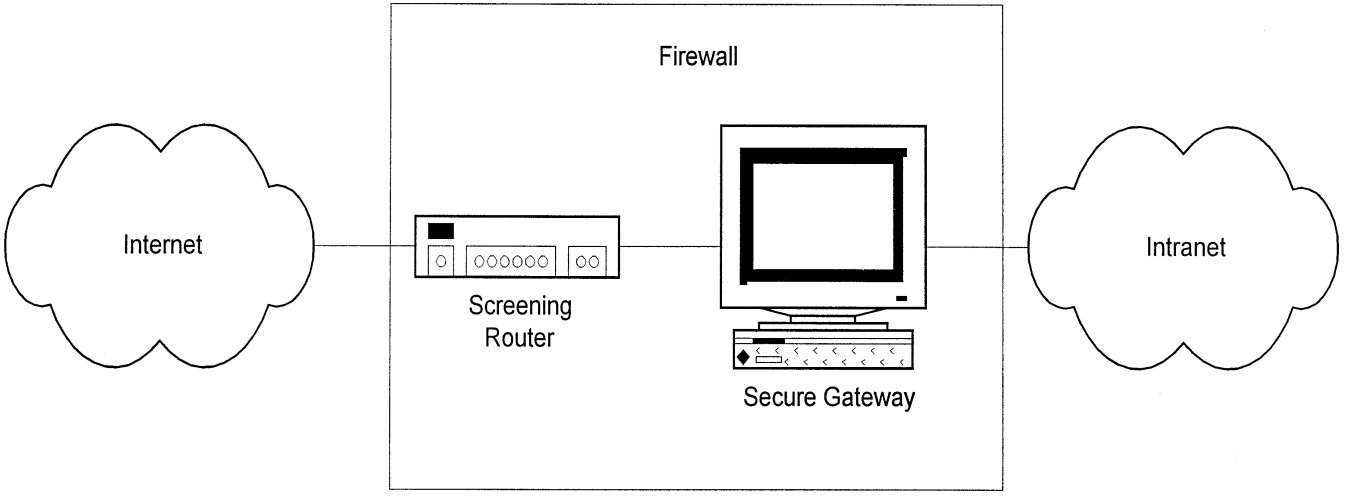


EXHIBIT 131.1 The firewall is composed of both the screening router and the secure gateway.

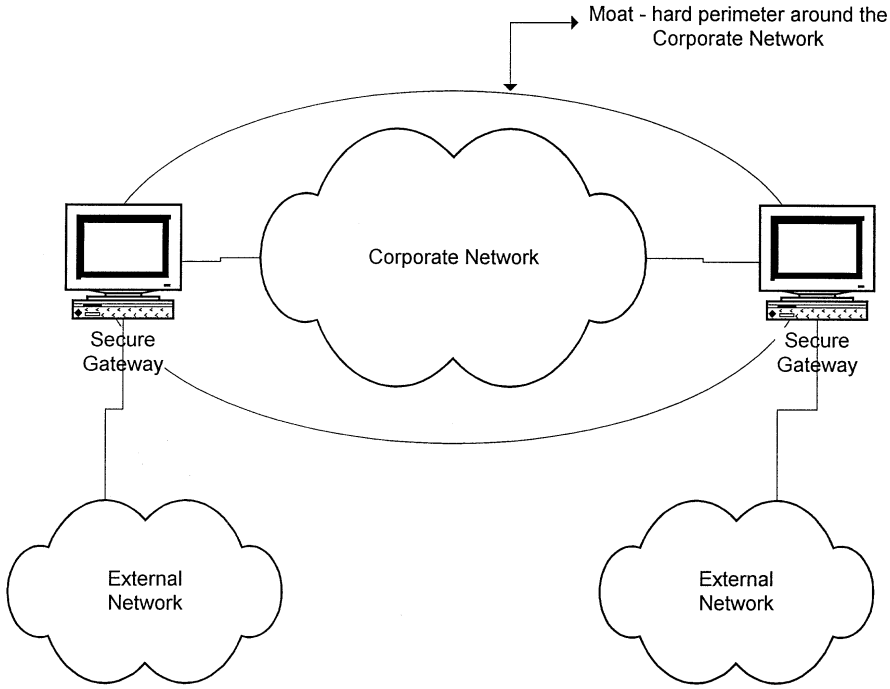


EXHIBIT 131.2 Establishing firewalls at the entry points to the corporate network creates a moat-like effect.

It is, in fact, highly recommended that today's organizations establish a centralized security team that is responsible for the operation of the various security devices. This places responsibility for the operation of that infrastructure on one group that must do the planning, implement the system, and take action to maintain it.

The Threat of Attack

The threat of attack comes from two major directions: attacks based outside the corporate network and attacks based from within. The moat security model, which is working effectively at many organizations, addresses the "attack from without" scenario. Even then, it cannot reliably provide information on the number of attacks, types of attacks, and their points of origin.

However, the moat cannot address the "attack from within" model, as the attack is occurring from within the walls. Consider the castle of medieval times. The moat was constructed to assist in warding off attacks from neighboring hostile forces. However, when fighting breaks out inside the castle walls, the moat offers no value.

The definition of an intrusion attempt is the potential possibility of a deliberate unauthorized attempt to:

- Access information
- Manipulate information
- Render a system unreliable or unusable

However, an attack is a single unauthorized access attempt, or unauthorized use attempt, regardless of success.

Unauthorized Computer Use

The problem is that the existing perimeter does not protect from an attack from within. The major security surveys continually report that the smallest percentage of loss comes from attacks that originate outside the organization. This means that the employees are really the largest threat to the organization.

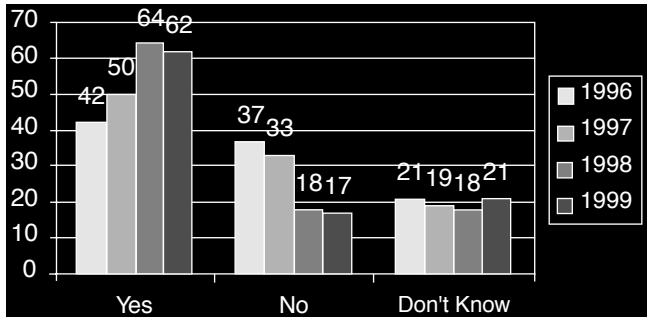


EXHIBIT 131.3 Computer Security Institute 1999 Survey.

The Computer Security Institute conducts an annual survey of its membership in conjunction with the FBI Computer Crime Unit. In the 1999 survey, the question was asked: "Has your organization experienced an incident involving the unauthorized use of a computer system?" (see Exhibit 131.3). As indicated, there was an overwhelming positive response, which had been climbing over the previous three years, but which saw a slight drop in affirmative response. Many organizations could answer "Yes" to this question, but there is also a strong element of "Don't Know." This element is because the only unauthorized use one is aware of is what is ultimately reported or found as a result of some other factor.

The cost of the information loss is staggering, as illustrated in the following information (also from the CSI Survey). From that survey, it is evident that unauthorized insider access and theft of proprietary information has the highest reported cost. Given the potential value of the technical, R&D, marketing, and strategic business information that is available on the network, more and more companies need to focus additional attention to the protection of the data and securing the network.

Financial Losses

The financial impact to organizations continues to add up to staggering figures: a total of over $123 million as reported in the survey (see Exhibit 131.4). The survey identified that there has been an increase in the cost of unauthorized access by insiders, and the cost in other areas has also risen dramatically. The survey also identified that there continues to be an increase in the number of attacks driven from outside the reporting organizations. This is largely due to the increasing sophistication of network attack tools and the number of attackers who are using them.

Intrusion detection and monitoring systems can assist in reducing the "Don't Know" factor by providing a point where unauthorized or undesirable use can be viewed, and appropriate action taken either in real-time or after the fact.

Our Employees Are Against Us

An often-quoted metric is that one of 700 employees is actively working against the company. This means that if an organization has 7000 employees, there are ten employees actively working against the organization's best interests. Although this sounds like a small number of people, the nature of who they are in an organization will dictate what they have access to and can easily use against the company.

A recent American Society for Industrial Security (ASIS, http://www.asisonline.org) "Trends in Intellectual Property Loss" survey suggested that approximately 75 percent of technology losses occur from employees and those with a trusted relationship to the company (i.e., contractors and subcontractors). Computer intrusions involve approximately 87 percent of the insider issue.

Although organizations typically have the perimeter secure, the corporate network is wide open, with all manner of information available to every one who has network access. This includes employees, contractors, suppliers, and customers! How does an organization know that its vital information is not being carried out of the network? The truth is that many do not know, and in many cases it is almost impossible to tell.

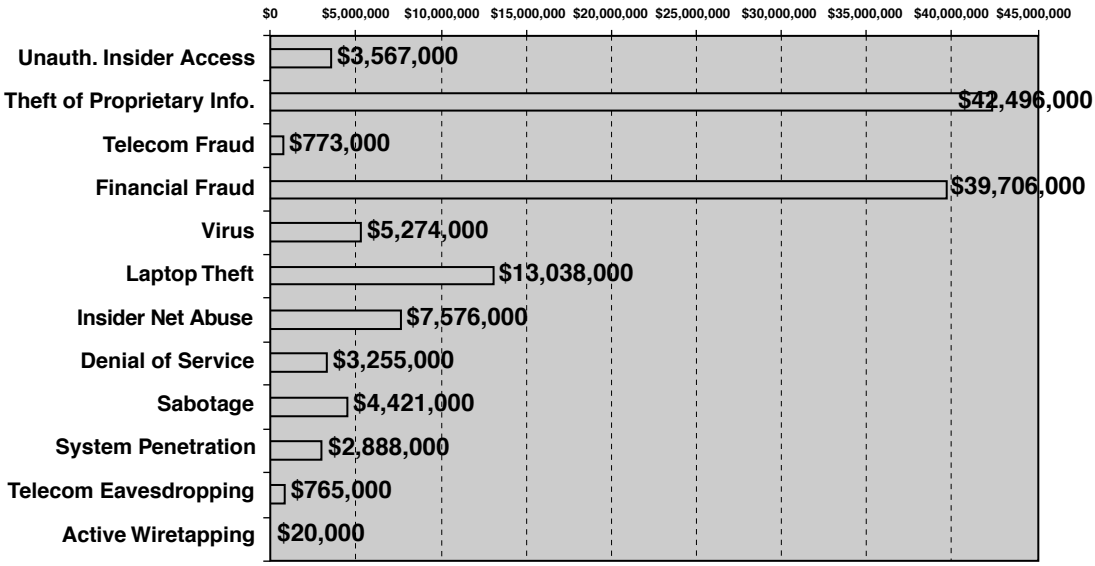


EXHIBIT 131.4 Dollar amount of losses by type.

Where Is the Critical Information?

The other aspect to this is that many organizations do not know where their critical information is stored. This does not even mean where the source code or technical information is stored. That is important, but one's competitors will be building similar products. The critical information is the strategic business plan, bids for new contracts, and financial information. There are various systems in place to control access to various components, but there are problems with the security components in those systems.

Regardless, the strategic business plan will be scattered throughout the corporation on different desktops and laptops. What is the value of that information? Who has it? Where is it going? In the current environment, few organizations can adequately identify the information, let alone where it is stored within the network.

This situation is even worse in government, military, or large corporations where they used to have dozens of filing cabinets to maintain a proper paper trail. Electronic mail has killed the chain of command and the proper establishment of a trail. Information is spread everywhere and important messages simply get deleted when employees leave the company.

The FBI has published a "Top Ten Technology List," which is still current according to the FBI's Awareness of National Security Issues and Response (FBI-ANSIR). This technology list includes:

- Manufacturing processes and technologies
- Information and communication technologies
- Aeronautic and surface transportation systems
- Energy and environmental-related technologies
- Semiconductor materials and microelectronic circuits
- Software engineering
- High-performance computing
- Simulation modeling
- Sensitive radar
- Superconductivity

Many high-tech companies operate within these areas and, as such, are prone to increased incidents of attack and intelligence-gathering operations. Because the primary threat is from internal or authorized users, it becomes necessary to apply security measures within the perimeter.

The Future of Network Security

However, the future of network security is changing. The secure gateway will be an integral part of that for a long time. However, implementation of the secure gateway is not the answer in some circumstances. Furthermore, users may be unwilling to accept the performance and convenience penalties created by the secure gateway.

Secure Gateway Types

There are two major types of secure gateways — packet filters and application proxy systems — and companies choose one or the other for various reasons. This chapter does not seek to address the strengths or weaknesses of either approach, but to explain how they are different.

The packet-filter gateway operates at the network and transport levels, performing some basic checks on the header information contained in the packet (see Exhibit 131.5). This means that the packet examination and transfer happens very fast, but there is no logical break between the internal and external network.

The application proxy provides a clear break between the internal and external networks. This is because the packet must travel farther up the TCP/IP protocol stack and be handled by a proxy (see Exhibit 131.6). The application proxy receives the packet, and then establishes a connection to the remote destination on behalf of the user. This is how a proxy works. It provides a logical break between the two networks, and ensures that no packets from one network are automatically sent to the other network.

The downside is that there must a proxy on the secure gateway for each protocol. Most secure gateway vendors do not provide a toolkit to build application proxies. Consequently, companies are limited in what services can be offered until the appropriate proxy is developed by the vendor.

The third type of firewall that is beginning to gain attention is the adaptive proxy (see Exhibit 131.7). In this model, the gateway can operate as both an application proxy and a packet filter. When the gateway receives a connection, it behaves like an application proxy. The appropriate proxy checks the connection. As discussed earlier, this has an effect on the overhead associated with the gateway. However, once the connection has been "approved" by the gateway, future packets will travel through the packet filter portion, thereby providing a greater level of performance throughput. There is currently only one vendor offering this technology, although it will expand to others in the future.

The adaptive proxy operates in a similar manner to stateful inspection systems, but it has a proxy component.

Whenever a firewall receives a SYN packet initiating a TCP connection, that SYN packet is reviewed against the firewall rule base. Just like a router, this SYN packet is compared to the rules in sequential order (starting

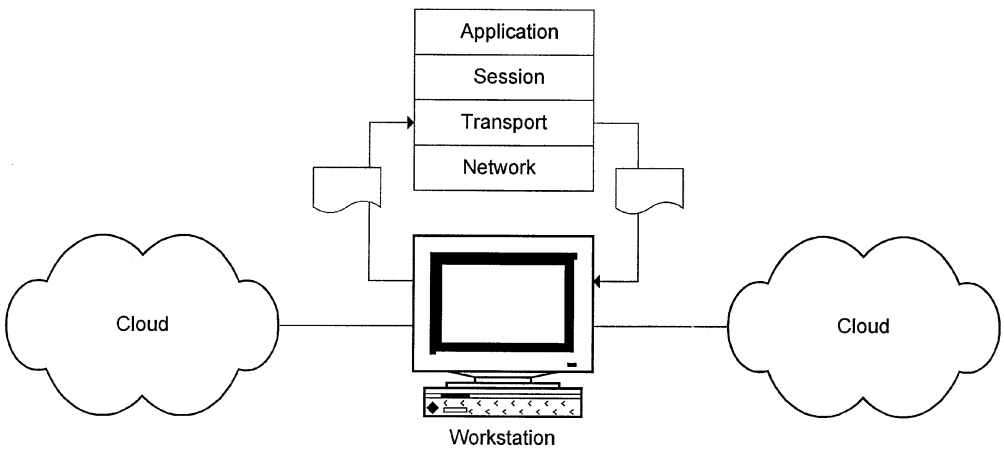


EXHIBIT 131.5 The packet-filter gateway operates at the network and transport levels, performing some basic checks on the header information contained in the packet.

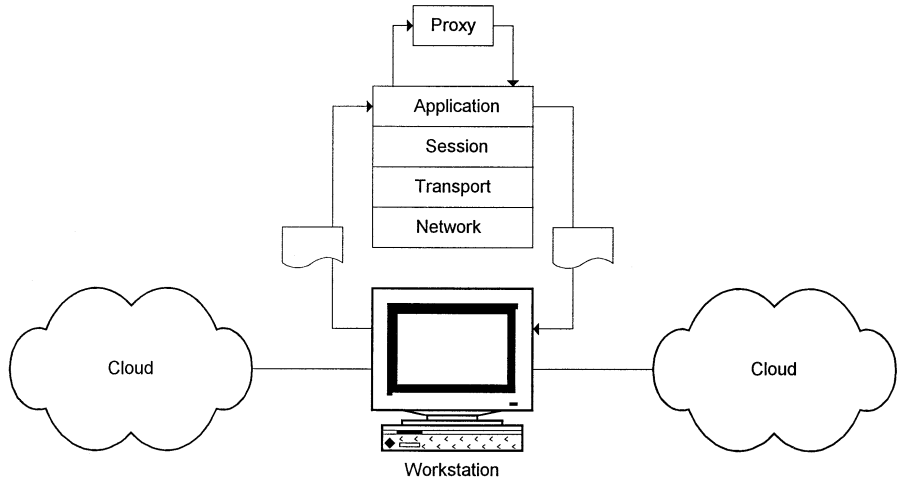


EXHIBIT 131.6 An application proxy provides a clear break between the internal and external network (this is because the packet must travel farther up the TCP/IP protocol stack and be handled by a proxy).

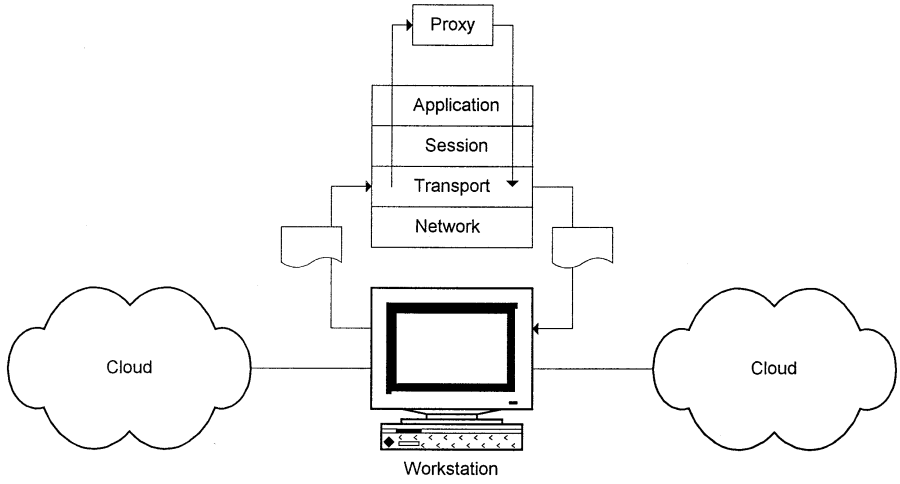


EXHIBIT 131.7 With an adaptive proxy, the gateway can operate as both an application proxy and a packet filter.

with rule 0). If the packet goes through every rule without being accepted, the packet is denied. The connection is then dropped or rejected (RST is sent back to the remote host). However, if the packet is accepted, the session is then entered into the firewall's stateful connection table, which is located in kernel memory. Every packet that follows (that does not have a SYN) is then compared to the stateful inspection table. If the session is in the table and the packet is part of that session, then the packet is accepted. If the packet is not part of the session, then it is dropped. This improves system performance, as every single packet is not compared against the rule base; only SYN packets initiating a connection are compared to the rule base. All other TCP packets are compared to the state table in kernel memory (very fast).

This means that, to provide increased protection for the information within the corporate network, organizations must deploy security controls within the corporate network that consist of both secure gateways (where there is a good reason) and intrusion and network monitoring and detection. Intrusion detection systems are used in a variety of situations.

Security Layering

Security is often layered to provide defense-in-depth. This means that at each layer, there are security controls to ensure that authorized people have access, while still denying access to those who are not authorized (see Exhibit 131.8). As seen in this diagram, this layering can be visualized as a series of concentric circles, with the level of protection increasing to the center.

Layer 1, or the network perimeter, guards against unauthorized access to the network itself. This includes firewalls, remote access servers, etc. Layer 2 is the network. Some information is handled on the network without any thought. As such, layer 2 addresses the protection of the data as it moves across the network. This technology includes link encryptors, VPN, and IPSec. Layer 3 considers access to the server systems themselves. Many users do not need access to the server, but to an application residing there. However, a user who has access to the server may have access to more information than is appropriate for that user. Consequently, layer 3 addresses access and controls on the server itself.

Finally, layer 4 considers the application-level security. Many security problems exist due to inconsistencies in how each application handles or does not handle security. This includes access and authorization for specific functions within that application.

There are occasions where organizations implement good technology in bad ways, which results in poor implementation. This generally leads to a false sense of security and lulls the organization into complacency.

Consequently, by linking each layer, it becomes possible to provide security that the user does not see in some cases, and will have to interact with at a minimal level to provide access to the desired services. This corresponds to the goals of the three-year architecture vision.

Security Goals

Organizations place a great deal of trust in the administrators of computer systems firs to keep things running, and then to make sure that the needed patches are applied whenever possible. It is very important that the

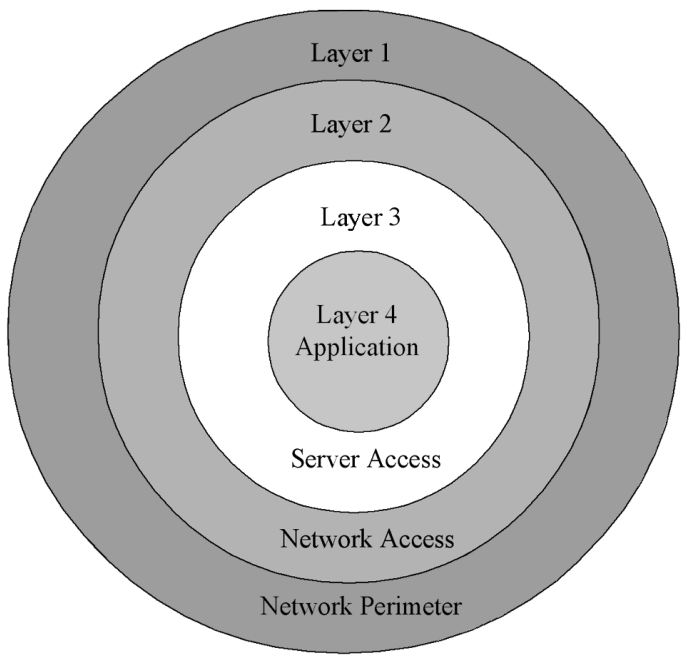


EXHIBIT 131.8 Security layering provides defensive depths (this means that at each layer, there are security controls to ensure that authorized people have access, while still denying access to those who are not authorized).

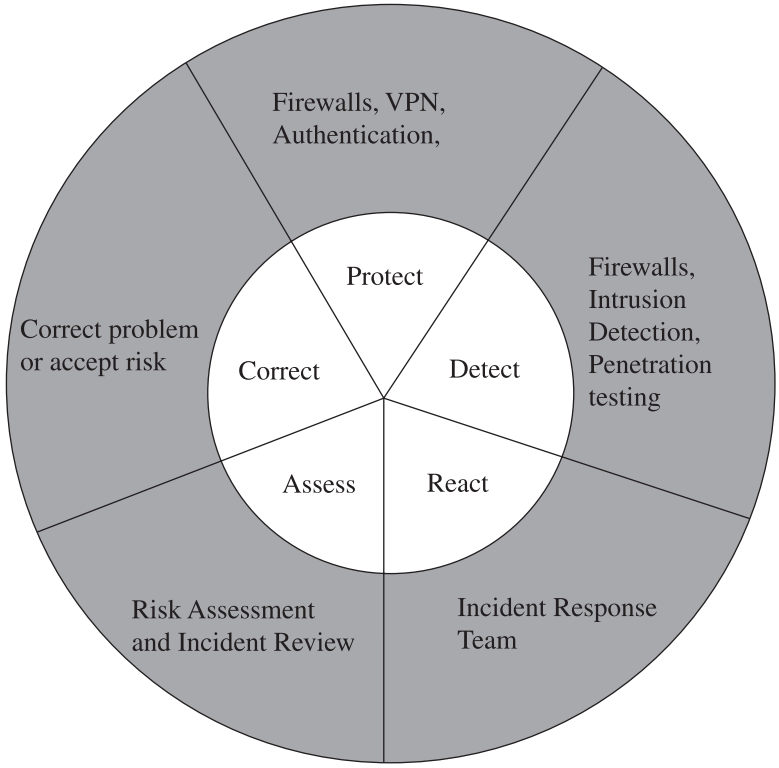


EXHIBIT 131.9 Five essential steps in the information protection arena: protect, detect, react, assess, and correct.

security measures of any system are configured and maintained to prevent unauthorized access. The major threats to information itself are:

- Disclosure, either accidental or intentional (confidentiality)
- Modification (integrity)
- Destruction (availability)

The goal of an information protection program is to maintain the confidentiality, integrity, and availability of information.

Exhibit 131.9 illustrates five essential steps in the information protection arena: protect, detect, react, assess, and correct.

Protection involves establishing appropriate policies procedures and technology implementations to allow for the protection of the corporation's information and technology assets.

Detection is the ability to determine when those assets have been, or are under attack from some source.

To be effective at maintaining the security goals of confidentiality, integrity, and availability, the corporation must be able to react to a detected intrusion or attack. This involves establishing a Computer Security Incident Response Team to review the alarm and act.

With the tactical response complete, the assessment phase reviews the incident and determines the factors that caused it. From there, a risk analysis is performed to determine:

- The risk of future occurrences
- What the available countermeasures are
- A cost/benefit analysis to determine if any of the available countermeasures should be implemented

The correct stage is where the countermeasures or other changes are implemented; or, if the level of risk is determined to be acceptable to the corporation, no action is taken.

Many of today's proactive organizations have the protection side operating well, as it relates to network protection. However, many have no systems in place to protect the internal data and network components.

Likewise, reaction mechanisms may be in place to address and investigate when an incident occurs. This is accomplished by establishing a Computer Incident Response Team to be used when an incident is detected in progress that requires the knowledge of a diverse group of computer and security specialists.

However, for many, their detection abilities are limited, which is the area that intrusion monitoring and detection is aimed at. By improving detection abilities, one can refine both protection strategies and technology, and how one reacts when an incident occurs.

Because today's computer systems must be able to keep information confidential, maintain integrity, and be available when needed, it is highly likely that any expectation of the system being able to completely prevent a security breach is unrealistic.

Types of Intrusion Monitoring and Detection Systems

There are two major types of intrusion detection: host and network based. Host-based products are based on the computer system and look for intrusions into its own environment. These host-based systems are capable of examining their own configuration and reporting changes to that configuration or to critical files that may result in unauthorized access or modification. For example, a product such as tripwire can be considered a host-based intrusion detection system. Changes in the configuration of the system or its files are detected and reported by tripwire and then captured at the next report.

Network-based products are those that are not bound to looking at intrusions on a specific host. Rather, they are looking for specific activity on the network that may be considered malicious. Network-based tools have the ability to find the attack in progress; host-based tools can actually see the changes inside the system. In fact, it is recommended that one runs both types of systems.

There are essentially two types of intrusion detection "engines." These are statistical anomaly detection and pattern-matching detection engines. Statistical engines look at deviation from statistical measurements to detect intrusions and unusual behaviors. The baseline established for the statistical variables is determined by observing "normal" activity and behavior. This requires significant data collection over a period of time to establish this "normal" or expected behavior. Statistical anomaly systems are generally not run in real-time due to the amount of statistical calculations required. Consequently, they are generally run against logs or other collected data.

Statistical anomaly systems offer some advantages. The well-understood realm of statistical analysis techniques is a major strength so long as the underlying assumptions in the data collection and analysis are valid. Statistical techniques also lend themselves better to analysis dealing with time.

However, the underlying assumptions about the data may not be valid, which causes false alarms and erroneous data reported. The tendency to link information from different variables to demonstrate trends may be statistically incorrect, leading to erroneous conclusions. The major challenge to this technique is establishing the baseline of what is considered expected behavior at the monitored site. This is easier if the users work within some predefined parameters. However, it is well-known that the more experienced users are, the less likely they will operate within those parameters.

One drawback to intrusion detection systems is false-positive alarms. A false-positive occurs when the intrusion detection system causes an alarm when no real intrusion exists. This can occur when a pattern or series of packets resemble an attack pattern but are in fact legitimate traffic.

Worth noting is that some of the major issues with statistical engines involve establishing the baseline. For example, how does one know when a user has read too many files?

Pattern-matching systems are more appropriate to run in real- or near-real-time. The concept is to look at the collected packets for a "signature," or activities that match a known vulnerability. For example, a port scan against a monitored system causes an alarm due to the nature of packets being sent. Due to the nature of some of the signatures involved, there is some overlap between the pattern-matching and anomaly detection systems.

The attack patterns provided by the vendors are compiled from CERT advisories, vendor testing, and practical experience. The challenge is for the vendor to create patterns that match a more-general class of intrusion, rather than being specific to a particular attack.

There are pros and cons to both types, but it is recommended that in the development of the tools, both forms be run. This means collecting the packets and analyzing them in near-real-time and collecting the log data from multiple sources to review it with an anomaly system as well.

In a pattern-matching system, the number and types of events that are monitored are constrained to only those items required to match a pattern. This means that if one is interested only in certain types of attacks, then one does not need to monitor for every event. As previously stated, the pattern-matching engine can run faster due to the absence of the floating-point statistical calculations.

However, pattern-matching systems can suffer from scalability issues, depending on the size of the hardware and the number of patterns to match. Even worse is that most vendors do not provide an extensible language to allow the network security administrator to define his own patterns. This makes adding one's own attack signatures a complicated process.

For both systems, neither really has a "learning" model incorporated into it, and certainly none of the commercial intrusion detection systems has a learning component implemented in it.

Why Intrusion Monitoring and Detection?

The incorporation of intrusion monitoring and detection systems provides the corporation with the ability to ensure that:

- *Protected information is not accessed by unauthorized parties; and if it is, there is a clear audit record.* Organizations must identify the location of various types of information and know where the development of protected technologies takes place. With the installation of an intrusion detection system within the corporate network, one can offer protection to that information without the need for a secure gateway. The intrusion detection system can monitor for connection requests that are not permitted and take appropriate action to block the connection. This provides a clear audit record of the connection request and its origination point, as well as preventing the retrieval of the information. There is no impact to the authorized users.
- *The ability to monitor network traffic without impact to the network.* A secure gateway is intrusive: all of the packets must pass through it before they can be transmitted on the remote network. An intrusion monitoring system is passive: it "listens" on the network and takes appropriate action with the packets.
- *Actively respond to attacks on systems.* Many implementations of intrusion monitoring systems have the ability to perform specific actions when an event takes place. Those actions range from notification to a human to automatic reconfiguration of a device and blocking the connection at the network level.
- *Information security organizations understand the attacks being made and can build systems and networks to resist those attacks.* As attacks are made against the organization, reviewing the information captured by the intrusion monitoring system can assist in the development of better tools, practices, and processes to improve the level of information security and decrease the risk of loss.
- *Metrics reporting is provided.* As in any program, the ability to report on the operation of the program through good quality metrics is essential. Most organizations do not know if there has been a successful penetration into their network because they have no good detection methods to determine this.

Implementation Examples

As more and more organizations enter the electronic business (E-biz) forum in full gear, the effective protection of those systems is essential to being able to establish trust with the customer base that will be using them. Monitoring of the activity around those systems will ensure that one responds to any new attacks in an appropriate fashion, and protects that area of the business — both from financial and image perspectives.

Implementing an intrusion monitoring and detection system enables monitoring at specific sites and locations within the network. For example, one should be immediately concerned with Internet access points and the extranets that house so many critical business services on the Internet.

Second, organizations should be working with information owners on the FBI's top-ten list on how to handle corporate strategic information. That venture would involve installing an intrusion monitoring system and identifying the information that people are not allowed to access, and then using that system to log the access attempts and block the network connections to that information.

The following examples are intended to identify some areas where an intrusion monitoring system could be installed and the benefits of each.

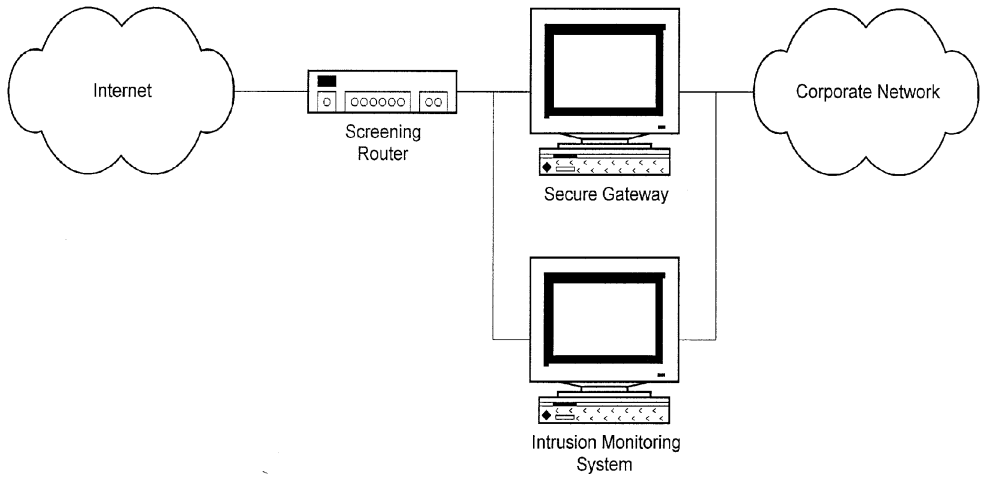


EXHIBIT 131.10 An intrusion monitoring system is configured to monitor the networks on both sides of the fiirewall.

Monitoring at the Secure Gateway

In Exhibit 131.10, the intrusion monitoring system is configured to monitor the networks on both sides of the firewall. The intrusion monitoring system is unable to pass packets itself from one side to the other. This type of implementation uses a passive or nonintrusive mode of network data capture.

To illustrate this, first consider the firewall. The firewall must retransmit packets received on one network to the other network. This is intrusive as the packet is handled by the firewall while in transit. The intrusion monitoring system, on the other hand, does not actually that the packet should be handled. It observes and examines the packet as it is transmitted on the network.

This example also lends itself to monitoring those situations where the traffic must be passed through the secure gateway using a local tunnel. As this provides essentially unrestricted access through the secure gateway, the intrusion monitoring system can offer additional support, and improved logging shows where the packet came from and what it looked like on the other side of the gateway.

Using an intrusion monitoring system in this manner allows metrics collection to support the operation of the perimeter and demonstration that the firewall technology is actually blocking the traffic it was configured to block. In the event of unexpected traffic being passed through anyway, the information provided by the intrusion monitoring system can be used by the appropriate support groups to make the necessary corrections and, if necessary, collect information for law enforcement action.

Monitoring at the Remote Access Service Entry

A second example involves the insertion of an intrusion monitoring device between the RAS access points and their connection to the corporate network (see Exhibit 131.11). In this implementation, the intrusion monitoring system is installed at the remote access point. With the clear realization that most technical and intellectual property loss is through authorized inside access, it makes sense to monitor one's remote access points. It is possible to look for this type of behavior, active attacks against systems, and other misuse of the corporate computing and network services.

Monitoring within the Corporate Network

As mentioned previously, there is no ability to monitor specific subnets within the corporate network where protected information is stored. Through the implementation of intrusion monitoring, it is possible to provide additional protection for that information without the requirement for a secure gateway.

Exhibit 131.12 reveals that the protected servers are on the same subnet as the intrusion monitoring system. When the corporate network user attempts to gain access to the protected servers, the intrusion monitoring

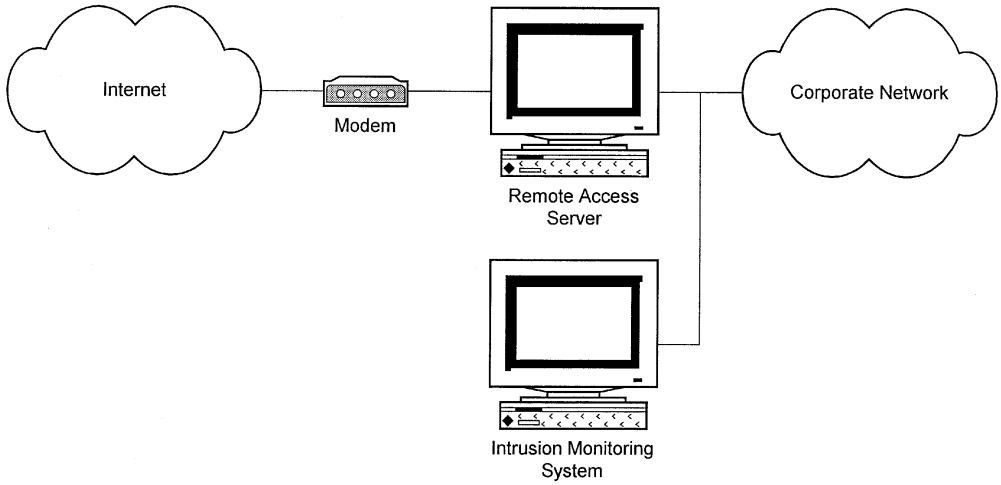


EXHIBIT 131.11 The intrusion monitoring system is installed at the remote access point.

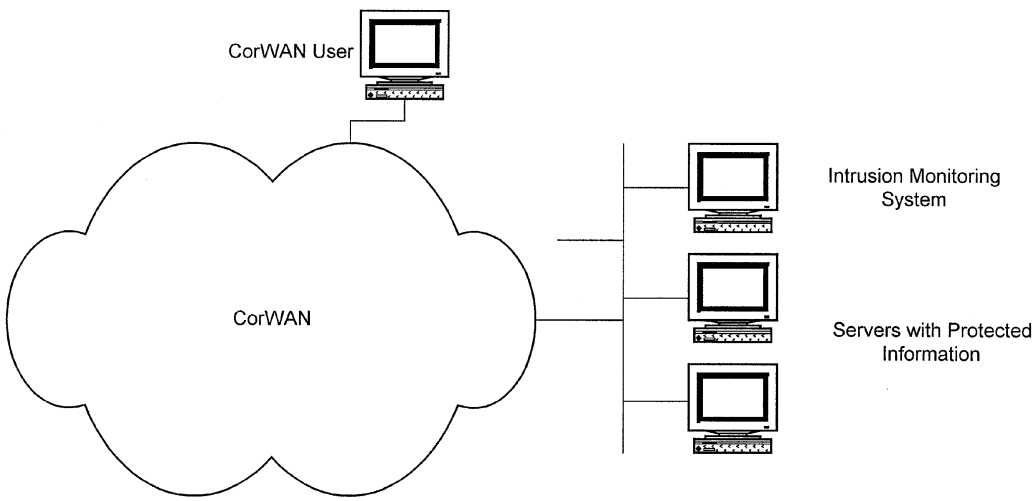


EXHIBIT 131.12 Protected servers are on the same subnet as the intrusion monitoring system.

server can log and, if configured, intercept the connection attempt. This also means that some guidelines on how to determine where to add an intrusion detection system within the corporate network are required. In many organizations, the corporate network is extensive and it may not be feasible to monitor them all.

Monitoring the Extranet

This will facilitate monitoring attacks against externally connected machines or, in the event that a proper extranet has been implemented, by monitoring any attacks against the systems connected to the extranet. However, in this instance, two IDSs may be required to offer detection capabilities for both the extranet and the firewall, as illustrated in Exhibit 131.13.

In this illustration, all activity coming into the extranet is monitored. The extranet itself is also protected as it is not directly on the Internet, but in a private organizationally controlled network. This allows additional controls to be in operation to protect those systems.

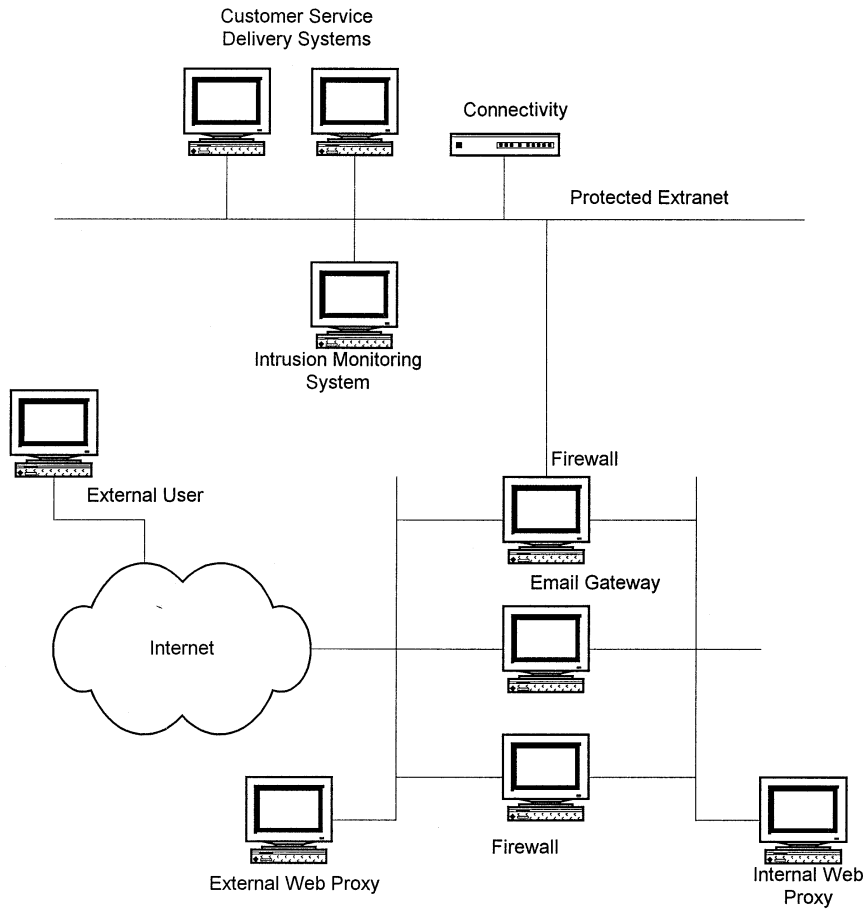


EXHIBIT 131.13 Two IDS systems may be required to offer detection capabilities for both the extranet and the firewall.

Security Is Difficult to Quantify

Security is a business element that is often very difficult to quantify. This is because security is a loss prevention exercise. Until something is missing, most people do not bother with it. However, application of an intrusion monitoring system external to network access points can provide valuable information that includes metrics describing the state of the security perimeter.

Aside from the monitoring component, some intrusion detection systems offer the ability to block network sessions where they are deemed inappropriate or undesirable. These systems offer additional opportunities. Deployment of secure gateways can be problematic as the services that are available to users on the external network are reduced due to limitations at the secure gateway. Using the blocking technology, it may be possible to deploy an intrusion monitoring and detection system to monitor the traffic, but also block connection requests to protected information or sites.

Proactive and Reactive Monitoring

The situations illustrated in Exhibits 131,10 through 131.13 are proactive implementations of an intrusion detection system. The other implementation (not illustrated here) is reactive. A proactive approach calls for the installation and operation of the system in an ongoing mode, as well as ongoing maintenance to ensure that the intrusion monitor is processing information correctly. A reactive mode approach involves having an

intrusion monitor system ready for installation, but not actually using it until some event occurs. The operation of an effective intrusion monitoring systems involves both of these elements.

However, there is the concept of real-time and interval-based intrusion detection. Real-time implies that the monitoring agent is run on a continuous basis; interval-based means that the monitor is run as needed, or at intervals. Vulnerability scanning is also seen as a form of intrusion detection by exposing holes in an operating system configuration. This is interval-based monitoring, as it cannot be done all the time.

Information security organizations are often focused on the prevention aspect of network security. They operate systems that are intended to limit access to information and connectivity. This is a proactive activity that requires ongoing analysis and corrective action to ensure that the network is providing the services it should, and that it is properly protected.

Computer Incident Response Team

The benefits of the intrusion detection system (i.e., the ability to detect undesirable activities) will be lost without the ability to respond to it. This is done most effectively through the operation of a Computer Security Incident Response Team (or CSIRT). Most CSIRT teams are modeled after the Carnegie-Mellon Computer Emergency Response Team.

The object of the CSIRT is to accept alarms from intrusion detection and other sources. Its role is to review the incident and decide if it is a real incident or not.

The CSIRT must include personnel from corporate and information security, internal audit, legal, and human resources departments. Other people may be called in as required, such as network engineering and application providers.

Normally, the alarm is provided to a small group of the CSIRT to evaluate. If it is agreed that there is an incident, then the entire CSIRT is activated. The operation of the CSIRT becomes a full-time responsibility until the issue is resolved. There are a variety of potential responses and issues to be resolved in establishing a CSIRT. These are well covered in other documents and will not be duplicated here.

The CSIRT forms an integral part of the intrusion detection capability by evaluating and responding to the alarms raised by the intrusion detection systems. As such, the personnel involved must have time dedicated to this function; it cannot take a back seat to another project.

Once the tactical response is complete, the CSIRT will closely evaluate the situation and make recommendations for review to prevent or reduce the risk of further occurrence. In the protection cycle, these recommendations are used to assess what further action is to be taken.

This being the case, a decision to implement intrusion detection is a decision to implement and support a CSIRT. Intrusion detection cannot exist without the CSIRT.

Penetration and Compliance Testing

The best method to test security implementation is to try it out. A penetration test simulates the various types of attacks — both internal and external, blind and informed — against the countermeasures of the network. Essentially, a penetration test attempts to gain access through available vulnerabilities.

Penetration testing is part of the detection strategy. Although intrusion detection capabilities are required to monitor access and network status on an ongoing basis, penetration is an interval-based targeted approach to testing both the infrastructure, and the detection and reaction capabilities.

Penetration testing should be done as part of the network security strategy for several purposes:

- *To provide confidence or assurance of systems integrity.* Vulnerability scans often do not include attempts to exploit any vulnerability found, or any of the long list of known vulnerabilities. This is because many of the systems being tested currently are in production. A successful penetration test could seriously affect normal business operations. However, the integrity of the system can be effectively tested in a nonproduction role.
- *To verify the impact of the security program.* Penetration testing is used to determine if the security program is performing as it should. There are a number of different products and services that work together to provide this infrastructure. Each can be evaluated on its own, but it is much more complicated to test them as a system.

- *To provide information that can be used in developing and prioritizing security program initiatives.* Any issues found during a penetration test can alter and affect the direction of the security program priorities. Should a major issue be found that requires correction, the security program goals may be altered to provide a timely resolution for the issue.
- *To proactively discover areas of the infrastructure that may be subject to intrusion or misuse.* People do not install an alarm system in their house and never test it. The same is true here. Ongoing evaluation allows for the identification of components in the infrastructure that may be less secure than desired, not operating as expected, or contain a flaw that can be exploited. Taking a proactive stance means that it becomes possible to find and correct problems before they are exploited.
- *To provide information that can be used in developing and prioritizing policy initiatives.* Policy is not cast in stone; it must be updated from time to time to reflect the changing needs of the business. Penetration tests can assist in the testing and development of policies. This is done using the information learned from the testing to evaluate whether one is compliant with the policies, and if not, which is correct — the implementation or the policy.
- *To assess compliance with standards and policies.* It is essential that the infrastructure, once in operation, be compliant with the relevant security policies and procedures. This verification is achieved through penetration testing, or what is also known as protection testing. Protection testing is the same as penetration testing but with a slightly different objective. Penetration testing attempts to find the vulnerabilities; protection testing proves that the infrastructure is working as expected.
- *To provide metrics that can be used to benchmark the security program.* The ability to demonstrate that the security infrastructure is operating as expected, and that improvement is visible, are important parts of the program. Metrics establish what has been *and* what is now. It is also possible from collected metrics to make "educated guesses" about the future. By collecting metrics, one also gathers data that can be used to benchmark the operation of our infrastructure as compared to other companies.
- *For preimplementation assessments of systems or services.* It is important that appropriate evaluations are performed to ensure that the addition of new services to the infrastructure, or that are dependent on the infrastructure operating correctly, be certified to ensure that no vulnerabilities exist that could be exploited. When a new application is developed that interconnects both internal and external systems, a penetration test against the application and its server is undertaken to verify that neither holds a vulnerability to be exploited. This also ascertains that if the external system is compromised, the attacker cannot gain access to the corporate network resources.

Types of Penetration Tests

There are essentially three major types of penetration testing, each with their own tools and techniques:

- *Level 1 — Zero Knowledge Penetration Testing:* This attempts to penetrate the network from an external source without knowledge of its architecture. However, information that is obtained through publicly accessible information is not excluded.
- *Level 2 — Full Knowledge Penetration Testing:* This attempts to penetrate the network from an external source with full knowledge of the network architecture and software levels.
- *Level 3 — Internal Penetration Testing:* This attempts to compromise network security and hosts from inside one's network.

Penetration testing is interval based, meaning that it is done from time to time and against different target points. Penetration testing is not a real-time activity.

The process consists of collecting information about the network and executing the test. In a Level 1 test, the only information available is what is published through open source information. This includes network broadcasts, upstream Internet service providers, domain name servers, and public registration records. This helps simulate an attack from an unsophisticated intruder who may try various standard approaches. This approach primarily tests one's ability to detect and respond to an attack.

A Level 2 penetration test assumes full knowledge of the hardware and software used on the network. Such information may be available to meticulous and determined intruders using whatever means, including social engineering, to increase their understanding of your network. This stage of the test assumes the worst-possible scenario and calls to light the maximum number of vulnerabilities.

A Level 3 penetration test, or acid test, is an attack from within the network. This is the best judge of the quality of the implementation of a company's security policy. A real attack from within a network can come from various sources, including disgruntled employees, accidental attacks, and brazen intruders who can socially engineer their way into a company.

Penetration testing should be considered very carefully in the implementation of an overall detection program, but it can lead to the negative side effects one is trying to prevent. Therefore, penetration testing should be used cautiously, but still be used to attempt to locate vulnerabilities and to assess the overall operation of the protection program.

Summary

This chapter has presented several implementations of secure gateway and intrusion detection techniques, while focusing on the business impact of their implementation. It is essential that the security professional consider the use of both network- and host-based intrusion detection devices, and balance their use with the potential for impact within the operating environment.

A key point worth remembering is that the implementation of technology is only part of the solution. There must be a well-thought-out strategy and a plan to achieve it.

132

Intelligent Intrusion Analysis: How Thinking Machines Can Recognize Computer Intrusions

Bryan D. Fish, CISSP

Risk management is the essence of information security. The most desirable approach is to avoid risk altogether, or prevent the associated threats from occurring. Preventive measures are important, but they sometimes fail to prevent security incidents. To account for this, it is important for organizations to be able to identify and respond to violations of their security policy. A complete risk mitigation strategy must include detective and corrective measures to supplement preventive measures. This chapter examines an artificial intelligence technique for detecting intrusions.

The knowledge of what constitutes an intrusion is key to distinguishing intrusions from authorized activity. It is difficult to express this knowledge in a way that makes sense to a machine, making intrusion detection a difficult problem to solve with computers. In contrast, most security professionals possess this knowledge tacitly, and are readily able to make such a distinction. The economics of human intrusion analysis are not in our favor, as the sheer capacity of today's information systems would overwhelm even a large staff of analysts. What is needed, then, is a system that combines the knowledge and accuracy of human intrusion analysts with the power and efficiency of the computer.

This chapter explores some artificial intelligence (AI) techniques that show promise as an intrusion detection system. The reader is introduced to the basic concepts of AI, and is then provided with an in-depth examination of one way in which AI techniques are being applied to the problem of intrusion detection. There are three objectives:

1. Motivate AI as a general class of problem-solving techniques.
2. Introduce the reader to basic AI concepts.
3. Explore AI intrusion analysis.

The first objective is addressed by contrasting traditional machine processing with human thought. And the second and third objectives are addressed by discussing existing research into AI-based methods of improving efficiency and accuracy in intrusion detection.

Why Artificial Intelligence?

Human intelligence is one of the most powerful and robust systems on the planet. Over the years, scientists have come to learn a great deal about intelligence, and have discovered striking differences between computers

and the human mind. Computers excel at certain tasks, and humans are quite good at others. Artificial intelligence research seeks to develop ways in which computers can become more proficient in the kinds of tasks that are currently best performed by humans.

For the purposes of understanding intelligence, it is useful to distinguish between three types of tasks: mundane, formal, and expert tasks. In general, the capabilities to perform these tasks build on one another. Expert tasks include tasks such as scientific analysis, engineering design, and medical diagnosis. To perform these tasks, one must first be able to master certain formal tasks, such as basic mathematical and logic operations. Execution of these formal tasks relies on one's ability to perform mundane tasks, such as perception, recognition, and language processing in the given problem space.

To be useful, formal tasks must be executed on a well-defined problem. One uses mundane skills, such as perception and reasoning, to understand and define the problem space. Without the refinement one gains through perceptual skills, formal methods are useless. In short, expert tasks require execution of the appropriate formal methods on problems one has come to understand through the application of mundane skills.

Computers do just that — they compute. They are built to perform simple operations using binary arithmetic with tremendous speed and accuracy. By orchestrating millions of these simple operations in a specific manner, one is able to perform more complex functions on a computer. The human mind, on the other hand, is naturally capable of advanced tasks that are difficult to replicate inside a computer. Computers are simply not good at replicating the capabilities of the human mind. Before exploring the ways in which AI research is closing this gap, take a look at two unique capabilities of the human mind: generalization and learning.

- *Generalization*. Humans are able to generalize concepts that are presented to them, and recognize things by their essence in addition to their specific characteristics. Humans identify the definitive characteristics of an input (an object, situation, concept, feeling, etc.) without having to remember every last bit of detail. Because the human mind allows us to understand the essence of an input, humans learn to understand concepts, not just remember objects. This allows them to recognize instances of a concept that may vary slightly from the original instance they learned to recognize.
- *Learning*. Humans differ from machines in their ability to learn from their experiences. If humans are presented with an object today and told it is a square, they will remember that and identify the same object as a square tomorrow, next week, and next year. The human mind has an enormous capacity for storing thought patterns and concepts. By organizing the information based on the manner in which it is likely to be used, the human mind provides the tremendous capability to recall this stored information when needed. This ability to store and recall thought patterns is known as learning.

The Role of Knowledge

Decades of AI research have demonstrated at least one incontrovertible assertion: intelligence requires knowledge. Knowledge provides context for our perceptual skills and a framework for the application of formal methods in problem-solving. Without knowledge, humans have the capability to execute basic skills over and over, but lack the ability to orchestrate these activities in a manner suggestive of intelligence.

Suppose a recipe calls for two onions. Perceptual skills allow one to recognize onions in the pantry. Formal mathematical skills allow one to determine that there is only one onion, and that one more onion is needed. Deciding that one needs to go to the store and purchase another onion is an expert task (although not a particularly challenging one). All of these basic skills are held together by knowledge. One knows where to look for onions that one already has. One knows that one must count the onions to see how many there are. One knows that one must perform simple subtraction to determine how many more are needed. Without all of these pieces of knowledge, one could not orchestrate the mundane, formal, and expert tasks to solve the problem.

Machines excel at executing formal tasks. Tasks such as mathematics and logic can be formally defined and then executed on a computer with tremendous speed and precision. As it turns out, however, it is quite difficult for a machine to perform the mundane and expert tasks discussed previously. This is due, in large part, to the difficulties associated with representing knowledge in a manner that the computer can understand.

Humans have a remarkable capability for creating, storing, recalling, and applying knowledge. Unfortunately, knowledge is inherently difficult to work with in machine space because it tends to be voluminous, difficult to characterize, and in a constant state of change. Furthermore, human knowledge is organized according to the manner in which it is likely to be used. This differs greatly from computer data, which is organized in a

more structured manner. If one expects machines to solve problems in an intelligent manner, one must arm them with the requisite knowledge and the ability to apply that knowledge. To be useful, that knowledge must exhibit certain characteristics:

- Knowledge must capture generalizations.
- Knowledge must be capable of simple modifications, corrections, and updates.
- Knowledge must be useful in myriad situations, even if it is not complete or totally accurate.
- Knowledge must be able to reduce the vastness of its own space to a subset that is relevant to a given situation.

Knowledge-based systems is a term used to describe problem-solving systems that represent specialized knowledge in a useful manner that meets the above criteria, and provide a means for applying it to solve a problem. Neural network pattern matching is one example of a knowledge-based system. This chapter discusses one use of this technique to represent and apply knowledge in the problem space of computer intrusion detection.

A Pattern-Matching Approach to Intrusion Detection

In applying AI techniques to intrusion detection, the hope is to improve the economics of human analysis. One wants to reduce human involvement in the investigation and response process, as well as reduce the number of false alarms they receive when they do get involved. This can be achieved by improving the accuracy of the intrusion detection system, as measured by the false-positive and false-negative error rates. The false-positive rate is the percentage of false alarms generated by the system. The false-negative rate is the percentage of actual intrusions missed by the system. Developing a system with an attractive false-positive rate reduces the number of incidents that must be investigated by a human. In driving down the false-positive rate, however, one must also take care to maintain an attractive false-negative rate to ensure that one does not fail to detect actual intrusions.

Pattern matching is a logical choice for intrusion detection. One of the most significant challenges in intrusion detection is recognizing new attacks. These attacks may be superficial variations of known techniques, or entirely new methods for breaking into systems. In either case, many traditional intrusion detection systems have trouble recognizing the attack. Pattern matching takes advantage of the power of generalization. Rather than performing an exact feature-wise match between a new input and a known pattern, pattern matching attempts to determine whether an input possesses the "essence" of a known pattern. This allows two entities to match even if they vary by some superficial features.

This chapter section examines a conceptual pattern matching intrusion detection system based on two specific AI techniques. A neural network serves as the brain of the system, storing knowledge about the problem space and applying that knowledge to detect intrusions. A self-organizing map is used to perform correlation on the raw data collected, parsing it into chunks that can be processed by the neural network.

One can begin by introducing some basic concepts of intrusion detection and then move on to a more thorough discussion of these two AI techniques and how they can be used to form an intelligent intrusion analysis system. The conceptual system described here has been developed and tested at the Georgia Tech Research Institute, a division of the Georgia Institute of Technology.

Intrusion Detection

The goal of intrusion detection is to identify activities that violate an organization's security policy. There are essentially two approaches to the intrusion detection problem: misuse detection and anomaly detection. Misuse detection systems define attack signatures — patterns of activity that are known to be undesirable. These systems spend their days monitoring system activity for the presence of these signatures, which indicates an attack. For example, if one sees an IP packet cross an interface with all of the TCP flags turned on, one is probably seeing an XMAS scan and can sound an alarm accordingly.

This approach can be effective, but has several drawbacks. It is a difficult and time-consuming task to create an exhaustive attack signature database. Furthermore, a slight variation of a known attack might differ enough from the predefined signature of that attack to cause the misuse detector to miss the event entirely. Because they look specifically for known attacks, misuse detectors usually have difficulty identifying new attacks for

which a signature does not appear in the database. Misuse detectors tend to have fewer false-positives, but more false-negatives.

Anomaly detection systems are based on a different principle. Anomaly detectors define a model of acceptable system activity and attempt to identify behavior that does not fit that model. Anomaly detectors do not know what specific intrusions look like; rather, they know what normal behavior looks like, and flag deviations from normalcy as potential intrusions. For example, assume that software engineers in a company log on to the system between 7 and 9 A.M. every morning during the week, and log out when they leave between 5 and 6 P.M. Further assume that the software engineers never log in to the systems during the weekend. Suppose one comes to work Monday and notices that all five software engineers logged in to the system at 2 A.M. the previous Sunday morning. This behavior stands out as abnormal, and could be a sign of unauthorized activity. By identifying this anomaly, one has identified a potential intrusion.

Anomaly detection systems are good at certain things, but introduce their own challenges as well. It can be just as difficult to model acceptable behavior (perhaps more so) as it is to model explicitly bad behavior. Anomaly detectors have difficulty adapting to abrupt changes in the way people use the system, which can happen frequently in large environments.

The pattern-matching intrusion detection system described in this chapter follows a misuse detection approach. The idea is to leverage the ability of a neural network to generalize its inputs and recognize superficial variations of that input. By recognizing variations of network-based attacks, the system should avoid many of the false-negatives produced by traditional misuse detectors.

Generalization allows the system to recognize when an attack has been mutated slightly, but remains fundamentally the same as its ancestor. The neural network should be able to recognize a variant that might escape a signature-based system. Furthermore, generalization may allow the system to recognize conditions that are indicative of an attack in general, not just a specific attack. If entirely new attacks exhibit these characteristics, the system may be able to identify them without ever having seen them before.

Neural Networks

Before building this system, take a look at some basic neural network concepts. In moving on to the construction of this intrusion detection system, the following discussion looks at some of these concepts in greater depth and extend their basic functionality.

DaVincian principles of intelligence encourage us to look to analogies in problem-solving. Leonardo observed the way that birds fly in order to better understand how people might one day do the same. So, in striving to evolve the computer into a more powerful and efficient problem-solving machine, one is naturally drawn to the most powerful information processing system known: the human mind. Connectionist AI theory conjectures that the very structure of the human brain facilitates the execution of tasks such as perception, reasoning, and learning. So, the theory goes, if one creates computational models based on the brain metaphor (rather than on the digital computer metaphor), computers can develop a proficiency for some of these human-oriented tasks. The neural network is one such connectionist model. Rather than mimicking the operation of the brain exactly, neural networks derive inspiration from the way the brain works, hoping to achieve some of the same capabilities as the human mind.

Neural networks are composed of two basic components: simple processing elements and weighted connections between these elements. Neural networks are highly parallel systems, as the processing elements operate independently of one another. Thus, control of the network is distributed across its processing elements. The weights between the processing elements in a connectionist model encode the system's knowledge.

Neural networks are particularly useful in pattern-matching problems, in which a given input is matched to a known pattern learned through previous experiences. Furthermore, neural networks have shown a penchant for performing approximate matching, in which incomplete or varied instances of a pattern are still recognized. The type of neural network used here — multilayer backpropagation networks — is quite popular, and is estimated to be in use in a majority of practical applications that use neural networks. These networks have a proven record for pattern-matching problems. Multilayer backpropagation networks are examined in more detail later; the focus here is on their simpler predecessor: the perceptron.

The perceptron is the simplest of neural networks. The perceptron is a network that takes an input vector of binary values, a weight vector of real-valued weights, and computes the cross-product of the two vectors. The result is then applied to a threshold function, which produces a binary output for the perceptron (see Exhibit 132.1).

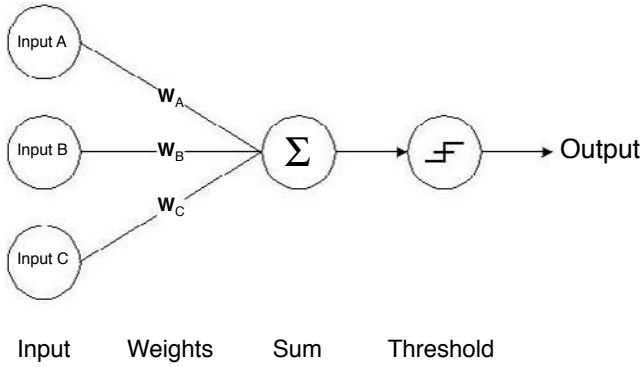


EXHIBIT 132.1 Simple perceptron.

As a pattern-matching system, this network could tell us whether an input matched a single concept, but little more. To form a pattern-matching system capable of distinguishing between several patterns, one can wire multiple processing units to a single input vector. Consider the simple network in Exhibit 132.2 that distinguishes apples, strawberries, and pears. Each processing unit computes a binary value for its corresponding output type (just as the simple perceptron did). The three-element output vector indicates the pattern that the input vector matched. For example, if this network sees a fruit with red skin and white meat (input vector [1,0,0,1]), it will produce the following results in the respective summation processors: Apple 2, Strawberry –2, and Pear –2. The threshold function produces a 1 if the input is positive, a zero otherwise. So, our resulting output vector would be [1,0,0], indicating that the fruit is an apple. Suppose the fruit has red skin and red meat (input vector [1,0,1,0]). The sums would be –2, 2, and –6, respectively. Thus, the output vector would be [0,1,0], indicating that the fruit is a strawberry. This simple network would clearly have trouble in many scenarios (such as recognizing a yellow apple from a pear), but it illustrates the basic concept of perceptron pattern matching.

The knowledge of any neural network is encoded in the weights between its processing elements. In a simple network such as the fruit classifier, it is not manually difficult to manually determine the weights. However, as the networks grow larger — and they must do so to match complex patterns — manual weight determination quickly becomes futile. The power of the connectionist model is that the network learns; it develops its own knowledge through a supervised learning process.

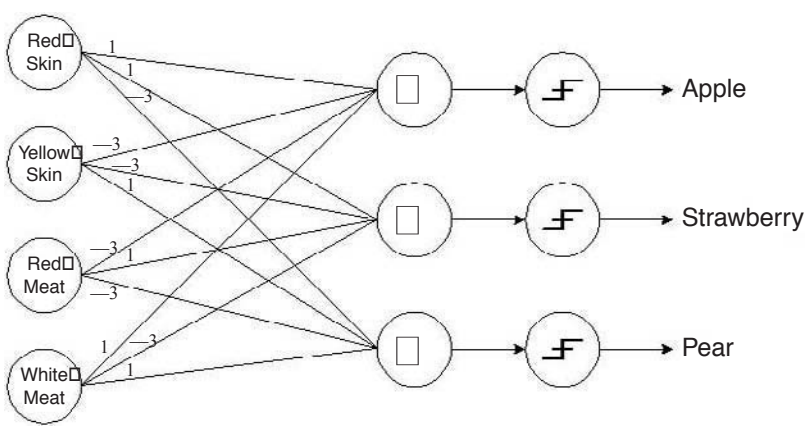


EXHIBIT 132.2 Fruit classification perceptron network.

A Pattern-Matching Intrusion Detection System

In general, the approach to intrusion detection is organized into five phases:

1. *Collect raw data.* For this example, IP packets are used; however, this raw data could be system log entries or any other raw measure of activity in an environment.
2. *Extract data elements from the raw data.* These elements should have meaning, but be basic in nature. In terms of IP packets, data elements might include things such as source and destination address/port, protocol type, flags, and some information about the payload.
3. *Combine selected data elements into a trace.* Related items are collected into a single unit that can be analyzed as a whole. For example, packets within a TCP session could be grouped into a trace.
4. *Evaluate the trace to determine whether it is an attack.* This is where knowledge is applied. Based on human knowledge (or the knowledge of the system), one determines whether the characteristics that indicate an attack are present in the trace being evaluated.
5. *Produce an output.* In this final phase, the system passes judgment on a trace and indicates whether or not it looks like an attack.

This is a generalized approach to detecting intrusions, and most misuse detectors follow a similar methodology. The AI-based approach discussed in this chapter uses this methodology, but applies some advanced techniques along the way with the hope of achieving improved effectiveness. Specifically, the system utilizes a discovery technique known as a self-organizing map to construct traces from data elements, and a pattern-matching technique known as a multilayer backpropagation network to evaluate traces for the presence of an attack. These concepts are presented in more detail later.

This illustration focuses on network-based intrusion detection, but these concepts can be directly applied to other forms of intrusion detection.

Data Gathering and Extraction

The first and second steps of this intrusion detection methodology can usually be accomplished through the application of existing tools and techniques. In the example of IP packets, a network sniffer or promiscuous interface is used to capture packets. A packet decoder can be used to parse and extract data elements from the captured packets. In the case of system logs, a remote logging server can be used to capture all system log entries, and a simple regular expression parser can be used to extract the data elements.

Trace Construction

People are constantly being bombarded with sensory data from many sources. This raw data must be parsed and combined into units on which our minds can operate. Network connections experience a similar phenomenon. They are constantly bombarded with packets with varying sources, destinations, protocols, and options. To use a neural network to recognize attack patterns in network traffic, one must first organize that data into meaningful collections; units of data on which the neural network can operate. This data unit is referred to as a trace.

A self-organizing map (SOM) is one approach to transforming data elements extracted from raw sensory inputs into meaningful clusters on which processing can take place. A SOM is essentially a two-dimensional grid of neuron-like cells. A transform function activates certain cells in the map based on the values present in an input vector. As shown in Exhibit 132.3, the SOM attempts to find correlations between inputs by ensuring that topological neighbors within the map share certain key characteristics from the input vector.

Exhibit 132.3 shows a conceptual representation of a small SOM with two sets of topological neighbors activated. Cells in close proximity to one another (a cluster) are activated when related inputs are presented to the map. A cluster in the map is effectively an index to the input vectors that activated the cells within the cluster. This map shows two clusters, indicating that the input vectors can be logically grouped into two classes. In this intrusion detection system, each of these clusters represents a trace.

The SOM learns to classify related inputs through an unsupervised learning process. In unsupervised learning, the system learns to organize data elements into clusters of related items without any *a priori* knowledge of what those clusters should look like. The network decides on its own how the data elements

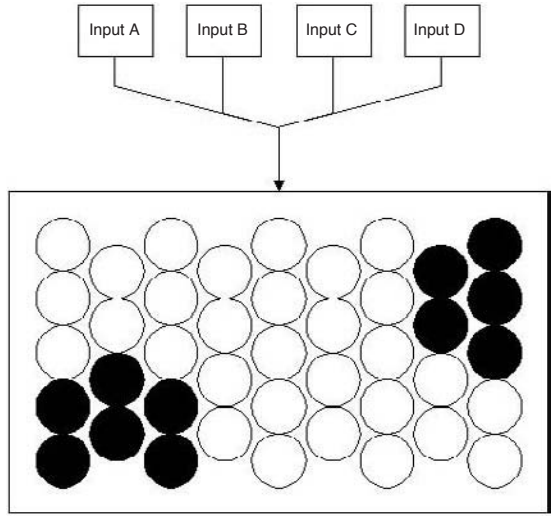


EXHIBIT 132.3 Conceptual SOM activation.

should be grouped. Unsupervised learning is often used as it is here, to discover key features in an input space prior to a supervised learning process.

In SOM learning, the parameters of the transform function are initialized to random values. Each input vector is then presented to the map in sequence. For each input vector, the SOM applies its transform function, which produces a numeric value. That value determines which cells in the map should be activated. The SOM then computes an error function that measures how well the input vectors have been grouped. Based on this result, the SOM adjusts the parameters of the transform function in such a way that would reduce the magnitude of that error function. A small error function indicates a strong correlation between the vectors in a cluster.

The SOM then advances to the next input vector and performs the same operation described above. When all of the input vectors have been processed, one has completed an epoch. The SOM then executes another epoch, processing all input vectors in sequence and adjusting parameters of the transform function accordingly. The correlation between vectors in each cluster improves with every epoch. The SOM continues executing epochs until this correlation reaches a certain predefined threshold. After the learning period concludes, the parameters of the transform function are frozen and the system moves into operational mode.

The output of the SOM is a representation of the data elements indexed by a given cluster. When applied to IP traffic, this representation is a collection of packets, or a trace. This trace becomes the input vector to the pattern-matching system. In addition to applying the trace itself as input to the network, one also computes some basic statistics on the trace (such as average size, packet count, packet frequency, etc.) to feed into the pattern-matching system.

In this system, the SOM produces an output every time the data extracted from a raw IP packet is applied as an input to the map. This produces some interesting temporal analysis capabilities, which are examined momentarily.

Trace Evaluation

The perceptron was previously introduced as a simple pattern-matching system. However, networks composed of simple perceptrons have significant limitations. These networks can only be used on certain types of input spaces that conform to some relatively strict constraints. This is due to the fact that perceptrons can only recognize simple concepts. To form a more robust pattern-matching system, one needs the ability to recognize involved concepts with complex features. This result can be achieved using an extension of the simple perceptron known as a multilayer backpropagation network.

Multilayer networks extend the simple perceptron model by adding another layer of processing units, as depicted in Exhibit 132.4. The layer of hidden processing units is used for complex feature representation.

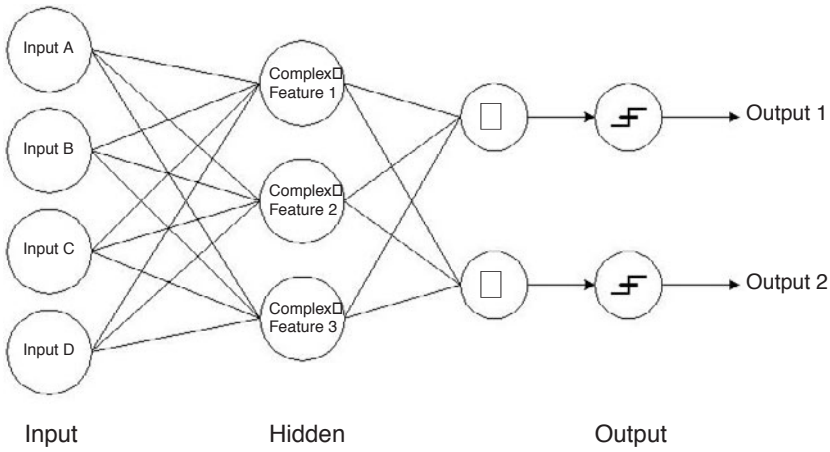


EXHIBIT 132.4 Multilayer back-propagation network.

Each of these hidden units can learn to recognize a single complex feature. A network with multiple hidden units can recognize involved concepts with many complex features.

Learning

Perhaps the most exciting characteristic of neural networks is their capability to learn. The network creates knowledge by developing its own internal representations of key concepts. The power of the neural network is that one does not have to program these concepts into the network; neither does one even have to know what they are. The hidden units start out as a blank slate, and the network is allowed to decide what concepts are key to the overall problem. The network develops its hidden processing units to represent those concepts.

Neural network pattern matchers learn through a supervised learning process. In supervised learning, a series of training inputs and their corresponding correct outputs are presented to the pattern-matching system. This allows the network to learn based on a notion of what the correct answer should be. The network determines the weights that will allow it to correctly match all of the input patterns. If presented with a well-crafted training set, the network can learn to match patterns with tremendous accuracy.

The basic learning algorithm is as follows. All of the weights on the network are initialized to a random value between –0.1 and 0.1. Each input vector is presented to the network in sequence. When presented with an input vector, the network propagates the activations in the input units to the hidden units based on an activation function that produces a real number between 0 and 1. This fuzzy result (as opposed to a strict Boolean 0 or 1 activation) allows one to more accurately reflect the degree to which key features are present in the input vector. Then, activations in the hidden units are propagated to the output units using the same activation function. This entire process is known as feedforward and results in a real number between 0 and 1 in the output elements.

Once the output units have been activated, one can compute an error function between the calculated result and the known correct result. Based on this error, the weights on the network are adjusted in a manner that reduces the magnitude of the error function, and the network moves on to the next input. The weight adjustment process is known as backpropagation.

When all of the input vectors have been processed, an epoch is concluded. The network iterates through as many epochs as it takes to drive the magnitude of the error function down to an acceptable level. Once this process is completed, the network will have learned to recognize the presence of patterns in the input vectors presented to it. As with all neural network learning, the accuracy of the neural network depends solely on the experience it gains on sample patterns during the learning period. Thus, selecting an ample training space is crucial to this process.

In training the intrusion detection system, the system is systematically exposed to both authorized network traffic and to all of the attacks one knows. If the system is trained on only attack traffic, the network would learn to recognize everything it sees as an attack. The converse is true if the system is trained on only authorized traffic. A balance between the two is required to ensure an effective learning process.

Operation

Once the learning process has completed, the weights of the neural network are frozen, and the system is ready for operation. During operation, an input vector (trace output from the SOM clustering map) is loaded into the input nodes of the multilayer backpropagation network. The network propagates the activations in the input units to the hidden units based on the same activation function used in learning. Then, activations in the hidden units are propagated to the output units, just as in the learning process. Once the output units have been activated, one has the result.

Because this result is a real value between 0 and 1, one can take action based not only on the result, but also on the magnitude of the activation. For example, one can apply a threshold function that reports any activation above 0.9 as an attack requiring immediate response, and any activation between 0.75 and 0.89 as an event of interest requiring further investigation.

The System at Work

We will observe how this system evaluates incoming traffic to determine the presence of attack patterns. Exhibit 132.5 is a conceptual illustration of the entire system.

A sniffer is used to capture IP packets. The packet is then decoded, and key data elements are extracted from it. These data elements are packaged as a unit and applied as an input to the self-organizing map. The map clusters this packet with other packets that share key characteristics, and outputs a trace containing the new packet and its topological neighbors. This trace, along with some basic statistics computed on its data, are applied as inputs to the neural network, which propagates activations through the hidden layer to the output layer. Based on the output activation, the trace is identified as an attack, an event of interest, or not an attack.

Detecting a Port Scan

Attackers often perform port scans to identify potential attack targets. Although it does not do any direct damage, one typically treats a port scan as an attack due to its malicious implications. A straightforward port scan is relatively easy to detect: same source address, same destination address, every destination port is tried eventually, etc. However, if the attacker spreads the scan out over time, for example, by probing a single port every few hours, it may be possible to evade the intrusion detection system.

The means by which traces are assembled from data elements produces a unique temporal analysis capability. When packets are added to a cluster, that cluster produces a trace. If clusters are allowed to remain in the SOM for a sufficient amount of time, additional packets will be added to the trace as they are received, although they are spread out over a long period of time. This provides correlation of incoming packets over a long period of time, defeating the slow scan approach to evading an intrusion detection system. The SOM serves as a time-lapse camera, allowing one to correlate events spread out over time into a single trace.

Detecting a SYN Flood

The SYN flood attack has been a particularly popular denial-of-service attack in recent years, and is often used as part of a collaborative attack process. Using two similar traces containing TCP-SYN packets as an example, one can better understand how this system recognizes both an actual SYN flood attack and an apparent SYN flood that is actually just normal Web traffic.

Consider the packet illustrated in Exhibit 132.6. This is a SYN packet, the first packet of a Telnet connection to server. Suppose eight or nine of these packets are seen within one or two seconds of one another, and these packets have:

- The same destination addresses
- A destination port of 23/TCP (Telnet)
- The same source address, with an incrementing source port
- Incrementing sequence and ACK numbers
- The same TCP flags enabled, specifically TCP-SYN

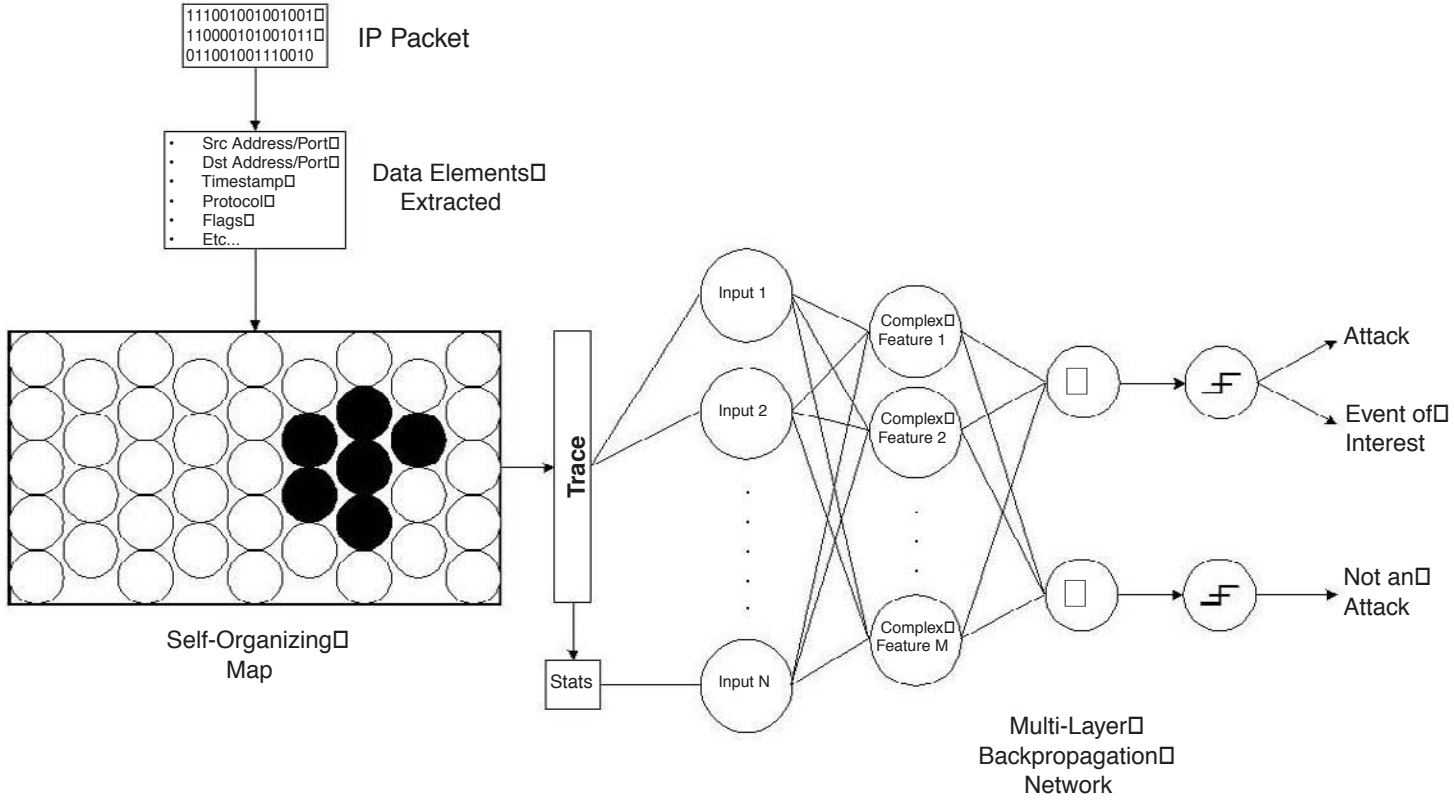


EXHIBIT 132.5 Conceptual layout of pattern-matching intrusion detection system.

EXHIBIT 132.6 SYN Packet

Field	Value
Timestamp	09:30:29.4527
Source Address/Port	attacker:320
Destination Address/Port	server:23/TCP (Telnet)
Flags	S (TCP-SYN)
Sequence Number	1094689872
ACK Number	1094689872

The SOM recognizes that these packets are related to one another and clusters them together into a trace. Each time a similar packet is received, the SOM adds it to the cluster and outputs the updated trace for evaluation by the neural network. Recall that both the trace itself and some basic statistics about the trace (packet count and packet frequency are of interest here) are applied as input to the neural network. In the first few microseconds of this activity, the neural network sees a high frequency of SYN packets, but a relatively low packet count. During the learning process, the neural network learned the packet count and frequency combination that, when associated with the above characteristics, constitutes a SYN flood attack. So, as the SOM clusters more and more of these packets together, all of the above characteristics remain present in the resulting trace, and the packet count and frequency statistics rise. As this happens, the trace pattern begins to match that of a SYN flood attack. When the threshold (defined by an internal representation learned by the neural network during training) is reached, the network produces an output vector indicative of an attack.

The neural network learned to recognize traces that fit the above pattern as an attack by training on actual SYN flood attacks. What happens, then, if one gets a trace that exhibits the following characteristics?

- The same destination addresses
- A destination port of 135/TCP (NetBIOS)
- Varying source address and ports
- The same TCP flags enabled, specifically TCP-SYN

Although the source address and port vary, the neural network is still capable of recognizing this attack as a SYN flood attack. Keep in mind that the system was never explicitly trained on this scenario.

A vector cross-product operation is used to compute node activation functions within the neural network. If the third feature of the original trace is removed, the corresponding element of the input vector would be 0, reducing the result of the cross-product operation. This reduction is proportional to the significance of that feature in the overall description of a SYN flood attack. Because this third feature is not the most prevalent characteristic of the attack, the reduction is relatively small. If all other aspects of the trace remain the same, this reduction may be enough to cause the network to miss the attack.

However, as more packets are received, the packet count will rise, causing an increase in that element of the input vector. This increases the result of the cross-product operation. This increase is proportional to the prevalence of "packet count" as a SYN flood feature. Because this feature is quite significant in the description of a SYN flood, the increase is relatively large, enough to compensate for the reduction caused by the removal of the other feature. This causes the activation function to increase beyond the threshold for alarming an attack.

Now consider another packet, only slightly modified from the original SYN flood example (see Exhibit 132.7). This is a SYN packet, the first packet of an HTTP connection to server. Again, suppose that eight or nine of these packets were seen within one or two seconds of one another, and these packets have:

- The same destination addresses
- A destination port of 80 (HTTP)
- The same source address, with an incrementing source port
- Incrementing sequence and ACK numbers
- The same TCP flags enabled, specifically TCP-SYN

Just as in the previous example, the SOM recognizes that these packets are related to one another and clusters them together into a trace. However, as this trace is presented to the network, it is not flagged as an attack —

EXHIBIT 132.7 SYN Packet

Field	Value
Timestamp	09:30:29.4527
Source Address/Port	attacker: 320
Destination Address/Port	server:80/TCP (HTTP)
Flags	S (TCP-SYN)
Sequence Number	1094689872
ACK Number	1094689872

even as it begins to resemble a SYN flood pattern. The difference between this example and the previous example is the destination port/service. Due to the nature of HTTP, it is normal to see a large number of SYN packets in a sequence such as this. Assuming one has provided the network with HTTP connections as samples of authorized (i.e., nonattack) traffic during the learning process, the network will recognize this distinction and refrain from alarming this activity as a SYN flood. Under the hood of the neural network, this exception to the general SYN flood pattern is represented as a large negative weight from the input node corresponding to a destination port of 80/TCP (or another service likely to trigger false-positive SYN flood alarms, such as 25/TCP or 443/TCP). When that input node is active, this connection strongly inhibits all of the other input features that contribute to a SYN flood activation within the network. Thus, the destination port allows the network to recognize this exception to the general SYN flood pattern.

Extensions to the Concept

This conceptual model can be extended in many ways. A few possibilities are examined here.

It is likely that the system could be extended to deliver not just a ruling on whether the trace is an attack, but also some indication of what kind of attack it appears to be. One of the drawbacks to this approach is that one must provide a great deal of additional information in the training data. This eliminates one of the very advantages of this approach, which was the fact that very little information about the attack training data needed to be provided to the network.

Another extension involves application of a continuous learning model. Rather than freezing the learning process after the network's initial learning period, continuous learning allows the network to periodically adjust its knowledge representation based on its real-world experiences. The goal behind this approach is to allow the system to adapt and learn along with changes in its environment.

One of the challenges with the SOM architecture is that an incoming packet must be assigned to a cluster so that it can be evaluated along with a trace. However, if the SOM misclassifies an incoming packet, one's ability to detect an intrusion may be reduced. Although a network packet may only belong to a single trace, it may fit into several traces until more information is received. An extension of the SOM allows the map to place copies of a given packet into multiple clusters. By doing so, one gives the SOM the opportunity to try to fit a packet into several traces to determine which is a best fit. This introduces a level of fault tolerance into the trace construction scheme.

Challenges and Limitations

In discussing the construction of this conceptual system, the advantages it affords in the intrusion detection game were mentioned. This approach, however, is not without its weaknesses.

Corrupted Learning

Neural networks suffer from tremendous exposure during their learning period. As the network learns to distinguish attacks from authorized behavior, the attacker has an opportunity to create a backdoor of sorts through the neural network. Recall that the network is trained on both attacks and authorized activity. If one does not specifically flag activity as an attack, the system will develop some internal knowledge structure that

allows it to recognize that activity as authorized. The HTTP exception to the SYN flood rule is a good example. The network learns that "if I see certain characteristics in a trace, it is an attack, unless it is destined for port 80/TCP, then it is not an attack." The network learned this exception because authorized HTTP connections were interleaved with the training data. If an attacker were able to surreptitiously insert SYN flood attacks from his network into our training data, the network might learn another exception. This time, the network might learn that "if I see certain characteristics in a trace, it is an attack, unless it is coming from attacker.net." By corrupting the training data, the attacker is able to teach the neural network to allow his attacks to pass, effectively evading the intrusion detection system.

Consider the following, more philosophical problem. Because one derives inspiration from the human mind, the possibility exists that one might introduce the limitations of that model into one's own system. For example, humans can become desensitized over time to certain types of sensory input. People constantly update their knowledge of the world around them. If introduced to small variations on a concept, people update that knowledge to reflect those changes. The cumulative effect of these small variations, when taken over a long period of time, can be dramatic. Cultural desensitization to violence on television is a good example of this. Continuous learning models can cause a gradual shift in the knowledge of a neural network over time. With well-crafted activities, an attacker could contribute to such a shift in knowledge within this conceptual system, effectively desensitizing the network to certain attacks. Over time, the attacker could use this technique to bore a hole through the system. This is similar to a prisoner digging a tunnel out of prison with a spoon.

The Science of Neural Networks

There are several challenges inherent in continuous learning models in a neural network. Continuous learning networks have been known to learn explicit mappings from certain inputs to their corresponding outputs. This is effectively memorization, and eliminates the power of generalization. One way to prevent this from happening is to cease learning once a certain performance level is reached. Another way to avoid this phenomenon is to introduce enough noise into the system to prevent memorization, but not so much as to confuse the network. Finally, one can reduce the number of hidden processing elements capable of storing complex features. This will result in a computational bottleneck that forces the network to learn more compact internal representations of complex features, preventing it from creating an explicit map of every input/output combination. That is, one wants the neural network to be good, but not too good.

There are several challenges inherent in the mathematics of neural network learning. Complex neural networks are often criticized for their slow learning speed. The size of the training space must be several orders of magnitude larger than the size of the network (for fans of complexity theory, the relationship is superlinear). This requires one to present a robust network with an extremely large number of training samples. Because the learning process is iterative by definition, learning can be painfully slow. Researchers have introduced techniques for acceleration that allow the algorithm to proceed naturally for several iterations to settle in on a good learning direction, and then advance progress rapidly in that direction. This has resulted in measurable improvements in neural network learning speeds, but it is still a slow process.

Furthermore, the learning algorithm may settle into a direction that drives the error function to one of several local minima, but not the global minimum. This results in an inaccurate pattern matcher, but it is not intuitive to determine when this has happened. This rarely happens in practice, due in part to the high degree of freedom provided by the high-dimensioned weight space present in most robust networks. However, this is a real problem of which one must be aware.

Practicality

It is not yet clear how well proof-of-concept prototypes will extend from the laboratory to commercial products. Many critics of the practicality of AI in the real world argue that, even if these systems did work, it would require a full-time staff of computer scientists just to keep the system running. Scientific research in this area has made many advances in the past decade, and many of these advances are beginning to find their way into commercial systems. Whether or not AI-based intrusion detection becomes a mainstream technology remains to be seen. The science shows tremendous promise, however; and it may not be wise to dismiss it as impractical just yet. Nevertheless, artificial intelligence is not a panacea, and one must avoid creating a false sense of security stemming from such improvements in technology.

Closing Remarks

As a community of practice, the human collective experience has repeatedly demonstrated that security is a difficult problem. Security problems are rarely black and white; there is almost always a broad spectrum of gray in between. Humans are quite capable of operating within these shades of gray, but computers are deterministic beasts and not readily equipped to do so. The main goal of artificial intelligence work is to enable machines to better solve subjective problems that are currently better suited for humans.

Intrusion detection is one such task. Traditional approaches to intrusion detection are based on the digital computing paradigm. They take advantage of the computer's specialty: executing objective operations with speed and accuracy. These approaches, however, have inherent limitations. Intrusion detection is a fuzzy, subjective task. It is difficult — if not impossible — to fully define that which constitutes an intrusion using the digital computer paradigm.

This chapter has explored an approach to intrusion detection based on the paradigm of the human brain; specifically, a data discovery technique known as a self-organizing map (SOM). A SOM correlates packets in the input space into meaningful traces that one can analyze for intrusions. A neural network pattern matcher analyzes traces for the presence of intrusive activity. These two techniques combine in a conceptual system that approaches intrusion detection in a manner inspired by human thought.

The conceptual system explored in this chapter shows evidence of improved false-negative and false-positive error rates, as observed through the SYN flood example. Systems based on the human brain paradigm show promise as pattern matching intrusion detectors. Perhaps, with continued research in AI, one will see intelligent intrusion analysis systems move from the laboratory into the mainstream.

Bibliography

1. Northcutt, Stephen, *Network Intrusion Detection: An Analyst's Handbook*. Indianapolis: New Riders Publishing, 1999.
2. Rich, Elaine and Knight, Kevin, *Artificial Intelligence,* 2nd ed., New York, McGraw-Hill, 1983.
3. Cannady, James and Mahaffey, James, "The Application of Artificial Neural Networks to Misuse Detection: Initial Results."
4. Frank, Jeremy, "Artificial Intelligence and Intrusion Detection: Current and Future Directions," 1994.
5. Endler, David, "Intrusion Detection: Applying Machine Learning to Solaris Audit Data."

How to Trap the Network Intruder

Jeff Flynn

The job of securing networks is quite difficult. Probably the most significant reason is system complexity. Networks are complicated. They are so complicated no one person can fully comprehend exactly how they work. The models that govern the designs were developed with this concept in mind and provide a layered view of networks that hide the true complexity. This makes it possible for programmers to work on various layers without understanding all the details of the other layers. Of course, programmers on occasion make mistakes, and these mistakes accumulate. Consequently, the Internet we have come to rely on is vulnerable to a wide variety of attacks. Some of the vulnerabilities are well known. Others are known only to a few or are yet to be discovered.

As the Internet grows, so too does the complexity. The growth of the Internet is still accelerating. Every year, more systems are connected to it than were connected the year before. These systems contain increasing amounts of memory. Larger memories allow programmers to develop larger and more complex programs, which provides the programmers with more opportunities to make mistakes. Larger programs also provide intruders with more places to hide malicious code.

Thus, a good network security manager must be very good indeed. The best network security managers may find themselves performing against the unrealistic expectation that they cannot be overwhelmed. These experts must keep up with all the latest attacks and countermeasures. Attackers, on the other hand, need to know only one or a small combination of attacks that will work against their opponents.

A common response to this situation is to simply fix the known problems. This involves closely monitoring reports from organizations such as

CERT or CIAC. As new vulnerabilities are discovered, the system manager responds appropriately. Unfortunately, the list of problems is also growing at an increasing rate. This can be a frustrating experience for the system manager who is forced to fight a losing battle. Likewise, financial managers are caught. They recognize that there are significant risks, yet no investment in safeguards can guarantee immunity from disaster.

It is hard to assess the extent to which tools have improved the situation. The Internet is a highly dynamic environment and does not provide good control samples for making such observations. The common-sense view might be, "However bad it is, it would be worse if we didn't have these devices." Unfortunately, the tools are not always applied properly and can lull management into thinking the situation is under control when it is not. In this situation, there is no benefit. The impact on the intruders is also quite difficult to assess. Serious intruders go to great lengths to keep their identities and approaches secret. Assessing the threat is, hence, a difficult aspect of evaluating the effectiveness of tools.

ASSESSING THE THREAT

There are many ways to gain a perspective on the threat. Most professionals in the field of network security use more than one. Some ways are more subjective than others. Yet there are several popular choices.

Reading

Several written information sources are available on the subject of network security. These include books, technical articles, newspaper articles, trade journal articles, newsgroups, and mailing lists. Each of these mediums has its strengths. Each also has its weaknesses. Trade journal articles, for example, can be biased and may attempt to use fear, uncertainty, and doubt to motivate buyers. Newspaper articles, although less biased, are driven by readership and limited in technical detail. Technical articles are many times too technical, sometimes describing threats that were not threats before publication. The information found in books is quickly dated. Finally, newsgroups and mailing lists, while providing timely information, are transmitted via networks that are subject to the same attacks we are attempting to prevent.

Experimentation

One way to see how difficult it is for someone to break into your system is to attempt to break into it yourself. The Self-Hack Audit, sometimes called Penetration Testing, is a useful means for finding weaknesses and is likely to improve awareness. Similarly, information warfare games provide true insight into how sophisticated intrusions can occur. Still, both of these methods are contrived and do not necessarily represent the actual threat.

Surveys

The 1997 CSI/FBI Computer Crime and Security Survey summarizes the anonymous responses of security professionals from a wide variety of industry segments. Respondents were asked, "If your organization has experienced computer intrusion(s) within the last 12 months, which of the following actions did you take?" Only 29.3% answered that they reported the incident to law enforcement or their own legal counsel. The remainder answered that they did not report the intrusion, or they did their best to "patch security holes." In fact, although 4,899 questionnaires were distributed, only 563 (11.5%) were returned. Of these security professionals, 99 acknowledged detecting "system penetrations," 101 acknowledged detecting "theft of proprietary information," 407 acknowledged detecting viruses, and 338 acknowledged detecting "insider abuse of net access." Security surveys produce statistics that provide managers with useful information for making decisions. Still, many computer incidents go undetected or unreported. This prevents surveys from being as valuable as they would be otherwise.

Firsthand Experience

Human nature seems to dictate that this is the path that most will follow. Firsthand experience occurs, for example, when a person buys a better lock after he detects a burglary. Firsthand experience involves a real threat, but the response comes after the fact. If the initial attack is sufficiently hostile, a response may be of limited use.

There is also a good chance the initial intrusion may go undetected. Network intruders are quite adept at installing back doors. The process is quite simple and may be the first act taken by an attacker after a successful intrusion. Consequently, it is far more difficult to restore security after a network intrusion than it is to prevent an intrusion. Before an individual decides to make firsthand experience his primary approach, he should ask himself, "Is this the kind of experience I want to have?" If the answer is, "I'm willing to take that risk," he should ask himself, "Is it morally responsible for me to make that decision on behalf of all those who may be affected?" What happens on networks can often affect more than the keepers of a network. A 911 emergency system in Florida that was taken down by network intruders provides a compelling example of this fact.

Measuring

Another option for network security managers is to measure the threat. This is critical, because one certainly cannot well manage what one cannot measure. This chapter has two purposes. The first is to suggest that the use of traps can be an effective way to gain a realistic assessment of the threat without exposing individuals and organizations to unreasonable risks. The second is to identify some of the qualities of a "good" trap.

THE BENEFIT OF TRAPS

Traps are attractive for three reasons. First, traps provide real-world information. If designed properly, the activation of the trap is highly correlated to real intrusions. This is not a contrived threat. The intruders detected are real, and they are targeting a particular organization. Second, well-designed traps can provide these measurements safely. Finally, traps can be used to deter future attacks. The trap response to a triggering event is part of the trap design. This goes beyond what intrusion detection systems provide, which may be considered components of traps. There are only three components to a trap: the bait, the trigger, and the snare.

THE QUALITIES OF A "GOOD" TRAP

It is obvious that a good trap is one that actually catches its prey. Good traps share other qualities too.

A Good Trap is Hidden

A hunter would not expect to catch his quarry if he simply left his trap lying on the ground. Animals are too smart or sensitive for this to work. The hunter must hide the trap, perhaps under a pile of leaves. Similarly, hacker traps should be invisible to the network intruder. Of course, one does not need to hide the bait portion of the trap. One only needs to ensure that characteristics of the bait do not betray the presence of the trap. There are many ways to make traps hard to detect. Devices such as in-circuit emulators, SCSI analyzers, and network protocol analyzers can monitor activities without affecting the behavior of the systems being monitored. Alternatively, log information can be transmitted via one-way connections to systems performing real-time intrusion detection functions. In tracking the activities of German hackers, Cliff Stoll transparently monitored modem ports with dramatic results.

A Good Trap Has Attractive Bait

If a trap is to be effective at luring its prey, it must have attractive bait. The trapper has several options in this area, and great care should be used in the selection. Just as a fly fisherman attempts to "match the hatch," the trapper must select a lure that is appropriate for the environment. In some cases, the bait might be a file or directory entitled "ops_planning." In other cases, it might be a file containing the words "security" or "intrusion detection." A continuous indecipherable sequence of bytes transmitted between two hosts may be sufficient. When selecting the bait, the network security manager should consider the possible goals of the intruder. The goals may have much to do with the business of the targeted organization, although this is not necessarily so. If previous intrusions were detected, the network manager might

determine what sort of things the intruder found interesting. Again, care should be taken to prevent the bait from betraying the trap. If it looks too good to be true, the intruder may decide to look elsewhere and thus avoid detection.

A Good Trap Has an Accurate Trigger

A good trap should trap intruders. It should not trap innocent souls who stumble across it in the course of their normal duties. Consequently, the trigger should be designed so that the probability of a false detection is very low. This is extremely important. The loss of trust and the dissension caused by false suspicions or accusations can be considerable. These events can quite possibly cause more damage to an organization than an actual intruder. Of course, real intrusions can result in serious damage too. Hence, if an actual intruder goes for the bait, the probability of detection should be very close to 100 percent. Trap placement can be a useful means to improve the selectivity of a trigger. If the trigger is positioned in a place where no one should legitimately be, false detections can be greatly reduced. Ideally, a trap should be designed so that the intruder has violated a law before he can activate the trigger.

A Good Trap Has a Strong Snare

If a hunter's trap does not have a strong snare, the quarry may simply destroy the device. Animal traps are effective because they are strong enough to hang onto the animal. Similarly, an effective intruder trap should hang onto the intruder. Admittedly, this is one of the most difficult aspects of designing an effective trap.

The identity of an intruder can be known, and the victim organization can have arrest powers. But if the location of the intruder is outside the jurisdiction of that organization, an arrest may not be practical. Currently, the best intruder traps are those that preserve evidence, involve law enforcement, and, in certain circumstances, attempt to bring the intruder into a jurisdiction where action can be taken.

Complicating matters is the hacker modus operandi of weaving (sometimes referred to as looping or hopping) through the Internet. During this process, the hacker may impersonate one or more individuals, systems, or processes. Thus, the path back to the intruder's lair can take many twists and turns. In some cases, the process of following this path might require penetration of a third-party organization's network. Although this is beyond what most would attempt, it is possible that such action could be deemed legal if done with the proper authority.

By way of analogy, one might compare the situation to that of a police officer in "hot pursuit" or acting under "exigent circumstances." If an

officer is in immediate pursuit of a criminal, and that criminal enters a residence, the officer does not wait for someone to grant him access. The officer does not wait for a warrant. He follows the criminal into the residence, breaking the lock on his way if necessary. If that criminal weaves in and out of one property after another, so too will the officer. This process continues until the criminal is apprehended, the criminal is lost, or the pursuit crosses a jurisdictional boundary. In the case of a jurisdictional border crossing, the officer might continue the pursuit, or he could pass the responsibility to another organization according to preexisting agreements between the various parties involved. Unfortunately, the present situation in the Internet is not so well organized. Perhaps, in time, as more laws and law enforcement personnel find their way into the Internet, the situation will improve.

Good Traps Are Used in Combination

To maximize the effectiveness of a trap, the trapper simply needs to add more traps. Just as a good fisherman keeps more than one line in the water, and perhaps more than one lure per line, the trapper should have more than one trap set. A good rule of thumb might be to count the number of targets an organization presents to a would-be intruder. The number of traps that are set should exceed that number. If the traps set are "good," it is more likely that an intruder will be detected than it is a target will be compromised. The approach scales nicely, allowing the trapping organization to select a security stance appropriate for its particular situation.

Good Traps Are Original

Once an intruder becomes aware of a particular type of trap, it is less likely that he can be fooled again in the same way. Hence, good traps should be unique. This is particularly true for the visible bait component of the trap. Other trap components should also be unique. If an intruder suspects a trap, he might try to trigger it from a safe circumstance. Likewise, he may know how to escape from a snare he encountered previously. The less an intruder can surmise about a trap, the better the trap. Originality in design then becomes the hallmark of a good trap. This fact should be viewed as good news for the network security administrator whose job has become an endless loop of applying patches. By developing traps, the network security administrator can have many opportunities to be creative.

Good Traps Do Not Entrap

Trapping and entrapping are two separate things. The difference is in the relation between the trap and the intruder. If the trap somehow induces someone to commit a crime, entrapment occurs, which adversely effects the strength of the trap's snare. Entrapment can prevent prosecution in

many legal systems, which is an important component of an effective snare. Entrapment is also counterproductive. One of the goals of trapping is to deter intruders. Entrapment techniques produce the opposite result by encouraging intrusions. To keep a trap from becoming an entrapping device, the trapper should make the bait invisible to those who have not yet committed a crime. It should be obvious to the intruder and the trapper that a crime has been committed before the bait has the effect of drawing the intruder to the trigger. Notifications and banners should be used to make this point clear. These should indicate the boundaries of legality. Good caveats should include words to the effect that intrusion is not invited or welcome, various laws will be broken by those who proceed without authorization, use of the system implies acknowledgment of this, and use of the system implies consent to monitoring. The name of the organization being protected is not necessary, but a number to contact for clarification should be provided.

When complete, a trap should resemble the situation encountered with silent burglar alarms found in banks. These are traps too. Banks contain such traps, and there is usually no question as to whether entrapment was involved.

PSYCHOLOGY AT WORK

As mentioned previously, one of the benefits of a trap is that it deters. When a hacker realizes that he is in a situation where he is as likely to encounter a trap as he is to obtain his objective, he is likely to slow his pace. When his partners in crime are trapped (i.e., prosecuted), he may consider abandoning the craft. Few things deter more than well-designed traps. Consider the psychological impact on soldiers knowing they are about to cross a minefield. How much slower do they proceed? How much more effective is this deterrent after a mine is detonated?

AN EXAMPLE TRAP

Once network security administrators are aware of the benefits and attributes of good traps, they should consider a working example. Imagine a host set up behind the perimeter of a networked organization. This system is on a network that is protected by banners and other methods (perhaps a firewall). On the host is a file that contains a short list of phone numbers with corresponding passwords. The passwords are long random sequences of alphanumeric characters. These phone numbers and passwords are the bait. To the intruder, they represent additional access. The trigger is a computer (with software) connected to one of these phone numbers. When an intruder attempts to access the trigger with the correct password, the trigger is activated. The probability that the trap was activated by an actual intruder is quite high. The probability that the trap can

be triggered by someone who did not break the rules is quite low. The telephone line is configured with caller ID (CNID) or automatic number identification, so that once triggered, the source of the call can be determined. This information can be used to draft an affidavit that might allow law enforcement to search the premises for the source of the attack. If the intruder was foolish enough to use his own line to make the call, there may be an opportunity for an arrest. If the intruder is not so foolish, at least the designer of the trap is aware that his barrier was penetrated. He does not need to know how it happened for this to be useful information. The mere fact that the intrusion occurred can be enough to justify investigation and additional investment in protective measures. It should be noted that intruders have circumvented CNID systems.

As an alternative to the snare just described, network security administrators could also imagine a trap that might physically capture an intruder, or someone acting on his behalf. By replacing the password bait with an electronic lock combination, a map, and a street address, one might be able to lure an intruder into a holding area disguised as a wiring closet. The use of the correct combination would notify authorities of the intrusion and allow entry. Once inside, the door would lock again and not allow exit. Great care would be required in the planning of such a trap to avoid physical risk to the intruder. Significant liability would result if harm were to come to the prisoner. It would not be reasonable to leave an intruder locked in a closet any significant length of time. Only when the safety of the prisoner can be guaranteed should such a trap be considered. Still, ideas like this may be attractive. In the event an intruder were to fall for this trap, the authorities would not only have a suspect; they would have probable cause for an arrest.

CONCLUSION

The network intruder can be quite clever and may attempt attacks that have not been previously encountered. Techniques are needed for detecting and deterring such intrusions. Although the use of traps will not necessarily free a network security administrator from the burden of simply patching one hole after another, it may help him to focus his efforts in the areas that are most important. It may also give him the well-needed opportunity to be creative. Perhaps the time has come for the network security manager to become more clever than the network intruder.

Intrusion Detection: How to Utilize a Still Immature Technology

E. Eugene Schultz and Eugene Spafford

Defending one's systems and networks is an arduous task indeed. The explosive growth of the Internet combined with the ever-expanding nature of networks makes simply keeping track of change nearly an overwhelming challenge. Add the task of implementing proper security-related controls and the problem becomes of far greater magnitude than even the most visionary experts could have predicted 20 years ago. Although victories here and there in the war against cybercriminals occur, reality echoes the irrefutable truth that "cyberspace" is simply too big a territory to adequately defend. Worse yet, security-related controls that work today will probably fail tomorrow as the perpetrator community develops new ways to defeat these controls. Also, the continuing rush to market software with more new features is resulting in poorly designed and poorly tested software being deployed in critical situations. Thus, the usual installation is based on poorly designed, buggy software that is being used in ways unanticipated by the original designers and that is under continuing attack from all over.

Schultz and Wack (SCHU96) have argued that InfoSec professionals need to avoid relying on an approach that is overly reliant on security-related controls. Determining the controls that most effectively reduce risk from a cost-benefit perspective, then implementing and maintaining those controls is an essential part of the risk management process. Investing all of one's resources in controls is, however, not wise because this strategy does not leave resources for detecting and responding to the security-related incidents that invariably occur. The so-called "fortress mentality"

(implementing security barrier after security barrier but doing nothing else) in the InfoSec arena does not work any better than did castles in the United Kingdom when Oliver Cromwell's armies aimed their cannons at them. It is far better to employ a layered, defense-in-depth strategy that includes protection, monitoring, and response (cf. Garfinkel and Spafford [GARF96, GARF97]).

Merely accepting the viewpoint that it is important to achieve some degree of balance between deploying controls and responding to incidents that occur, unfortunately, does little to improve the effectiveness of an organization's InfoSec practice. An inherent danger in the incident response arena is the implicit assumption that if no incidents surface, all is well. Superficially this assumption seems logical. Studies by the U.S. Defense Information Systems Agency (DISA) in 1993 and again in 1997, however, provide statistics that prove it is badly flawed. Van Wyk (VANW94) found that of nearly 8800 intrusions into Department of Defense systems by a DISA tiger team, only about one in six was detected. Of the detected intrusions, approximately only 4 percent were reported to someone in the chain of command. This meant that of all successful attacks, less than 1 percent were both noticed and reported. A similar study by the same agency 3 years later produced nearly identical results.

One could argue that perhaps many Department of Defense personnel do not have as high a level of technical knowledge as their counterparts in industry because industry (with its traditionally higher salaries) can attract top technical personnel who might more readily be able to more readily recognize the symptoms of attacks. In industry, therefore, according to this line of reasoning, it would be much more likely that some technical "guru" would notice intrusions that occurred. This reasoning is at best only partially true, however, in that in the DISA studies little attempt was made to cover up the intrusions in the first place. In what might be called "more typical" intrusions, in contrast, attackers typically devote a large proportion of their efforts to masquerade the activity they have initiated to avoid being noticed. This is further supported by the latest CSI/FBI survey (POWER99) that indicated that many firms are unable to determine the number or nature of intrusions and losses to their enterprise from IT system attacks, but that losses and number of incidents are continuing to increase.

The main point here is that effective incident response is important and necessary, but it hardly does any good if people do not notice incidents that occur in the first place. Human efforts to notice incidents, as good as they may be, are in many if not most operational settings inadequate. InfoSec professionals often need something more, an automated capability that enables them to be able to discover incidents that are attempted or actually succeed. The solution is intrusion detection. This chapter covers

the topic of intrusion detection, discussing what it is, the types of requirements that apply to intrusion detection systems, and ways that intrusion detection systems can be deployed.

ABOUT INTRUSION DETECTION

What is Intrusion Detection?

Intrusion detection refers to the process of discovering unauthorized use of computers and networks through the use of software designed for this purpose. Intrusion detection software in effect serves a vigilance function. An effective intrusion detection system both discovers and reports unauthorized activity, such as log-on attempts by someone who is not the legitimate user or an account and unauthorized transfer of files to another system. Intrusion detection may also serve a role of helping to document the (attempt at) misuse so as to provide data for strengthening defenses, or for investigation and prosecution after the fact.

Intrusion detection is misnamed. As a field, it started as a form of misuse detection for mainframe systems. The original idea behind automated intrusion detection systems is often credited to James P. Anderson for his 1980 paper on how to use accounting audit files to detect inappropriate use. Over time, systems have become more connected via networks; attention has shifted to penetration of systems by "outsiders," thus including detection of "intrusion" as a goal. Throughout our discussion, we will use the common meaning of "intrusion detection" to include detection of both outsider misuse and insider misuse; users of ID systems should likewise keep in mind that insider misuse must be detected, too.

Why Utilize Intrusion Detection?

One possible approach to intrusion detection would be to deploy thousands of specially trained personnel to continuously monitor systems and networks. This approach would in almost every setting be impossible to implement because it would be impractical. Few organizations would be willing to invest the necessary level of resources and time required to train each "monitor" to obtain the needed technical expertise. Running one or more automated programs, designed effectively to do the same thing but without the involvement of thousands of people, is a more logical approach, provided of course that the program yields acceptable results in detecting unauthorized activity. Additionally, although many people with high levels of technical expertise could be deployed in such a monitoring role, it may not be desirable to do so from another perspective. Even the most elite among the experts might miss certain types of unauthorized actions given the typically gargantuan volume of activity that occurs within

today's systems and networks. A suitable intrusion detection program could thus uncover activity that experts miss.

Detection *per se* is not the only purpose of intrusion detection. Another very important reason to use IDSs is that they often provide a reporting capability. Again, the worst-case scenario would be relying on a substantial number of human beings to gather intrusion data when each person uses a different format to record the data, in addition to using terms and descriptions ambiguous to everyone but that person. Trying to combine each observer's data and descriptions to derive patterns and trends would be virtually impossible; making sense out of any one observer's data would be very challenging. An effective intrusion detection system provides a reporting capability that not only produces human-friendly information displays but also interfaces with a central database or other capability that allows efficient storage, retrieval, and analysis of data.

How IDSs Work

IDSs work in a large variety of ways related to the type of data they capture as well as the types of analysis they perform. At the most elementary level, a program that runs on one or more machines receives audit log data from that machine. The program combs through each entry in the audit logs for signs of unauthorized activity. This type of program is part of a host or system-based IDS. At the other extreme, an IDS may be distributed in nature (MUKH94). Software (normally referred to as agent software) resides in one or more systems connected to a network. Manager software in one particular server receives data from the agents it knows about and analyzes the data (CROS95). This second approach characterizes a network-based IDS (see Exhibit 31.1).

Note that if the data that each agent sends to the manager has not been tampered with, the level of analysis possible is more powerful than with host or system-based IDSs for several reasons:

1. Although a host-based IDS may not depend upon audit data (if it has its own data-capturing service independent of auditing), audit and other types of data produced within single systems are subject to tampering and/or deletion. An attacker who disables auditing and/or an intrusion data collection service on a given machine effectively disables the IDS that runs on that machine. This is not true, however, in the case of a network-based IDS, which can gather data from individual machines and from passive devices (e.g., protocol analyzers) and other, more difficult-to-defeat machines such as firewalls. In other words, network-based IDSs are not as dependent on data from individual systems.

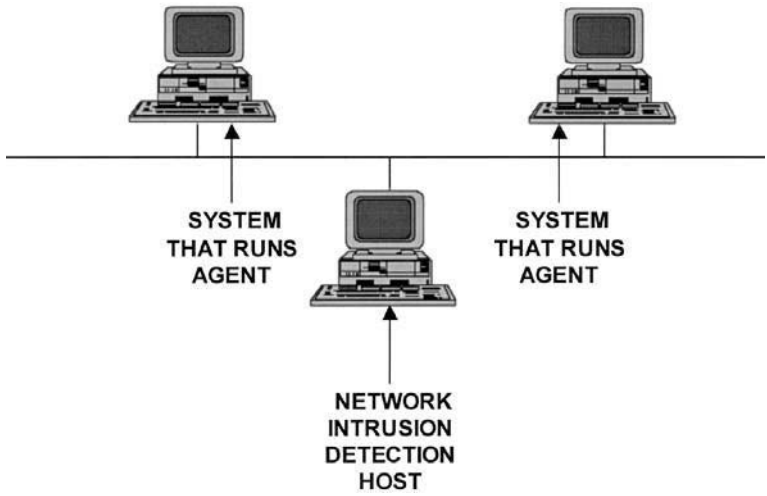


Exhibit 31.1. A Deployment of an IDS in Which Agent Software Running on Hosts Sends Data to a Central Network Intrusion Detection Capability for Analysis

2. Network-based IDSs, furthermore, can utilize data that are not available in system-based IDSs (HERR97). Consider, for example, an attacker who logs on to one system as user "BROWN," then logs on to another system on the same network as "SMITH." The manager software can assign a net ID to each user, thus enabling it to know that the user who has a log-on shell in both systems is the same user. This IDS can then generate an alarm based on the fact that the user in this example has logged on to different accounts with different names. This level of analysis is not possible if an IDS does not have data from multiple machines on the net.

A third form of ID system, currently quite popular, involves one or more systems that observe network traffic (usually at a border location such as near a firewall) and scan for packet traffic that indicates misbehavior. These "network intrusion detection systems" are easy to deploy to protect an enterprise from attack from the outside, but they have the drawback of missing internal behavior that may also be of interest.

APPROACHES TO INTRUSION DETECTION

Not only do different implementations of IDSs work using fundamentally different kinds of data and analysis methods, but they also differ in the types of approaches to intrusion detection that have been incorporated into their design. The correct question here is not "do you want to deploy

an intrusion detection system (IDS)," but rather "which type of IDS do you want to deploy?" The following are the major types of IDSs:

Anomaly Detection Systems

Anomaly Detection Systems are designed to discover anomalous behavior, i.e., behavior that is unexpected and abnormal. At the most elementary level, anomaly detection systems look for use of a computer system during a time of the day or night in which the legitimate user hardly ever uses the computer. Statistical profiles indicating percentiles of measurable behavior and what falls within one standard deviation of the norm, two standard deviations, and so forth are often the basis for determining whether or not a given user action is anomalous. At a more sophisticated level, one might profile variables and processes such as types of usage by each specific user. One user, for example, might access a server mostly to read e-mail; another may balance usage time between e-mail and using spreadsheet-based applications; and a third might mostly write and compile programs. If the first user suddenly starts compiling programs, an anomaly detection system should flag this type of activity as suspicious.

Misuse Detection Systems

The main focus of misuse detection systems is upon symptoms of misuse by authorized users. These symptoms include unauthorized log-ons or bad log-on attempts to systems in addition to abuse of services (e.g., Web-based services, file system mounts, and so on) in which users do not need to authenticate themselves. In the latter case, therefore, good misuse detection systems will identify specific patterns (called "signatures") of anomalous actions. If an anonymous FTP user, for example, repeatedly enters cd .., cd .., cd .. from a command line, there is a good chance that the user is attempting a "dotdot" attack to reach a higher-level directory than FTP access is supposed to allow. It is very unlikely that a legitimate user would repeatedly enter these keystrokes.

Target Monitoring Systems

Target monitoring systems represent a somewhat radical departure from the previously discussed systems in that they do not attempt to discover anomalies or misuse. Instead they report whether certain target objects have been changed; if so, an attack may have occurred. In UNIX systems, for example, attackers often change the /sbin/login program (to cause a pseudo-login to occur in which the password of a user attempting to login is captured and stored in a hidden file) or the /etc/passwd file (which holds names of users, privilege levels, and so on). In Windows NT systems someone may change .DLL (dynamically linked library) files to alter system behavior. Most target monitoring systems use a cryptographic

algorithm to compute a cryptochecksum for each target file. Then if the cryptochecksum is calculated later in time and the new cryptochecksum is different from the previous one, the IDS will report the change. Although this type of IDS superficially does not seem as sophisticated as the previous ones, it has several advantages over anomaly and misuse detection systems:

1. When intruders break into systems, they frequently make changes (sometimes accidentally, sometimes on purpose). Therefore, changed files, executables that are replaced with Trojan Horse versions, and so forth are excellent potential indications that an attack has occurred.
2. Target monitoring systems are not based on statistical norms, signatures, and other indicators that may or may not be valid. These systems are, therefore, not as model-dependent. They are simple and straightforward. Furthermore, they do not really need to be validated because the logic behind them is so obvious.
3. They do not have to be continuously run to be effective. All one has to do is run a target monitoring program at one point in time, then another. Target monitoring systems thus do not generally result in as much performance overhead as do other types of IDSs.

Systems that Perform Wide-Area Correlation of Slow and "Stealth" Probes

Not every attack that occurs is an all-out attack. A fairly typical attack pattern is one in which intruders first probe remote systems and network components such as routers for security-related vulnerabilities, then actually launch attacks later. If attackers were to launch a massive number of probes all at once, the likelihood of noticing the activity would increase dramatically. Many times, therefore, attackers probe one system, then another, then another at deliberately slow time intervals. The result is a substantial reduction in the probability that the probes will be noticed. A fourth type of IDS performs wide-area collection of slow and stealth probes to discover the type of attacks mentioned in this section.

MAJOR ADVANTAGES AND LIMITATION OF INTRUSION DETECTION TECHNOLOGY

Advantages

Intrusion detection is potentially one of the most powerful capabilities that an InfoSec practice can deploy. Much of attackers' ability to perpetrate computer crime and misuse depends on their ability to escape being noticed until it is too late. The implications of the DISA statistics cited earlier are potentially terrifying; in the light of these findings, it might be more

reasonable to ask how an InfoSec practice that claims to observe the principle of "due diligence" could avoid using an IDS enterprise-wide. We strongly assert that any InfoSec practice that does not utilize IDS technology at least to some degree is not practicing due diligence because it will necessarily overlook a large percentage of the incidents that occur. Any practice that remains unaware of incidents does not understand the real risk factor; sadly, it only mimics the behavior of an ostrich with its head in the sand. Simply put, an effective IDS can greatly improve the capability to discover and report security-related incidents.

We also note that the complexity of configuration of most systems and the poor quality of most commercial software effectively guarantees that new flaws will be discovered and widely reported that can be used against most computing environments. Patches and defenses are often not as quickly available as attack tools, and defenses based on monitoring and response are the only way to mitigate such dangers. A failure to use such mechanisms is a failure to adequately provide comprehensive security controls.

In addition to increasing an organization's capability to notice and respond to incidents, intrusion detection systems offer several other major benefits. These include:

1. Cost reduction. Automated capabilities over time generally cost less than humans performing the same function. Once an organization has paid the cost of purchasing and installing one or more IDSs, the cost of an intrusion detection capability can be quite reasonable.
2. Increased detection capability. As mentioned earlier, an effective IDS is able to perform more sophisticated analysis (e.g., by correlating data from a wide range of sources) than are humans. The epitome of the problem of reading and interpreting data through human inspection is reading systems' audit logs. These logs typically produce a volume of data that system administrators seldom have time to inspect, at least in any detail. Remember, too, that attackers often have the initial goal of disabling auditing once they compromise a system's defenses. IDSs do not necessarily rely on audit logs.
3. Deterrent value. Attackers who know intrusion detection capabilities are in place are often more reluctant to continue unauthorized computer-related activity. IDSs thus serve to deter unauthorized activity to some degree.
4. Reporting. An effective IDS incorporates a reporting capability that utilizes standard, easy-to-read and understand formats and database management capabilities.

5. Forensics. A few IDSs incorporate forensics capabilities. Forensics involves the proper handling of evidence that may be used in court. A major goal of forensics is to collect and preserve evidence about computer crime and misuse that will be admissible in a court of law.
6. Failure detection and recovery. Many failures exhibit features similar to misuse or intrusion. Deployment of good IDSs may result in advance notice of these symptoms before they result in full failures. Furthermore, some IDSs can provide audit data about changes, thus allowing failed components to be restored or verified more quickly.

Disadvantages

Intrusion detection is also beset with numerous limitations. Some of the most critical of these drawbacks include:

1. Immaturity. Most (but not all) IDSs available today have significant limitations regarding the quality of functionality they provide. Some are little more than prototypes with a sophisticated user interface. Others purport to compare signatures from a signature library to events that occur in systems and/or networks, but the vendors or developers refuse to allow potential customers to learn how complete and how relevant these libraries are. Equally troubling is the fact that new types of attacks occur all the time; unless someone updates the signature library, detection efficiency will fall. Still other IDSs rely on statistical indicators such as "normal usage patterns" for each user. A clever perpetrator can, however, patiently and continuously engage in activity that does not fall out of the normal range but comes close to doing so. The perpetrator thus can adjust the statistical criteria over time. Someone who normally uses a system between 8 a.m. and 8 p.m. may want to attack the system at midnight. If the perpetrator were to simply attack the system at midnight, alarms might go off because the IDS may not consider midnight usage within the normal range for that user. But if the perpetrator keeps using the system from, say, 11 a.m. to 11 p.m. every day for one week, usage at midnight might no longer be considered statistically deviant.
2. False positives. Another serious limitation of today's IDSs is false positives (Type I errors). A false positive occurs when an IDS signals that an event constitutes a security breach, but that event in reality does not involve such a breach. An example is multiple, failed logins by users who have forgotten their passwords. Most IDS customers today are concerned about false alarms because they are often disruptive and because they sidetrack the people who investigate the false intrusions away from other, legitimately important tasks.

3. Performance decrements. Deploying IDSs results in system and/or network performance hits. The actual amount of decrement depends on the particular IDS; some are very disruptive to performance. Anomaly-based systems are often the most disruptive because of the complexity of matching required.
4. Initial cost. The initial cost of deploying IDSs can be prohibitive. When vendors of IDS products market their products, they often mention only the purchase cost. The cost to deploy these systems may require many hours of consultancy support, resulting in a much higher cost than originally anticipated.
5. Vulnerability to attack. IDSs themselves can be attacked to disable the capabilities they deliver. The most obvious case is when a trusted employee turns off every IDS, engages in a series of illegal actions, then turns every IDS on again. Any attacker can flood a system used by IDS capability with superfluous events to exceed the disk space allocated for the IDS data, thereby causing legitimate data to be overwritten, systems to crash, and a range of other, undesirable outcomes.
6. Applicability. IDSs are designed to uncover intrusions, unauthorized access to systems. Yet a large proportion of the attacks reported during the past year (at the time this chapter was written) were either probes (e.g., use of scanning programs to discover vulnerabilities in systems) or denial-of-service attacks. Suppose that an attacker wants to cause as many systems in an organization's network to crash as possible. Any IDSs in place may not be capable of discovering and reporting many denial-of-service attacks in the first place. Even if they are capable of doing so, knowing that "yes, there was a denial-of-service attack" hardly does any good if the attacked systems are already down! Additionally, many (if not most) of today's IDSs do a far better job of discovering externally initiated attacks than ones that originate from inside. This is unfortunate given that expected loss for insider attacks is far higher than for externally originated attacks.
7. Vulnerability to tampering. IDSs are vulnerable to tampering by unauthorized as well as authorized persons. Many ways to defeat IDSs are widely known within both the InfoSec and perpetrator communities. In a highly entertaining article, Cohen describes 50 of these ways (COHE97).
8. Changing technology. Depending on a particular technology may result in loss of protection as the overall computing infrastructure changes. For instance, network-based intrusion detection is often foiled by switch-based IP networks, ATM-like networks, VPNs, encryption, and alternate routing of messages. All of these technologies are becoming more widely deployed as time goes on.

The advantages and disadvantages of intrusion detection technology are summarized in Exhibit 31.2.

ADVANTAGES	DISADVANTAGES
Cost reduction (at least over time) resulting from automation	Many IDSs do not deliver the functionality that is needed
Increased efficiency in detecting incidents	Unacceptably high false alarm rates
Can deter unauthorized activity	Generally produce performance decrements
Built-in reporting, data management, and other functions	Initial cost may be prohibitive
	May yield superfluous data
Built-in forensics capabilities	IDSs themselves are vulnerable to attack

Exhibit 31.2. Summary of Advantages and Disadvantages of Intrusion Detection Technology

ASSESSING INTRUSION DETECTION REQUIREMENTS

The Relationship of Intrusion Detection to Risk

A large number of organizations go about the process of risk management by periodically performing risk assessments, determining the amount of resources available, then allocating resources according to some method of priority-based risk mitigation strategy, i.e., introducing one or more controls that counter the risk with the greatest potential for negative impact, then implementing one or more measures that address the risk with the second greatest negative impact, and so on until the resources are spent. Regardless of whether or not one agrees with this mode of operation, it tends to guarantee that intrusion detection will be overlooked. In simple terms, intrusion detection does not address any specific risk as directly as measures such as encryption and third-party authentication solutions.

Developing Business-Related Requirements

Developing specific, business-related requirements concerning intrusion detection is anything but an easy process. The difficulty of doing so is, in all likelihood, one of the major detractors in organizations' struggles in dealing with intrusion detection capabilities. Business units, furthermore, may be the most reluctant to utilize intrusion detection technology because of the typical level of resources (personnel and monetary) required and because this technology may superficially seem irrelevant to the needs of fast-paced business units in today's commercial environments.

On the other hand, obtaining buy-in from business units and developing business requirements for intrusion detection at the business unit level is probably not the primary goal anyway. In most organizations if intrusion

detection technology is to be infused successfully, it must be introduced as a central capability. Business requirements and the business rationale for intrusion detection technology are likely to be closely related to the requirements for an organization's audit function. The ultimate goal of intrusion detection technology in business terms is the need to independently evaluate the impact of system and network usage patterns in terms of the organization's financial interests. As such, it is often easiest to put intrusion detection technology in the hands of an organization's audit function.

Decision Criteria

Suppose that your organization decides to introduce intrusion detection technology. After you derive the business requirements that apply to your organization, the next logical step is to determine whether your organization will build a custom IDS or buy a commercial, off-the-shelf version. The latter is generally a much wiser strategy — building a custom IDS generally requires far more time and resources than you might ever imagine. Additionally, maintenance of custom-built IDSs is generally a stumbling block in terms of long-term operations and cost. The exception to the rule is deploying very simple intrusion detection technology. Setting up and deploying "honey pot" servers, for example, is one such strategy. Honey pot servers are alarm servers connected to a local network. Normally nobody uses a honey pot server, but this host is assigned an interesting but bogus name (e.g., patents.corp.com). If anyone logs in or even attempts to login, software in this type of server alerts the administrator, perhaps by having the administrator paged. The major function of honey pot servers is to indicate whether an unauthorized user is "loose on the net" so that one or more individuals can initiate suitable incident response measures. This strategy is not elegant in terms of the intrusion detection capability that it provides, but it is simple and very cost effective. Better yet, an older, reasonably low-ended platform (e.g., a Sparcstation 5) is generally more than sufficient for this type of deployment.

Buying a commercial IDS product is easier when one systematically evaluates the functionality and characteristics of each candidate product against meaningful criteria. We suggest that at a minimum you apply the following criteria:

1. Cost. This includes both short- and long-term costs. As mentioned previously, some products may appear to cost little because their purchase price is low, but life-cycle deployment costs may be intolerable.
2. Functionality. The difference between a system- versus network-based IDS is very important here. Many intrusion detection experts assert that system-based IDSs are better for detecting insider activity, whereas network-based IDSs are better for detecting externally

originated attacks. This consideration is, however, only a beginning point with respect to determining whether or not a product's functionality is suitable. The presence or absence of functions, such as reporting capabilities, data correlation from multiple systems, and near real-time alerting, is also important to consider.

3. Scalability. Each candidate tool should scale not only to business requirements but also to the environments in which it is to be deployed. In general, it is best to assume that whatever product one buys will have to scale upward in time, so obtaining a product that can scale not only to the current environment, but also to more complex environments is frequently a good idea.
4. Degree of automation. The more features of an IDS product that are automated, the less human intervention is necessary.
5. Accuracy. An IDS product should not only identify any *bona fide* intrusion that occurs but should also minimize the false alarm rate.
6. Interoperability. Effective IDSs can interoperate with each other to make data widely available to the various hosts that perform intrusion detection management and database management.
7. Ease of operation. An IDS that is easy to deploy and maintain is more desirable than one that is not.
8. Impact on ongoing operations. An effective IDS causes little disruption in the environment in which it exists.

DEVELOPING AN INTRUSION DETECTION ARCHITECTURE

After requirements are in place and the type of IDS to be used is selected, the next logical phase is to develop an architecture for intrusion detection. In the current context, the term "architecture" is defined as a high-level characterization of how different components within a security practice are organized and how they relate to each focus within that practice. Consider, for example, the components of an InfoSec practice shown in Exhibit 31.3.

To develop an intrusion detection architecture, one should start at the highest level, ensuring that the policies include the appropriate provisions for deploying, managing, and accessing intrusion detection technology. For example, some policy statement should include the provision that no employee or contractor shall access or alter any IDS that is deployed. Another policy statement should specify how much intrusion detection data are to be captured and how they must be archived. It is also important to ensure that an organization's InfoSec policy clearly states what constitutes "unauthorized activity" if the output of IDSs is to have any real meaning.

At the next level down, one might write specific standards appropriate to each type of IDS deployed. For IDSs with signature libraries, for example,

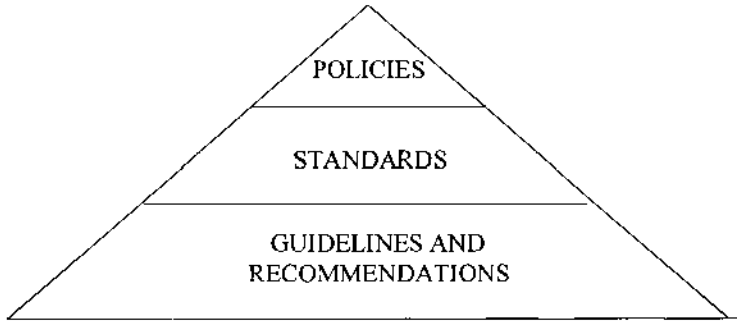


Exhibit 31.3. A Simple Framework for a Security Architecture

it is important to specify how often the libraries should be upgraded. At the lowest level one might include recommendations such as how much disk space to allocate for each particular IDS installation. It is important to realize that an intrusion detection capability does not work well in isolation; it needs to be part of the inner fabric of an organization's culture. As such, developing an intrusion detection architecture is a very important step in successfully deploying intrusion detection technology. Note also that developing such an architecture is not as simple as diagrams such Exhibit 31.3 might imply; it requires carefully analyzing exactly what intrusion detection requires for each component of the architecture and how to embody the solution for each need within that component. Equally important, it requires consensus among organizations that will or may be affected by the rollout of intrusion detection technology in addition to buy-in from senior-level management.

CONCLUSION

We have examined intrusion detection and its potential role in an InfoSec practice, arguing against the "fortress mentality" that results in implementation of security control measures such as password checkers without realizing that no defense measure is 100 percent effective anyway. It is important, therefore, to devote a reasonable portion of an organization's resources to detecting incidents that occur and effectively responding to them. We have taken a look at its advantages and disadvantages, then discussed how one can effectively introduce intrusion detection technology into an organization. Finally, we explained considerations related to deploying IDSs.

Intrusion detection in many ways stands at the same crossroads that firewall technology did nearly a decade ago. The early firewalls were really rather crude and most organizations viewed them as interesting but

impractical. Intrusion detection technology has been available before the first firewall was ever implemented, but the former has always faced more of an uphill battle. The problem can be characterized as due to the mystery and evasiveness that has surrounded IDSs. Firewalls are more straightforward — the simplest firewalls simply block or allow traffic destined for specific hosts. You can be reasonably sure when you buy a firewall product how this product will work. The same has not been true in the intrusion detection arena. Yet at the same time, intrusion detection is rapidly gaining acceptance among major organizations around the world. Although the technology surrounding this area is far less than perfect, it is now sufficiently reliable and sophisticated to warrant its deployment. To ignore and avoid deploying this technology now, in our judgment, constitutes a failure to adopt the types of measures responsible organizations are now putting in place, which in simple terms is a failure to observe "due care" standards.

The good news is that intrusion detection technology is becoming increasingly sophisticated every year. Also encouraging is the fact that performance-related problems associated with IDSs are becoming relatively less important because operating systems and the hardware platforms on which they run are constantly improving with respect to performance characteristics. The research community, additionally, is doing a better job in pioneering the way for the next generation of intrusion detection technology. Some current advances in intrusion detection research include areas such as interoperability of IDSs, automatic reporting, and automated response (in which the IDS takes evasive action when it determines that an attack is in progress).

The bad news is that if your organization does not currently use intrusion detection technology, it is badly behind the intrusion detection "power curve." Consider, furthermore, that an organization that buys, then rolls out a new IDS product is by no means ready to reap the benefits immediately. A definite, steep learning curve for using intrusion detection technology exists. Even if you start deploying this technology now, it takes time to assimilate the mentality of intrusion detection and the technology associated with it into an organization's culture. It is important, therefore, to become familiar with and start using this technology as soon as possible to avoid falling behind even further. The alternative is to continue to function as the proverbial ostrich with its head beneath the sand.

References

COHE97 Cohen, F., Managing network security - Part 14: 50 ways to defeat your intrusion detection system. *Network Security*, December, 1997, pp. 11 – 14.

CROS95 Crosbie, M. and Spafford, E.H., Defending a computer system using autonomous agents. *Proceedings of 18th National Information Systems Security Conference*, 1995, pp. 549 – 558.

GARF96 Garfinkel, S. and Spafford, G., *Practical Unix and Internet Security*, O'Reilly & Associates, inc., 1996.
GARF97 Garfinkel, S. and Spafford, G., *Web Security & Commerce*, O'Reilly & Associates, inc., 1997.
HERR97 Herringshaw, C. Detecting attacks on networks. *IEEE Computer*, 1997, Vol. 30 (12), pp. 16 – 17.
MUKH94 Mukherjee, B., Heberlein, L.T., and Levitt, K.N., Network intrusion detection. *IEEE Network*, 1994, Vol. 8 (3), pp. 26 – 41.
POWER99 Power Richard, Issues and Trends: 1999 CSI/FBI computer crime and security survey, *Computer Security Journal*, Vol. XV, No. 2, Spring 1999
SCHU96 Schultz, E.E. and Wack, J., Responding to computer security incidents, in M. Krause and H.F. Tipton (Eds.), *Handbook of Information Security*. Boston: Auerbach, 1996, pp. 53 – 68.
VANW94 Van Wyk, K.R., Threats to DoD Computer Systems. Paper presented at 23rd Information Integrity Institute Forum. (Cited with author's permission.)

133

Directory Security

Ken Buszta, CISSP

Many organizations have invested in a wide variety of security technologies and appliances to protect their business assets. Some of these projects have taken their toll on the organization's IT budget in the form of time, money, and the number of personnel required to implement and maintain them. Although each of these projects may be critical to an organization's overall security plan, IT managers and administrators continue to overlook one of the most fundamental and cost-effective security practices available — directory and file permission security. This chapter addresses the dilemma created by this issue, the threats it poses, offers potential solutions, and then discusses several operating system utilities that can aid the practitioner in managing permissions.

Understanding the Dilemma

Today, people desire products that are quick to build and even easier to use, and the information technology world is no different. The public's clamor for products that support such buzzwords as *user friendly* and *feature-enriched* has been heard by a majority of the vendors. We can press one button to power-on a computer, automate signing into an operating system, and have a wide variety of services automatically commence when we start up our computers. In the past, reviews referring to these as ease-of-use features have generally led to increased market share and revenues for these vendors. Although the resulting products have addressed the public's request, vendors have failed to address the business requirements for these products, including:

- *Vendors have failed to understand the growing business IT security model: protect the company's assets.* Vendors have created the operating systems with lax permissions on critical operating files and thereby placed the organization's assets at risk. By configuring the operating system permissions to conform to a stricter permission model, we could reduce the amount of time a practitioner spends in a reactive role and increase the time in proactive roles, such as performance management and implementing new technologies that continue to benefit the organization.
- *Vendors fail to warn consumers of the potential pitfalls created by using the default installation configuration.* Operating system file permissions are associated to user and group memberships and are among the largest pitfalls within the default installation. The default configuration permissions are usually excessive for the average user; and as a result, they increase the potential for unauthorized accesses to the system.
- *Vendors fail to address the average user's lack of computer knowledge.* Many engineers work very diligently to fully understand the operating system documentation that arrives with the software. Even with their academic backgrounds and experience, many struggle and are forced to invest in third-party documentation to understand the complex topics. How can vendors then expect the average user to decipher their documentation and configure their systems correctly?

Threats and Consequences

For experienced security practitioners, we understand it is essential to identify all potential threats to an environment and their possible consequences. When we perform a business impact analysis on data, we must

take into consideration two threats that arise from our file and directory permissions — user account privilege escalation and group membership privilege escalation.

User account privileges refer to the granting of permissions to an individual account. Group membership privileges refer to the granting of permissions to a group of individuals. Improperly granted permissions, whether they are overly restrictive or unnecessarily liberal, pose a threat to the organization. The security practitioner recognizes both of these threats as direct conflicts with the principle of least privilege.

The consequences of these threats can be broken into three areas:

1. *Loss of confidentiality.* Much of our data is obtained and maintained through sensitive channels (i.e., customer relationships, trade secrets, and proprietary methodologies). A disgruntled employee with unnecessarily elevated privileges could easily compromise the system's confidentiality. Such a breach could result in a loss of client data, trust, market share, and profits.
2. *Loss of integrity.* Auditing records, whether they are related to the financial, IT, or production environments, are critical for an organization to prove to its shareholders and various government agencies that it is acting with the level of integrity bestowed upon it. Improper permissions could allow for accidental or deliberate data manipulation, including the deletion of critical files.
3. *Loss of availability.* If permissions are too restrictive, authorized users may not be able to access data and programs in a timely manner. However, if permissions are too lenient, a malicious user may manipulate the data or change the permissions of others, rendering the information unavailable to personnel.

Addressing the Threat

Before we can address the threats associated with file and directory permissions, we must address our file system structure. In this context, we are referring to the method utilized in the creation of partitions. File allocation tables (FAT or FAT32), the Microsoft NT File System (NTFS), and Network File Systems (NFS) are examples of the more commonly used file systems. If practitioners are heavily concerned about protecting their electronic assets, they need to be aware of the capabilities of these file systems. Although we can set permissions in a FAT or FAT32 environment, these permissions can be easily bypassed. On the other hand, both NTFS and NFS allow us to establish the owners of files and directories. This ownership allows us to obtain a tighter control on the files and directories. Therefore, InfoSec best practices recommend establishing and maintaining all critical data on non-FAT partitions.

Once we have addressed our file systems, we can address the permission threat. Consider the following scenario. Your team has been charged with creating the administration scheme for all of KTB Corporation's users and the directory and file permissions. KTB has a centralized InfoSec department that provides support to 10,000 end users. Conservative trends have shown that 25 new end users are added daily, and 20 are removed or modified due to terminations or job transitions. The scheme should take into account heavier periods of activity and be managed accordingly. What would be the best way to approach this dilemma?

As stated earlier, operating systems associate files with users and group memberships. This creates two different paths for the practitioner to manage permissions — by users or by groups. After applying some thought to the requirements, part of your team has developed Plan A to administer the permissions strictly with user accounts. In this solution, the practitioner provides the most scrutiny over the permissions because he or she is delegating permissions on an individual case-by-case basis. The team's process includes determining the privileges needed, determining the resources needed, and then assigning permissions to the appropriate users. The plan estimates that with proper documentation, adding users and assigning appropriate permissions will take approximately five minutes, and a deletion or modification will take ten minutes. The additional time for deletions and modifications can be attributed to the research required to ensure all of the user permissions have been removed or changed. Under this plan, our administrator will need a little over five and a half hours of time each day to complete this primary function. This would allow us to utilize the administrator in other proactive roles, such as implementation projects and metric collection.

Another part of your team has developed Plan B. Under this plan, the administrator will use a group membership approach. The team's process for this approach includes determining the privileges needed, determining the resources needed, examining the default groups to determine if they meet the needs, creating custom groups to address the unmet needs, assigning permissions to the appropriate groups, and then providing

groups with the permissions required to perform their tasks. The team estimates that an administrator will spend approximately five minutes configuring each new user and only two minutes removing or modifying user permissions. The difference in the removal times is attributed to having only to remove the user from a group, as opposed to removing the user from each file or directory. Under Plan B, the administrator will need slightly over four hours to perform these primary duties.

Up until now, both plans could be considered acceptable by management. Remember: there was a statement in the scenario about "heavier periods of activity." What happens if the company goes through a growth spurt? How will this affect the availability of the administrator of each plan? On the other hand, what happens if the economy suffered a downturn and KTB was forced to lay off ten percent, or 2000 members, of its workforce? What type of time would be required to fulfill all of the additional tasking? Under Plan A, the administrator would require over 330 tech hours (or over eight weeks) to complete the tasking, while Plan B would require only 67 hours.

As one can see, individual user permissions might work well in a small environment, but not for a growing or large organization. As the number of users increase, the administration of the permissions becomes more labor intensive and sometimes unmanageable. It is easy for a practitioner to become overwhelmed in this scenario.

However, managing through group memberships has demonstrated several benefits. First, it is scalable. As the organization grows, the administrative tasking grows but remains manageable. The second benefit is ease of use. Once we have invested the time to identify our resources and the permissions required to access those resources, the process becomes templated. When someone is hired into the accounts payable department, we can create the new user and then place the user into the accounts payable group. Because the permissions are assigned to the group and not the individual, the user will inherit the permissions of the group throughout the system. Likewise, should we need to terminate an employee, we simply remove that person from the associated group. (*Note:* The author realizes there will be more account maintenance involved, but it is beyond the scope of this discussion.)

The key to remember in this method is for the practitioner to create groups that are based on either roles or rule sets. Users are then matched against these standards and then placed in the appropriate groups. This method requires some planning on the front end by the practitioner; but over time, it will create a more easily managed program than administering by user. When developing your group management plan, remember to adhere to the following procedure:

- Determine the privileges needed.
- Determine the resources needed.
- Examine the default groups to determine if they meet the needs.
- Create custom groups to address unmet needs.
- Assign users to the appropriate groups.
- Give groups the privileges and access necessary to perform their tasks.

Because each network's design is unique to the organization, careful consideration should be given to the use of custom groups. In 1998, Trusted Systems Services, Inc. (TSSI) addressed this very issue in its Windows NT Security Guidelines study for NSA Research. In this study, TSSI recommends alleviating most of the permissions applied to the public (everyone) group except for Read and Execute. TSSI then suggested the formation of the custom group called Installers that would take on all of these stripped permissions. The purpose of this group is to provide the necessary permissions for technicians who were responsible for the installation of new applications. Although this group would not enjoy the privileges of the administrator's group, it is still an excellent example of supporting the principle of least privilege through group memberships.

Establishing Correct Permissions

When establishing the correct permissions, it is important to understand not only the need to correctly identify the permissions at the beginning of the process but also that the process is an ongoing cycle. Regular audits on the permissions should be performed, including at least once a year by an independent party. This will help address any issues related to collusion and help ensure the integrity of the system.

EXHIBIT 133.1 Windows-Based File Permissions

Special Permissions	Full Control	Modify	Read & Execute	Read	Write
Traverse Folder/Execute File	x	x	x		
List Folder/Read Data	x	x	x	x	
Read Attributes	x	x	x	x	
Read Extended Attributes	x	x	x	x	
Create Files/Write Data	x	x			x
Create Folders/Append Data	x	x			x
Write Attributes	x	x			x
Write Extended Attributes	x	x			x
Delete Subfolders and Files	x				
Delete	x	x			
Read Permissions	x	x	x	x	x
Change Permissions	x				
Take Ownership	x				
Synchronize	x	x	x	x	x

Account maintenance is also a piece of the ongoing cycle. Whether an employee is transferred between departments or is terminated, it is essential for the practitioner to ensure that permissions are redefined for the affected user in a timely manner. Failure to act in such a manner could result in serious damage to the organization.

Permissions Settings

For demonstration purposes of this chapter, we examine the permission settings of two of the more popular operating systems — Microsoft and Linux. The practitioner will notice that these permissions apply to the server as well as the client workstations.

Windows-based permissions are divided into two categories — file and directory. The Window-based file permissions include Full Control, Modify, Read & Execute, Read, and Write. Each of these permissions consists of a logical group of special permissions. Exhibit 133.1 lists each file permission and specifies which special permissions are associated with that permission. Note that groups or users granted Full Control on a folder can delete any files in that folder, regardless of the permissions protecting the file.

The Windows-based folder permissions include Full Control, Modify, Read & Execute, List Folder Contents, Read, and Write. Each of these permissions consists of a logical group of special permissions. Exhibit 133.2 lists each folder permission and specifies which special permissions are associated with it. Although List Folder Contents and Read & Execute appear to have the same special permissions, these permissions are inherited differently. List Folder Contents is inherited by folders but not files, and it should only appear when you view folder permissions. Read & Execute is inherited by both files and folders and is always present when you view file or folder permissions.

For the Linux-based operating systems, the file permissions of Read, Write, and Execute are applicable to both the file and directory structures. However, these permissions may be set on three different levels: User ID, Group ID, or the sticky bit. The sticky bit is largely used on publicly writeable directories to ensure that users do not overwrite each other's files.

When the sticky bit is turned on for a directory, users can have read and/or write permissions for that directory; but they can only remove or rename files that they own. The sticky bit on a file tells the operating system that the file will be executed frequently. Only the administrator (root) is permitted to turn the sticky bit on or off. In addition, the sticky bit applies to anyone who accesses the file.

EXHIBIT 133.2 Windows-Based Folder Permissions

Special Permissions	Full Control	Modify	Read & Execute	List Folder Contents	Read	Write
Traverse Folder/Execute File	x	x	x	x		
List Folder/Read Data	x	x	x	x	x	
Read Attributes	x	x	x	x	x	
Read Extended Attributes	x	x	x	x	x	
Create Files/Write Data	x	x				x
Create Folders/Append Data	x	x				x
Write Attributes	x	x				x
Write Extended Attributes	x	x				x
Delete Subfolders and Files	x					
Delete	x	x				
Read Permissions	x	x	x	x	x	x
Change Permissions	x					
Take Ownership	x					
Synchronize	x	x	x	x	x	x

Permission Utilities

To effectively manage permissions, the practitioner should understand the various tools made available to them by the vendors. Both vendors provide a graphical user interface (GUI) and a command line interface (CL). Although there are several high-profile third-party tools available, we will concentrate on the CL utilities provided by the operating system vendors. Exhibit 133.3 lists the various CL tools within the Windows- and Linux-based operating systems. A brief discussion of each utility follows.

You can use *cacls* to display or modify access control lists (ACLs) of files or folders in a Windows-based environment. This includes granting, revoking, and modifying user access rights. If you already have permissions set for multiple users or groups on a folder or file, be careful using the different variables. An improper variable setting will remove all user permissions except for the user and permissions specified on the command line. It is recommended that the practitioner utilize the edit parameter (/e) whenever using this command line utility.

There are several parameters associated with the *calcs* command, and they can be viewed by simply entering *calcs* at the command prompt. The administrator can then view the permissions set for each of the files within the present directory.

The *chmod* command is used to change the permissions mode of a file or directory.

The *chown* command changes the owner of a file specified by the file parameter to the user specified in the owner parameter. The value of the owner parameter can be a user ID or a log-in name found in the password file. Optionally, a group can also be specified. Only the root user can change the owner of a file. You can change

EXHIBIT 133.3 Permission Management Utilities

Utility	Operating Environment
calcs	Windows
chmod	Linux/UNIX
chown	Linux/UNIX
usermod	Linux/UNIX

the group only if you are a root user or own the file. If you own the file but are not a root user, you can change the group only to a group of which you are a member.

Usermod is used to modify a user's log-in definition on the system. It changes the definition of the specified log-in and makes the appropriate log-in-related system file and file system changes.

The *groupmod* command modifies the definition of the specified group by modifying the appropriate entry in the /etc/ group file.

Specific Directory Permissions

As we consider directory permissions, there are three different types of directories — data directories, operating system directories, and application directories. Although the permission standards may differ among each of these directory types, there are two common permission threads shared among all of them — the system administrator group and the system will maintain inclusive permissions to each of them. (*Note*: The administrator's group does not refer to a particular operating system but to a resource level in general. We could easily substitute *root* for the administrator's title.) Because the administrator is responsible for the network, including the resources and data associated with the network, he must maintain the highest permission levels attainable through the permission structure. The *system* refers to the computer and its requirements for carrying out tasking entered by the user. Failure to provide this level of permission to the system could result in the unit crashing and a potential loss of data. Otherwise, unless explicitly stated, all other parties will maintain no permissions in the following discussions.

The data directories may be divided into home directories and shared directories. Home directories provide a place on the network for end users to store data they create or to perform their tasking. These directories should be configured to ensure adequate privacy and confidentiality from other network services. As such, the individual user assigned to the directory shall maintain full control of the directory. If the organization has defined a need for a dedicated user data manager resource, this individual should also have full control of the directory.

Share directories are placed on the network to allow a group of individuals access to a particular set of data. These directories should not be configured with individual permissions but with group permissions. For example, accounts payable data may be kept in a shared directory. A custom group could be created and assigned the appropriate permissions. The user permissions are slightly different from home directories. Instead of providing the appropriate user with full control, it has been recommended to provide the group with Read, Write, Execute, and Delete. This will only allow the group to manipulate the data within the file; they cannot delete the file itself. Additionally, these permissions should be limited to a single directory and not passed along to the subdirectories.

Security is often an after-thought in the actual application design, especially in the proprietary applications designed in-house. As unfortunate as this is, it is still a common practice; and we must be careful to check the directory permissions of any newly installed application — whether it is developed within the organization or purchased from a third party — because users are often given a full set of permissions in the directory structure. Generally, the application users will not need more than read permissions on these directories, unless a data directory has been created within the application directory structure. If this case exists, the data directory should be treated according to the shared data directory permissions previously discussed. Additionally, the installers group should have the ability to implement changes to the directory structure. This would allow them to apply service patches and upgrades to the application.

The third division is the operating system directories. It is critical for the practitioner to have the proper understanding of the operating system directory and file structure before beginning any installation. Failure to understand the potential vulnerabilities, whether they are in the directory structure or elsewhere, will result in a weak link and an opportunity for the E-criminal.

As stated earlier in the chapter, vendors often create default installations to be user friendly. This provides for the most lenient permissions and the largest vulnerabilities to our systems. To minimize the vulnerability, establish read-only permissions for the average user. There will be situations in which these permissions are insufficient, and they should be dealt with on a case-by-case basis. Personnel who provide desktop and server support may fall into this category. In this case, create a custom group to support the specific activities and assign permissions equivalent to read and add. Additionally, all operating system directories should be owned by the administrator only. This will limit the amount of damage an E-criminal could cause to the system.

Sensitive File Permissions

Until now, we have only looked at the directory permissions. Although this approach addresses many concerns, it is only half of our battle. Several different file types within a directory require special consideration based on their roles. The particular file types are executable/binary compiled, print drivers, scripting files, and help files.

Executable/binary files are dangerous because they direct the system or application to perform certain actions. Examples of these file extensions are DLL, EXE, BAT, and BIN. The average user should be restricted to read and execute permissions. They should not have the ability to modify these files.

Print drivers are often run with a full permission set. Manipulation of these files could allow the installation of a malicious program that runs at the elevated privilege. The average user should be limited to a read and execute permission set.

Improperly set permissions on scripting files, such as Java and ActiveX, could allow for two potential problems. By providing the elevated privileges on these files, the user has the ability to modify these files to place a call to run a malicious program or promote program masquerading. Program masquerading is the act of having one program run under the pretext that it is actually another program. For these reasons, these files should also have a read and execute permission set.

Help files often contain executable code. To prevent program masquerading and other spoofing opportunities, these files should not be writeable.

Monitoring and Alerts

After we have planned and implemented our permission infrastructure, we will need to establish a methodology to monitor and audit the infrastructure. This is key to ensuring that unauthorized changes are identified in a timely manner and to limit the potential damage that can be done to our networks. This process will also take careful planning and administration.

The practitioner could implement a strategy that would encompass all of the permissions, but such a strategy would become time-consuming and ineffective. The more effective approach would be to identify the directories and files that are critical to business operations. Particular attention should be given to sensitive information, executables that run critical business processes, and system-related tools.

While designing the monitoring process, practitioners should be keenly aware of how they will be notified in the event a monitoring alarm is activated and what type of actions will be taken. As a minimum, a log entry should be created for each triggered event. Additionally, a mechanism should be in place to notify the appropriate personnel of these events. The mechanism may be in the form of an e-mail, pager alert, or telephone call. Unfortunately, not all operating systems have these features built in; so the practitioner may need to invest in a third-party product. Depending on the nature of the organization's business, the practitioner may consider outsourcing this role to a managed services partner. These partnerships are designed to quickly identify a problem area for the client and implement a response in a very short period.

Once a response has been mounted to an alert, it is also important for the team to review the events leading up to the alert and attempt to minimize the event's recurrence. One can take three definitive actions because of these reviews:

1. *Review the present standards and make changes accordingly.* If we remember that security is a business enabler and not a disabler, we understand that security must be flexible. Our ideal strategy may need slight modifications to support the business model. Such changes should be documented for all parties to review and approve and to provide a paper trail to help restore the system in the event of a catastrophic failure.
2. *Educate the affected parties.* Often, personnel may make changes to the system without notifying everyone. Of course, those who were not notified are the ones affected by the changes. The practitioner may avoid a repeat of the same event by educating the users on why a particular practice is in place.
3. *Escalate the issue.* Sometimes, neither educating users nor modifying standards is the correct solution. The network may be under siege either from an internal or external source, and it is the practitioner's duty to escalate these issues to upper management and possibly law enforcement officials. For further guidance on handling this type of scenario, one should contact one's legal department and conduct further research on the CERT and SANS Web sites.

Auditing

Auditing will help ensure that file and directory systems are adhering to the organization's accepted standards. Although an organization may perform regular internal audits, it is recommended to have the file and directory structure audited by an external company annually. This process will help validate the internal results and limit any collusion that may be occurring within the organization.

Conclusion

While most businesses are addressing the markets' calls for user-friendly and ease-of-use operating systems, they are overlooking the security needs of most of the corporate infrastructure. This has led to unauthorized accesses to sensitive file structures and, as a result, is placing the organization in a major dilemma. Until we take the time to properly identify file and directory security permissions that best fit our organization's business charter, we cannot begin to feel confident with our overall network security strategy.

References

1. Anonymous, *Maximum Linux Security*, Sams Publishing, Indiana, 1999.
2. Jumes, James G. et al., *Microsoft Windows NT 4.0 Security, Audit and Control*, Microsoft Press, Redmond, Washington, 1999.
3. Internet Security Systems, Inc., *Microsoft Windows 2000 Security Technical Reference*, Microsoft Press, Redmond, Washington, 2000.
4. Kabir, Mohammed J., *Red Hat Linux Administrator's Handbook*, 2nd ed., M&T Books, California, 2001.
5. Schultz, E. Eugene, *Windows NT/2000 Network Security*, Macmillan Technical Publishing, New York, 2000.
6. Sutton, Steve, *Windows NT Security Guidelines*, Trusted Systems Services, Inc., 1999.

Domain 8
Business Continuity Planning

The Business Continuity Planning Domain addresses actions to preserve the business in the face of disruptions to normal business operations, including both natural and man-made events. Information systems and processing continuity are subject to many natural and man-made threats. Organizations must continually plan for potential business disruption, and test the recovery plans for their automated systems. Moreover, these organizations must continue to reengineer the continuity planning process, given the challenges of evolving technologies, including distributed computing and the World Wide Web.

Measures taken to ensure business continuity and disaster recovery have always been a challenge in the IT environment. The current information processing environment is much more complex to manage than those in the past. As systems and networks become more distributed, the control and manageability of those systems travels further away from a central source. In the world of Web applications, much of the control lies outside of the organization owning the resources. Thus, management may well be aware that continuity planning (CP) is important, but does not effectively execute their plans.

The chapters in this domain present a structured approach to contingency planning, including measures to demonstrate its value. Business Continuity Planning, of course, is necessary to ensure that the systems critical to keeping the organization viable are processed at an alternate site in time to avoid an intolerable business impact. The concepts of Business Impact Analysis (BIA) are examined as key tools to assist in the identification of critical applications, systems, and supporting resources.

Contents

134

Reengineering the Business Continuity Planning Process

Carl B. Jackson, CISSP, CBCP

The initial version of this chapter was written for the 1999 edition of the *Information Security Management Handbook*. Since then, E-commerce has seized the spotlight and Web-based technologies are the emerging solution for almost everything. The constant throughout these occurrences is that no matter what the climate, fundamental business processes have changed little. And, as always, the focus of any business impact assessment is to assess the time-critical priority of these business processes. With these more recent realities in mind, this chapter has been updated and is now offered for the reader's consideration.

Continuity Planning: Management Awareness High — Execution Effectiveness Low

The failure of organizations to accurately measure the contributions of the continuity planning (CP) process to their overall success has led to a downward spiraling cycle of the total business continuity program. The recurring downward spin or decomposition includes planning, testing, maintenance, decline → replanning, testing, maintenance, decline → replanning, testing, maintenance, decline, etc.

In the past, *Contingency Planning & Management (CPM)/Ernst & Young Continuity Planning Benchmark* surveys have repeatedly confirmed that continuity planning (CP) is ranked as being either "extremely important" or "very important" to executive management. The most recent *2000–2001 CPM/KPMG Continuity Planning Survey*[1] clearly supports this observation. This study indicates that a growing number of CP professional positions are migrating from the IT infrastructure to corporate or general management positions; however, CP reporting within the IT organization is still the norm. Approximately 40 percent of CP professionals currently report to IT, while around 30 percent report to corporate positions.

Continuity Planning Measurements

While the trends of this survey are encouraging, there is a continuing indication of a disconnect between executive management's perceptions of CP objectives and the manner in which they measure its value. Traditionally, CP effectiveness was measured in terms of a pass/fail grade on a mainframe recovery test, or on the perceived benefits of backup/recovery sites and redundant telecommunications weighed against the expense for these capabilities. The trouble with these types of metrics is that they only measure CP direct costs, or indirect perceptions as to whether a test was effectively executed. These metrics do not indicate whether a test validates the appropriate infrastructure elements or even whether it is thorough enough to test a component until it fails, thereby extending the reach and usefulness of the test scenario.

Thus, one might inquire as to the correct measures to use. Although financial measurements do constitute one measure of the CP process, others measure the CPs contribution to the organization in terms of quality

and effectiveness, which are not strictly weighed in monetary terms. The contributions that a well-run CP process can make to an organization include:

- Sustaining growth and innovation
- Enhancing customer satisfaction
- Providing people needs
- Improving overall mission-critical process quality
- Providing for practical financial metrics

A Receipt for Radical Change: CP Process Improvement

Just prior to the millennium, experts in organizational management efficiency began introducing performance process improvement disciplines. These process improvement disciplines have been slowly adopted across many industries and companies for improvement of general manufacturing and administrative business processes. The basis of these and other improvement efforts was the concept that an organization's processes (Process; see Exhibit 134.1) constituted the organization's fundamental lifeblood and, if made more effective and more efficient, could dramatically decrease errors and increase organizational productivity.

An organization's processes are a series of successive activities; and when they are executed in the aggregate, they constitute the foundation of the organization's mission. These processes are intertwined throughout the organization's infrastructure (individual business units, divisions, plants, etc.) and are tied to the organization's supporting structures (data processing, communications networks, physical facilities, people, etc.).

A key concept of the process improvement and reengineering movement revolves around identification of process enablers and barriers (see Exhibit 134.1). These enablers and barriers take many forms (people, technology, facilities, etc.) and must be understood and taken into consideration when introducing radical change into the organization.

The preceding narration provides the backdrop for the idea of focusing on continuity planning not as a project, but as a continuous process, that must be designed to support the other mission-critical processes of the organization. Therefore, the idea was born of adopting a continuous process approach to CP, along with understanding and addressing the people, technology, facility, etc., enablers and barriers. This constitutes a significant or even radical change in thinking from the manner in which recovery planning has been traditionally viewed and executed.

Radical Changes Mandated

High awareness of management and low CP execution effectiveness, coupled with the lack of consistent and meaningful CP measurements, call for radical changes in the manner in which one executes recovery planning responsibilities. The techniques used to develop mainframe-oriented disaster recovery (DR) plans of the 1980s and 1990s consisted of five to seven distinct stages, depending on whose methodology was being used, that required the recovery planner to:

1. Establish a project team and a supporting infrastructure to develop the plans.
2. Conduct a threat or risk management review to identify likely threat scenarios to be addressed in the recovery plans.
3. Conduct a business impact analysis (BIA) to identify and prioritize time-critical business applications and networks and determine maximum tolerable downtimes.
4. Select an appropriate recovery alternative that effectively addressed the recovery priorities and timeframes mandated by the BIA.
5. Document and implement the recovery plans.
6. Establish and adopt an ongoing testing and maintenance strategy.

Shortcomings of the Traditional Disaster Recovery Planning Approach

The old approach worked well when disaster recovery of "glass-house" mainframe infrastructures was the norm. It even worked fairly well when it came to integrating the evolving distributed client/server systems into the overall recovery planning infrastructure. However, when organizations became concerned with business unit recovery planning, the traditional DR methodology was ineffective in designing and implementing busi-

EXHIBIT 134.1 Definitions

Activities: Activities are things that go on within a process or sub-process. They are usually performed by units of one (one person or one department). An activity is usually documented in an instruction. The instruction should document the tasks that make up the activity.
Benchmarking: Benchmarking is a systematic way to identity, understand, and creatively evolve superior products, services, designs, equipment, processes, and practices to improve the organization's real performance by studying how other organizations are performing the same or similar operations.
Business process improvement: Business process improvement (BPI) is a methodology that is designed to bring about self-function improvements in administrative and support processes using approaches such as FAST, process benchmarking, process redesign, and process reengineering.
Comparative analysis: Comparative analysis (CA) is the act of comparing a set of measurements to another set of measurements for similar items.
Enabler: An enabler is a technical or organizational facility/resource that make it possible to perform a task, activity, or process. Examples of technical enablers are personal computers, copying equipment, decentralized data processing, voice response, etc. Examples of organizational enablers are enhancement, self-management, communications, education, etc.
Fast analysis solution technique: *FAST* is a breakthrough approach that focuses a group's attention on a single process for a one- or two-day meeting to define how the group can improve the process over the next 90 days. Before the end of the meeting, management approves or rejects the proposed improvements.
Future state solution: A combination of corrective actions and changes that can be applied to the item (process) under study to increase its value to its stakeholders.
Information: Information is data that has been analyzed, shared, and understood.
Major processes: A major process is a process that usually involves more than one function within the organization structure, and its operation has a significant impact on the way the organization functions. When a major process is too complex to be flowcharted at the activity level, it is often divided into sub-processes.
Organization: An organization is any group, company, corporation, division, department, plant, or sales office.
Process: A process is a logical, related, sequential (connected) set of activities that takes an input from a supplier, adds value to it, and produces an output to a customer.
Sub-process: A sub-process is a portion of a major process that accomplishes a specific objective in support of the major process.
System: A system is an assembly of components (hardware, software, procedures, human functions, and other resources) united by some form of regulated interaction to form an organized whole. It is a group of related processes that may or may not be connected.
Tasks: Tasks are individual elements or subsets of an activity. Normally, tasks relate to how an item performs a specific assignment.

From Harrington, H.J., Esseling, E.K.C., and Van Nimwegen, H., *Business Process Improvement Workbook,* McGraw-Hill, 1997, 1–20.

ness unit/function recovery plans. Of primary concern when attempting to implement enterprisewide recovery plans was the issue of functional interdependencies. Recovery planners became obsessed with identification of interdependencies between business units and functions, as well as the interdependencies between business units and the technological services supporting time-critical functions within these business units.

Losing Track of the Interdependencies

The ability to keep track of departmental interdependencies for CP purposes was extremely difficult and most methods for accomplishing this were ineffective. Numerous circumstances made consistent tracking of inter-

dependencies difficult to achieve. Circumstances affecting interdependencies revolve around the rapid rates of change that most modern organizations are undergoing. These include reorganization/restructuring, personnel relocation, changes in the competitive environment, and outsourcing. Every time an organizational structure changes, the CPs must change and the interdependencies must be reassessed; and the more rapid the change, the more daunting the CP reshuffling. Because many functional interdependencies could not be tracked, CP integrity was lost and the overall functionality of the CP was impaired. There seemed to be no easy answers to this dilemma.

Interdependencies Are Business Processes

Why are interdependencies of concern? And what, typically, are the interdependencies? The answer is that, to a large degree, these interdependencies are the business processes of the organization and they are of concern because they must function in order to fulfill the organization's mission. Approaching recovery planning challenges with a business process viewpoint can, to a large extent, mitigate the problems associated with losing interdependencies, and also ensure that the focus of recovery planning efforts is on the most crucial components of the organization. Understanding how the organization's time-critical business processes are structured will assist the recovery planner in mapping the processes back to the business units/departments; supporting technological systems, networks, facilities, vital records, people, etc.; and keeping track of the processes during reorganizations or during times of change.

The Process Approach to Continuity Planning

Traditional approaches to mainframe-focused disaster recovery planning emphasized the need to recover the organization's technological and communications platforms. Today, many companies have shifted away from technology recovery and toward continuity of prioritized business processes and the development of specific business process recovery plans. Many large corporations use the process reengineering/improvement disciplines to increase overall organizational productivity. CP itself should also be viewed as such a process. Exhibit 134.2 provides a graphical representation of how the enterprisewide CP process framework should look.

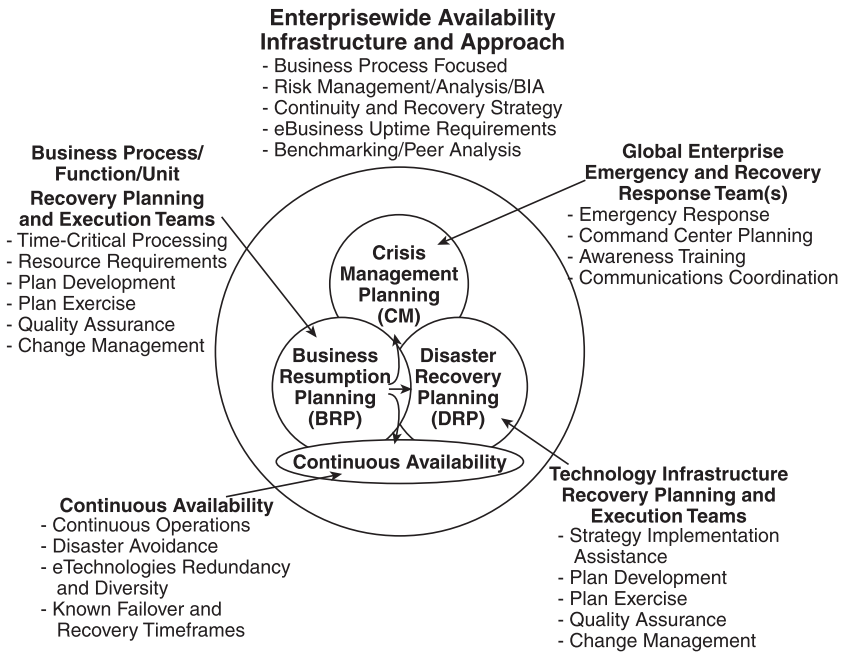


EXHIBIT 134.2 The enterprisewide CP process framework.

This approach to continuity planning consolidates three traditional continuity planning disciplines, as follows:

1. *IT disaster recovery planning (DRP)*. Traditional IT DRP addresses the continuity planning needs of the organizations' IT infrastructures, including centralized and decentralized IT capabilities and includes both voice and data communications network support services.
2. *Business operations resumption planning (BRP)*. Traditional BRP addresses the continuity of an organization's business operations (e.g., accounting, purchasing, etc.) should they lose access to their supporting resources (e.g., IT, communications network, facilities, external agent relationships, etc.).
3. *Crisis management planning (CMP)*. CMP focuses on assisting the client organization develop an effective and efficient enterprisewide emergency/disaster response capability. This response capability includes forming appropriate management teams and training their members in reacting to serious company emergency situations (e.g., hurricane, earthquake, flood, fire, serious hacker or virus damage, etc.). CMP also encompasses response to life-safety issues for personnel during a crisis or response to disaster.
4. *Continuous availability (CA)*. In contrast to the other CP components as explained above, the recovery time objective (RTO) for recovery of infrastructure support resources in a 24×7 environment has diminished to *zero* time. That is, the client organization cannot afford to lose operational capabilities for even a very short period of time without significant financial (revenue loss, extra expense) or operational (customer service, loss of confidence) impact. The CA service focuses on maintaining the highest uptime of support infrastructures to 99 percent and higher.

Moving to a CP Process Improvement Environment

Route Map Profile and High-Level CP Process Approach

A practical, high-level approach to CP process improvement is demonstrated by breaking down the CP process into individual sub-process components as shown in Exhibit 134.3.

The six major components of the continuity planning business process are described below.

1. *Current State Assessment/Ongoing Assessment.* Understanding the approach to enterprisewide continuity planning as illustrated in Exhibit 134.3, one can measure the "health" of the continuity planning process. During this process, existing continuity planning business sub-processes are assessed to gauge their overall effectiveness. It is sometimes useful to employ gap analysis techniques to understand current state, desired future state, and then understand the people, process, and technology barriers and enablers that stand between the current state and the future state. An approach to co-development of current state/future state visioning sessions is illustrated in Exhibit 134.4.
   The current state assessment process also involves identifying and determining how the organization "values" the CP process and measures its success (often overlooked and often leading to the failure of the CP process). Also during this process, an organization's business processes are examined to determine the impact of loss or interruption of service on the overall business through performance of a business impact assessment (BIA). The goal of the BIA is to prioritize business processes and assign the recovery time objective (RTO) for their recovery, as well as for the recovery of their support resources. An important outcome of this activity is the mapping of time-critical processes to their support resources (e.g., IT applications, networks, facilities, communities of interest, etc.).
2. *Process Risk and Impact Baseline.* During this process, potential risks and vulnerabilities are assessed, and strategies and programs are developed to mitigate or eliminate those risks. The stand-alone risk management review (RMR) commonly looks at the security of physical, environmental, and information capabilities of the organization. In general, the RMR should identify or discuss the following areas:

   - Potential threats
   - Physical and environmental security
   - Information security
   - Rrecoverability of time-critical support functions
   - Single-points-of-failure

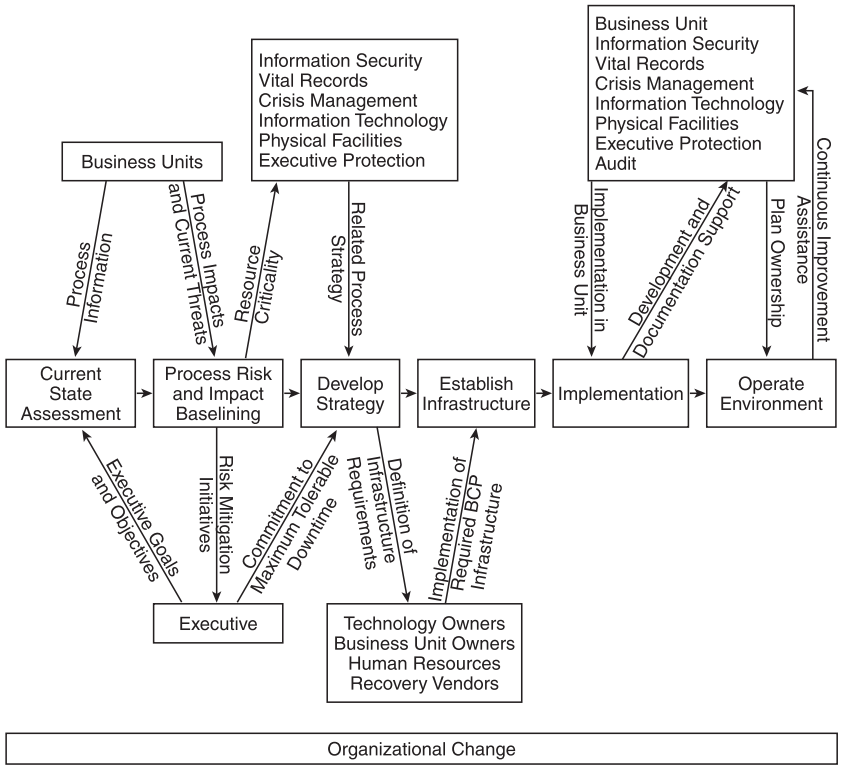


EXHIBIT 134.3 A practical, high-level approach to CP process improvement.

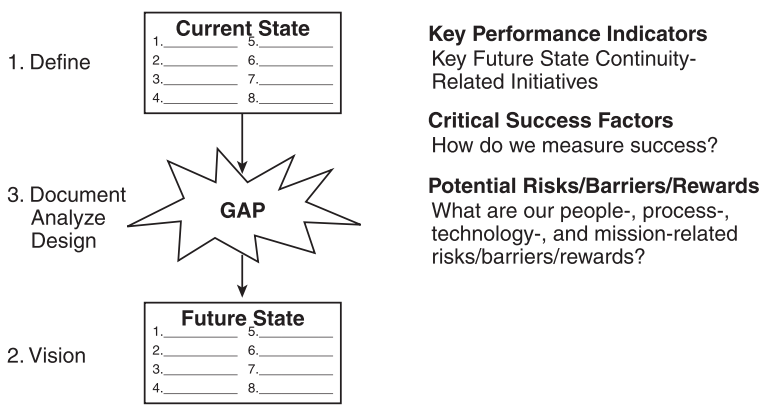


EXHIBIT 134.4 Current state/future state visioning overview.

- Problem and change management
- Business interruption and extra expense insurance
- An offsite storage program, etc.

3. *Strategy Development.* This process involves facilitating a workshop or series of workshops designed to identify and document the most appropriate recovery alternative to CP challenges (e.g., determining if a hotsite is needed for IT continuity purposes, determining if additional communications circuits should

be installed in a networking environment, determining if additional workspace is needed in a business operations environment, etc.). Using the information derived from the risk assessments above, design long-term testing, maintenance, awareness, training, and measurement strategies.

4. *Continuity Plan Infrastructure*. During plan development, all policies, guidelines, continuity measures, and continuity plans are formally documented. Structure the CP environment to identify plan owners and project management teams, and to ensure the successful development of the plan. In addition, tie the continuity plans to the overall IT continuity plan and crisis management infrastructure.
5. *Implementation.* During this phase, the initial versions of the continuity or crisis management plans are implemented across the enterprise environment. Also during this phase, long-term testing, maintenance, awareness, training, and measurement strategies are implemented.
6. *Operate Environment.* This phase involves the constant review and maintenance of the continuity and crisis management plans. In addition, this phase may entail maintenance of the ongoing viability of the overall continuity and crisis management business processes.

How Does One Get There? The Concept of the CP Value Journey

The CP value journey is a helpful mechanism for co-development of CP expectations by the organization's top management group and those responsible for recovery planning. To achieve a successful and measurable recovery planning process, the following checkpoints along the CP value journey should be considered and agreed upon. The checkpoints include:

- *Defining success*. Define what a successful CP implementation will look like. What is the future state?
- *Aligning the CP with business strategy*. Challenge objectives to ensure that the CP effort has a business-centric focus.
- *Charting an improvement strategy*. Benchmark where the organization and the organization's peers are, the organization's goals based on their present position as compared to their peers, and which critical initiatives will help the organization achieve its goals.
- *Becoming an accelerator*. Accelerate the implementation of the organization's CP strategies and processes. In today's environment, speed is a critical success factor for most companies.
- *Creating a winning team*. Build an internal/external team that can help lead the company through CP assessment, development, and implementation.
- *Assessing business needs*. Assess time-critical business process dependence on the supporting infrastructure.
- *Documenting the plans*. Develop continuity plans that focus on ensuring that time-critical business processes will be available.
- *Enabling the people*. Implement mechanisms that help enable rapid reaction and recovery in times of emergency, such as training programs, a clear organizational structure, and a detailed leadership and management plan.
- *Completing the organization's CP strategy*. Position the organization to complete the operational and personnel related milestones necessary to ensure success.
- *Delivering value*. Focus on achieving the organization's goals while simultaneously envisioning the future and considering organizational change.
- *Renewing/recreating*. Challenge the new CP process structure and organizational management to continue to adapt and meet the challenges of demonstrate availability and recoverability.

The Value Journey Facilitates Meaningful Dialogue

This value journey technique for raising the awareness level of management helps to both facilitate meaningful discussions about the CP process and ensure that the resulting CP strategies truly add value. As discussed later, this value-added concept will also provide additional metrics by which the success of the overall CP process can be measured.

The Need for Organizational Change Management

In addition to the approaches of CP process improvement and the CP value journey mentioned above, the need to introduce people-oriented organizational change management (OCM) concepts is an important component in implementing a successful CP process.

H. James Harrington et al., in their book *Business Process Improvement Workbook*,[2] point out that applying process improvement approaches can often cause trouble unless the organization manages the change process. They state that, "Approaches like reengineering only succeed if we challenge and change our paradigms and our organization's culture. It is a fallacy to think that you can change the processes without changing the behavior patterns or the people who are responsible for operating these processes."[3]

Organizational change management concepts, including the identification of people enablers and barriers and the design of appropriate implementation plans that change behavior patterns, play an important role in shifting the CP project approach to one of CP process improvement. The authors also point out that, "There are a number of tools and techniques that are effective in managing the change process, such as pain management, change mapping, and synergy. The important thing is that every BPI (Business Process Improvement) program must have a very comprehensive change management plan built into it, and this plan must be effectively implemented."[4]

Therefore, it is incumbent on the recovery planner to ensure that, as the concept of the CP process evolves within the organization, appropriate OCM techniques are considered and included as an integral component of the overall deployment effort.

How Is Success Measured? Balanced Scorecard Concept[5]

A complement to the CP process improvement approach is the establishment of meaningful measures or metrics that the organization can use to weigh the success of the overall CP process. Traditional measures include:

- How much money is spent on hotsites?
- How many people are devoted to CP activities?
- Was the hotsite test a success?

Instead, the focus should be on measuring the CP process contribution to achieving the overall goals of the organization. This focus helps to:

- Identify agreed-upon CP development milestones.
- Establish a baseline for execution.
- Validate CP process delivery.
- Establish a foundation for management satisfaction to successfully manage expectations.

The CP balanced scorecard includes a definition of the:

- Value statement
- Value proposition
- Metrics/assumptions on reduction of CP risk
- Implementation protocols
- Validation methods

Exhibits 134.5 and 134.6 illustrate the balanced scorecard concept and show examples of the types of metrics that can be developed to measure the success of the implemented CP process. Included in this balanced scorecard approach are the new metrics upon which the CP process will be measured.

Following this balanced scorecard approach, the organization should define what the future state of the CP process should look like (see the preceding CP value journey discussion). This future state definition should be co-developed by the organization's top management and those responsible for development of the CP process infrastructure. Exhibit 134.4 illustrates the current state/future state visioning overview, a technique that can also be used for developing expectations for the balanced scorecard. Once the future state is defined, the CP process development group can outline the CP process implementation critical success factors in the areas of:

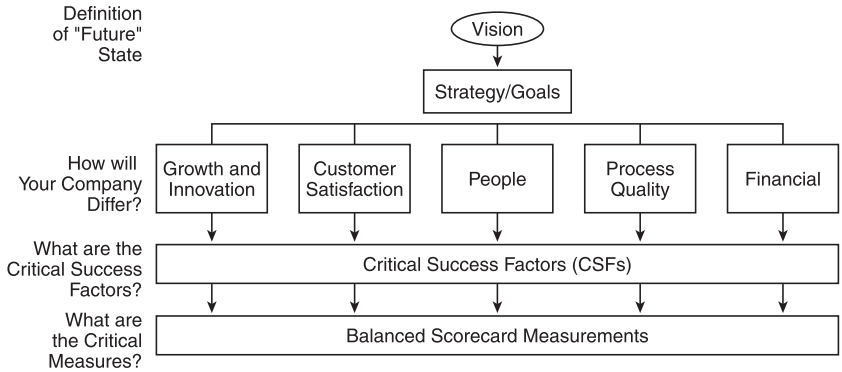


EXHIBIT 134.5 Balanced scorecard concept.

EXHIBIT 134.6 Continuity Process Scorecard

Question: How should the organization benefit from implementation of the following continutity process components in terms of people, processes, technologies, and mission/profits?

Continuity Planning Process Components	People	Processes	Technologies	Mission/Profits
Process methodology				
Documented DRPs				
Documented BRPs				
Documented crisis management plans				
Documented emergency response procedures				
Documented network recovery plan				
Contingency organization walk-throughs				
Employee awareness program				
Recovery alternative costs				
Continuous availability infrastructure				
Ongoing testing programs				
etc.				

- Growth and innovation
- Customer satisfaction
- People
- Process quality
- Financial state

These measures must be uniquely developed based on the specific organization's culture and environment.

What about Continuity Planning for Web-Based Applications?

Evolving with the birth of the Web and Web-based businesses is the requirement for 24×7 uptime. Traditional recovery time objectives have disappeared for certain business processes and support resources that support the organizations' Web-based infrastructure. Unfortunately, simply preparing Web-based applications for sustained 24×7 uptime is not the only answer. There is no question that application availability issues must be addressed, but it is also important that the reliability and availability of other Web-based infrastructure components (such as computer hardware, Web-based networks, database file systems, Web servers, file and print servers, as well as preparing for the physical, environmental, and information security concerns relative to each of these [see RMR above]) also be undertaken. The terminology for preparing the entirety of this

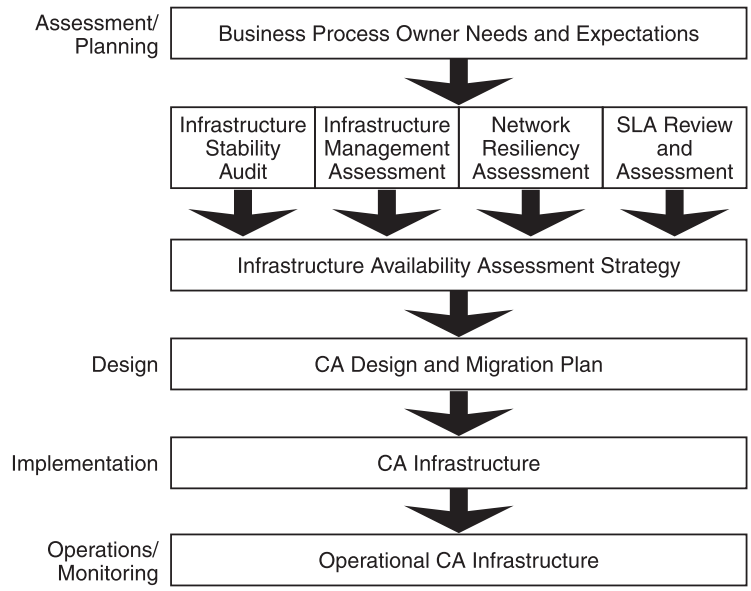


EXHIBIT 134.7 Continuous availability methodological approach.

infrastructure to remain available through major and minor disruptions is usually referred to as continuous or high availability.

Continuous availability (CA) is not simply bought; it is planned for and implemented in phases. The key to a reliable and available Web-based infrastructure is to ensure that each of the components of the infrastructure have a high-degree of resiliency and robustness. To substantiate this statement, *Gartner Research* reports "Replication of databases, hardware servers, Web servers, application servers, and integration brokers/suites helps increase availability of the application services. The best results, however, are achieved when, in addition to the reliance on the system's infrastructure, the design of the application itself incorporates considerations for continuous availability. Users looking to achieve continuous availability for their Web applications should not rely on any one tool but should include the availability considerations systematically at every step of their application projects."[7]

Implementing a continuous availability methodological approach is the key to an organized and methodical way to achieve 24×7 or near 24×7 availability. Begin this process by understanding business process needs and expectations, and the vulnerabilities and risks of the network infrastructure (e.g., Internet, intranet, extranet, etc.), including undertaking single-points-of-failure analysis. As part of considering implementation of continuous availability, the organization should examine the resiliency of its network infrastructure and the components thereof, including the capability of its infrastructure management systems to handle network faults, network configuration and change, the ability to monitor network availability, and the ability of individual network components to handle capacity requirements. See Exhibit 134.7 for an example pictorial representation of this methodology.

The CA methodological approach is a systematic way to consider and move forward in achieving a Web-based environment. A very high-level overview of this methodology is as follows.

- *Assessment/planning*. During this phase, the enterprise should endeavor to understand the current state of business process owner expectations/requirements and the components of the technological infrastructure that support Web-based business processes. Utilizing both interview techniques (people to people) and existing system and network automated diagnoses tools will assist in understanding availability status and concerns.
- *Design*. Given the results of the current state assessment, design the continuous availability strategy and implementation/migration plans. This will include developing a Web-based infrastructure classification system to be used to classify the governance processes used for granting access to and use of support for Web-based resources.

- *Implementation*. Migrate existing infrastructures to the Web-based environment according to design specifications as determined during the design phase.
- *Operations/monitoring*. Establish operational monitoring techniques and processes for the ongoing administration of the Web-based infrastructure.

Along these lines, in their book *Blueprints for High Availability: Designing Resilient Distributed Systems*,[8] Marcus and Stern recommend several fundamental rules for maximizing system availability (paraphrased):

- *Spend money…but not blindly*. Because quality costs money, investing in an appropriate degree of resiliency is necessary.
- *Assume nothing*. Nothing comes bundled when it comes to continuous availability. End-to-end system availability requires up-front planning and cannot simply be bought and dropped in place.
- *Remove single-points-of-failure*. If a single link in the chain breaks, regardless of how strong the other links are, the system is down. Identify and mitigate single-points-of-failure.
- *Maintain tight security*. Provide for the physical, environmental, and information security of Web-based infrastructure components.
- *Consolidate servers*. Consolidate many small servers' functionality onto larger servers and less numerous servers to facilitate operations and reduce complexity.
- *Automate common tasks*. Automate the commonly performed systems tasks. Anything that can be done to reduce operational complexity will assist in maintaining high availability.
- *Document everything*. Do not discount the importance of system documentation. Documentation provides audit trails and instructions to present and future systems operators on the fundamental operational intricacies of the systems in question.
- *Establish service level agreements (SLAs)*. It is most appropriate to define enterprise and service provider expectations ahead of time. SLAs should address system availability levels, hours of service, locations, priorities, and escalation policies.
- *Plan ahead*. Plan for emergencies and crises, including multiple failures in advance of actual events.
- *Test everything*. Test all new applications, system software, and hardware modifications in a production-like environment prior to going live.
- *Maintain separate environments*. Provide for separation of systems, when possible. This separation might include separate environments for the following functions: production, production mirror, quality assurance, development, laboratory, and disaster recovery/business continuity site.
- *Invest in failure isolation*. Plan — to the degree possible — to isolate problems so that if or when they occur, they cannot boil over and affect other infrastructure components.
- *Examine the history of the system*. Understanding system history will assist in understanding what actions are necessary to move the system to a higher level of resiliency in the future.
- *Build for growth*. A given in the modern computer era is that system resource reliability increases over time. As enterprise reliance on system resources grow, the systems must grow. Therefore, adding systems resources to existing reliable system architectures requires preplanning and concern for workload distribution and application leveling.
- *Choose mature software*. It should go without saying that mature software that supports a Web-based environment is preferred over untested solutions.
- *Select reliable and serviceable hardware*. As with software, selecting hardware components that have demonstrated high mean times between failures is preferable in a Web-based environment.
- *Reuse configurations*. If the enterprise has stable system configurations, reuse or replicate them as much as possible throughout the environment. The advantages of this approach include ease of support, pretested configurations, a high degree of confidence for new rollouts, bulk purchasing possible, spare parts availability, and less to learn for those responsible for implementing and operating the Web-based infrastructure.
- *Exploit external resources*. Take advantage of other organizations that are implementing and operating Web-based environments. It is possible to learn from others' experiences.
- *One problem, one solution*. Understand, identify, and utilize the tools necessary to maintain the infrastructure. Tools should fit the job; so obtain them and use them as they were designed to be used.

- *KISS: keep it simple….* Simplicity is the key to planning, developing, implementing, and operating a Web-based infrastructure. Endeavor to minimize Web-based infrastructure points of control and contention, as well as the introduction of variables.

Marcus and Stern's book[8] is an excellent reference for preparing for and implementing highly available systems.

Reengineering the continuity planning process involves not only reinvigorating continuity planning processes, but also ensuring that Web-based enterprise needs and expectations are identified and met through the implementation of continuous availability disciplines.

Summary

The failure of organizations to measure the success of their CP implementations has led to an endless cycle of plan development and decline. The primary reason for this is that a meaningful set of CP measurements has not been adopted to fit the organization's future-state goals. Because these measurements are lacking, expectations of both top management and those responsible for CP often go unfulfilled. Statistics gathered in the *Contingency Planning & Management/KPMG Continuity Planning Survey* support this assertion. Based on this, a radical change in the manner in which organizations undertake CP implementation is necessary. This change should include adopting and utilizing the business process improvement (BPI) approach for CP. This BPI approach has been implemented successfully at many Fortune 1000 companies over the past 20 years. Defining CP as a process, applying the concepts of the CP value journey, expanding CP measurements utilizing the CP balanced scorecard, and exercising the organizational change management (OCM) concepts will facilitate a radically different approach to CP. Finally, because Web-based business processes require 24×7 uptime, implementation of continuous availability disciplines are necessary to ensure that the CP process is as fully developed as it should be.

References

1. *Contingency Planning & Management*, January/February 2001. (The survey was conducted in the U.S. in October 2000 and consisted of readers and respondents drawn from *Contingency Planning & Management* magazine's domestic subscription list. Industries represented by respondents include Financial Services; Manufacturing/Industrial, Telecommunications, Education, Utilities, Healthcare, Insurance, Retail/Wholesale, Petroleum/Chemical, Information/Data Processing, Media/Entertainment; and Computer Services/Systems.)
2. Harrington, H.J., Esseling, E.K.C., and Van Nimwegen, H., *Business Process Improvement Workbook*, McGraw-Hill, 1997.
3. Harrington, p. 18.
4. Harrington, p. 19.
5. Robert S. Kaplan and David P. Norton, *Translating Strategy into Action: The Balanced Scorecard,* HBS Press, 1996.
6. Harrington, p. 1-20.
7. Gartner Group RAS Services, COM-12-1325, 29 September 2000.
8. Marcus, E. and Stern, H., *Blueprints for High Availability: Designing Resilient Distributed Systems,* John Wiley & Sons, 2000.

136

The Role of Continuity Planning in the Enterprise Risk Management Structure

Carl Jackson, CISSP, CBCP

Driving Continuity Planning to the Next Level

Traditional approaches to IT-centric disaster planning emphasized the need to recover the organization's technological and communications platforms. Today, many organizations have shifted away from focusing strictly on technology recovery and more toward continuity of prioritized business processes and the development of specific business process recovery plans. In addition, continuity planners are also beginning to articulate the value of a fully functioning and ongoing continuity planning (CP) business process to the enterprise, and not just settling for BCP as usual. In fact, many organizations are expanding the CP business process beyond traditional boundaries to combine and support a larger organizational component, i.e., enterprise risk management (ERM) functionality.

The purpose of this chapter is to discuss the role of continuity planning business processes in supporting an enterprise view of risk management and to highlight how the ERM and CP organizational components, working in harmony, can provide measurable value to the enterprise, people, technologies, processes, and mission. The chapter also focuses briefly on additional continuity process improvement techniques.

If not already considered a part of the organization's overall enterprise risk management program, why should business continuity planning professionals seriously pursue aligning their continuity planning programs with ERM initiatives? The answer follows.

The Lack of Meaningful Metrics

Lack of suitable business objectives-based metrics has forever plagued the CP profession. As CP professionals, we have for the most part failed to sufficiently define and articulate a high-quality set of metrics by which we would have management gauge the success of CP business processes. So often, we allow ourselves to be measured either by way of fiscal measurements (i.e., cost of hot-site contracts, cost of software, cost of head count, etc., all in comparison to some ill-defined percentage of the annual IT budget), or in terms of successful or nonsuccessful CP tests, or in the absence of unfavorable audit comments.

On the topic of measurement, the most recent Contingency Planning & Management/KPMG 2002 Business Continuity Planning Survey,[1] (http://www.contingencyplanning.com/) had some interesting insights. When asked how their organization measured the performance of their BCP program, survey respondents answered as shown in Exhibit 136.1.

EXHIBIT 136.1 How Does an Organization Measure the Performance of Its BCP Program?

	Percent
Service-level monitoring	26
Results of BCP testing	54
Audit findings	40
Performance reviews	30
Benchmarking/comparison to industry norms	14

This annual BCP survey makes it clear that rather than measure CP program effectiveness based on value-added contributions to enterprise value drivers, management continues to base CP performance on the results of tests or on adverse audit comments.

Shareholder Expectations

Should shareholders hold an executive manager responsible for overall enterprise performance? Or should management be held accountable for the success or failure of individual board of director votes, or one or two tactical decisions in support of strategic goals? Overall enterprise performance against revenue, profit, and marketplace goals is the usual answer given to these questions. Tactical decisions made to achieve those goals sometimes are successful and sometimes they are not, but it is the overall effect that is important.

Rather than being measured on quantitative financial measures only, why should the CP profession not consider developing both quantitative *and* qualitative metrics that are based on the value drivers and business objectives of the enterprise? We need to be phrasing CP business process requirements and value contributions in terms with which executive management can readily identify. Consider the issues from the executive management perspective. They are interested in ensuring that they can support shareholder value and clearly articulate this value in terms of business process contributions to organizational objectives. As we recognize this, we need to begin restructuring how the CP processes are measured. Many organizations have redefined or are in the process of redefining CP as part of an overarching ERM structure. The risks that CP processes are designed to address are just a few of the many risks that organizations must face. Consolidation of risk-focused programs or organizational components, like information security, risk management, legal, insurance, etc., makes sense; and in most cases capitalizes on economies of scale.

Given this trend, consider the contribution an enterprise risk management program should make to an organization.

The Role of Enterprise Risk Management

The Institute of Internal Auditors (IIA), in its publication, *Enterprise Risk Management: Trends and Emerging Practices*,[2] describes the important characteristics of a definition for ERM as:

- Inclusion of risks from all sources (financial, operational, strategic, etc.) and exploitation of the "natural hedges" and "portfolio effects" from treating these risks in the collective
- Coordination of risk management strategies that span:
  - — Risk assessment (including identification, analysis, measurement, and prioritization)
  - — Risk mitigation (including control processes)
  - — Risk financing (including internal funding and external transfer such as insurance and hedging)
  - — Risk monitoring (including internal and external reporting and feedback into risk assessment, continuing the loop)
- Focus on the impact to the organization's overall financial and strategic objectives

According to the IIA, the true definition of ERM is "dealing with uncertainty" and is defined by them as "a rigorous and coordinated approach to assessing and responding to all risks that affect the achievement of an organization's strategic and financial objectives. This includes both upside and downside risks."

It is the phrase "coordinated approach to assessing and responding to all risks" that is driving many continuity planning and risk management professionals to consider proactively bundling their efforts under the banner of ERM.

Trends

What are the trends that are driving the move to include traditional continuity planning disciplines within the ERM arena? Following are several examples of the trends that clearly illustrate that there are much broader risk issues to be considered, with CP being just another mitigating or controlling mechanism.

- *Technology risk:* To support mission-critical business processes, today's business systems are complex, tightly coupled, and heavily dependent on infrastructure. The infrastructure has a very high degree of interconnectivity in areas such as telecommunications, power generation and distribution, transportation, medical care, national defense, and other critical government services. Disruptions or disasters cause ripple effects within the infrastructure with failures inevitable.
- *Terrorism risk:* Terrorists have employed low-tech weapons to inflict massive physical or psychological damage (box cutters, anthrax-laden envelopes). Technologies and tools that have the ability to inflict massive damage are getting cheaper and easier to obtain every day, and are being used by competitors, customers, employees, litigation teams, etc. Examples include:
- *Cyber-activism:* The Electronic Disturbance Theater and Floodnet, which conducts virtual protests by flooding a particular Web site in protest
- *Cyber-terrorism:* NATO computers hit with e-mail bombs and denial-of-service attacks during the 1999 Kosovo conflict.
- *Legal and regulatory risk:* There is a large and aggressive expansion of legal and regulatory initiatives, including the Sarbanes–Oxley Act (accounting, internal control review, executive verification, ethics and whistleblower protection), HIPAA (privacy, information security, physical security, business continuity), Customs-Trade Partnership Against Terrorism (process control, physical security, personnel security), and the Department of Homeland Security initiatives, including consolidation of agencies with various risk responsibilities.
- *Recent experience:* Recent events including those proclaimed in headlines and taking place in such luminary companies as Enron, Arthur Andersen, WorldCom, Adelphia, HealthSouth, and GE have shaken the grounds of corporate governance. These experiences reveal and amplify underlying trends impacting the need for an enterprise approach to risk management.

Response

Most importantly, the continuity planner should start by understanding the organization's value drivers, those that influence management goals and answer the questions as to how the organization actually works. Value drivers are the forces that influence organizational behavior, how the management team makes business decisions, and where it spends its time, budgets, and other resources. Value drivers are the particular parameters that management expects to impact its environment. Value drivers are highly interdependent. Understanding and communicating value drivers and the relationship between them are critical to the success of the business to enable management objectives and prioritize investments.

In organizations that have survived through events such as September 11, 2001, the War on Terrorism, Wall Street roller coasters, world economics, and the like, there is a realization that ERM is broader than just dealing with insurance coverage. The enterprise risk framework is similar to the route map pictured in Exhibit 136.2. Explanations of the key components of this framework are as follows:

Business Drivers

Business drivers are the key elements or levers that create value for stakeholders, and particularly shareholders. Particular emphasis should be made on an organization's ability to generate excess cash, and the effective use of that cash. Business drivers vary by industry; however, they will generally line up in four categories:

1. *Manage growth:* Increasing revenue or improving the top line is achieved in many ways, such as expanding into new markets, overseas expansion, extending existing product lines, developing new product areas, and customer segments.

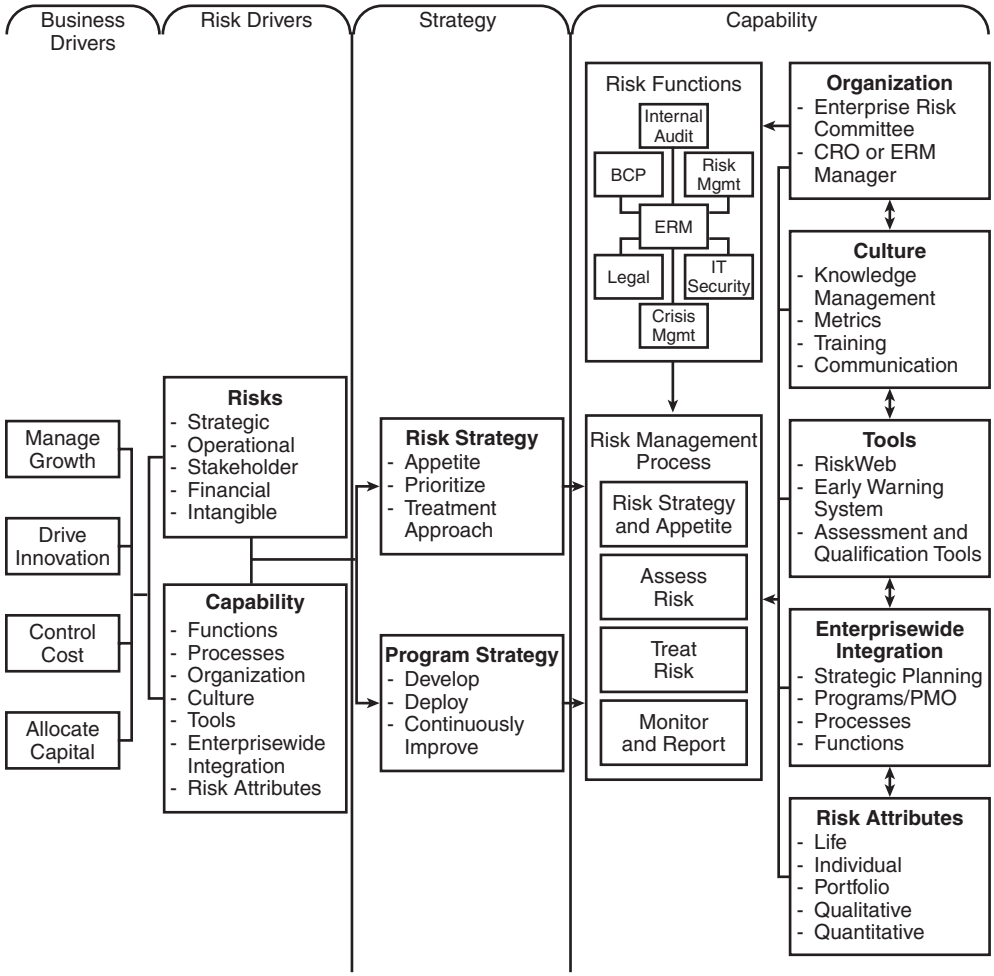


EXHIBIT 136.2 Enterprise risk management framework.

2. *Drive innovation:* The ability to create new products and markets through product innovativeness, product development, etc. New products and markets often give the creator a competitive advantage, leading to pricing power in the market, which allows the company to generate financial returns in excess of its competition.
3. *Control costs*: Effectively managing cost increases the competitive positioning of the business and the amount of cash left over.
4. *Allocate capital:* Capital should be effectively allocated to those business units, initiatives, markets, and products that will have the highest return for the least risk. These are the primary business drivers; they are what the organization does and the standards by which it expects to be measured.

Risk Drivers

Both the types of risk and the capability of the organization to manage those risks should be considered.

- *Risk types*: The development of a risk classification or categorization system has many benefits for an organization. The classification system creates a common nomenclature that facilitates discussions about risk issues within the organization. The system also facilitates the development of information systems that gather, track, and analyze information about various risks, including the ability to correlate cause

and effect, identify interdependencies, and track budgeting and loss experience information. Although many risk categorization methods exist, Exhibit 136.3 provides examples of risk types and categories.

- *Risk capability:* The ability of the organization to absorb and manage various risks, including how well the various risk management-related groups work together, what the risk process is within the enterprise, what organizational cultural elements should be considered, etc. The key areas of the risk capability will be discussed in greater detail later.

Risk Strategy

The strategy development section focuses management attention on both risk strategy and program strategy.

- *Risk appetite:* Of importance in the risk strategy is the definition of appetite for risk. Risk appetite levels need to be set for various types of impacts. Each risk level should have a corresponding response that then is cascaded throughout the organization.
- *Prioritization*: Based on the risk level, the inventory of risks should be prioritized and considered for the treatment approach.
- *Treatment approach*: Although most continuity planners focus on reducing risk through contingency planning, many alternatives exist and should be thoroughly considered.
  - — *Accept risk:* Management decides to continue operations as-is with a consensus to accept the inherent risks.
  - — *Transfer risk:* Management decides to transfer the risk, for example, from one business unit to another or from one business area to a third party (i.e., insurer).
  - — *Eliminate risk:* Management decides to eliminate risk through the dissolution of a key business unit or operating area.
  - — *Acquire risk:* Management decides that the organization has a core competency managing this risk, and seeks to acquire additional risk of this type.
  - — *Reduce risk:* Management decides to reduce current risks through improvement in controls and processes.
  - — *Share risk:* Management attempts to share risk through partnerships, outsourcing, or other risk-sharing approaches.

Program Strategy

Business continuity planning programs, like all other risk management programs, require strategic planning and active management of the program. This includes developing a strategic plan and implementation work plans, as well as obtaining management support, including required resources (people, time, and funding) necessary to implement the plan.

EXHIBIT 136.3 Risk Types and Categories

Strategic	Operational	Stakeholder	Financial	Intangible
Macro trends	Business interruption	Customers	Transaction fraud	Brand/reputation
Competitor	Privacy	Line employees	Credit	Knowledge
Economic	Marketing	Management	Cash management	Intellectual property
Resource allocations	Processes	Suppliers	Taxes	Information systems
Program/project	Physical assets	Government	Regulatory compliance	Information for decision making
Organization structure	Technology infrastructure	Partners	Insurance	
Strategic planning	Legal	Community	Accounting	
Governance	Human resources			
Brand/reputation				
Ethics				
Crisis				
Partnerships/JV				

Capabilities

The risk management capability speaks to the ability of the organization to effectively identify and manage risk. Following is a list of some of the key elements that make up the risk management capability:

- *Risk Functions*: Various risk management functions must participate, exchange information and processes, and cooperate on risk mitigation activities to fully implement an ERM capability. Some of these risk management functions might include:
  - Business continuity planning
  - Internal audit
  - Insurance
  - Crisis management
  - Privacy
  - Physical security
  - Legal
  - Information security
  - Credit risk management

Defining Risk Management Processes

Effective risk management processes can be used across a wide range of risk management activities, including:

- Risk strategy and appetite
  - Define risk strategy and program
  - Define risk appetite
  - Determine treatment approach
  - Establish risk policies, procedures, and standards
- Assess risk
  - Identify and understand value and risk drivers
  - Categorize risk within the business risk framework
  - Identify methods to measure risk
  - Measure risk
  - Assemble risk profile and compare to risk appetite and capability
- Treat risk
  - Identify appropriate risk treatment methods
  - Implement risk treatment methods
  - Measure and assess residual risk
- Monitor and report
  - Continuously monitor risks
  - Continuously monitor risk management program and capabilities
  - Report on risks and effectiveness of risk management program and capabilities

Organization

A Chief Risk Officer (CRO), an enterprise risk manager, or even an enterprise risk committee may manage the enterprise risk management activities. Their duties would typically include:

- Provide risk management program leadership, strategy, and implementation direction.
- Develop risk classification and measurement systems.
- Develop and implement escalation metrics and triggers (events, incidents, crisis, operations, etc.).
- Develop and monitor early warning systems based on escalation metrics and triggers.
- Develop and deliver organizationwide risk management training.

- Coordinate risk management activities; some functions may report to the CRO, others will be coordinated.

Culture

Creating and maintaining an effective risk management culture is very difficult. Special consideration should be given to the following areas:

- *Knowledge management:* Institutional knowledge about risks, how they are managed, and experiences by other business units should be effectively captured and shared with relevant peers and risk managers.
- *Metrics:* The accurate and timely collection of metrics is critical to the success of the risk management program. Effort should be made to connect the risk management programs to the Balanced Scorecard, EVA, or other business management and metrics systems.
  - The Balanced Scorecard is a management system (not only a measurement system) that enables organizations to clarify their vision and strategy and translate them into action. It provides feedback around both the internal business processes and external outcomes to continuously improve strategic performance and results. When fully deployed, the Balanced Scorecard transforms strategic planning from an academic exercise into the reality of organizational measurement processes.[3]
  - EVA (Economic Value Added) is net operating profit minus an appropriate charge for the opportunity cost of all capital invested in an enterprise. As such, EVA is an estimate of true "economic" profit, or the amount by which earnings exceed or fall short of the required minimum rate of return that shareholders and lenders could get by investing in other securities of comparable risk. Stern Stewart developed EVA to help managers incorporate two basic principles of finance into their decision making. The first is that the primary financial objective of any company should be to maximize the wealth of its shareholders. The second is that the value of a company depends on the extent to which investors expect future profits to exceed or fall short of the cost of capital.[4]
- *Training:* Effective training programs are necessary to ensure that risk management programs are effectively integrated into the regular business processes. For example, strategic planners will need constant reinforcement in risk assessment processes.
- *Communication:* Frequent and consistent communications around the purpose, success, and cost of the risk management program are a necessity to maintain management support and to continually garner necessary participation of managers and line personnel in the ongoing risk management program.
- *Tools:* Appropriate tools should be evaluated or developed to enhance the effectiveness of the risk management capability. Many commercial tools are available and their utility across a range of risk management activities should be considered. Quality information about risks is generally difficult to obtain and care should be exercised to ensure that information gathered by one risk function can be effectively shared with other programs. For example, tools used to conduct the business impact assessment should facilitate the sharing of risk data with the insurance program.
- *Enterprisewide Integration:* The ERM and BCP programs should effectively collaborate across the enterprise and should have a direct connection to the strategic planning process, as well as the critical projects, initiatives, business units, functions, etc. Broad, comprehensive integration of risk management programs across the organization generally lead to more effective and efficient programs.

Risk Attributes

Risk attributes relate to the ability or sophistication of the organization to understand the characteristics of specific risks, including their life cycle, how they act individually or in a portfolio, and other qualitative or quantitative characteristics.

- *Life Cycle:* Has the risk been understood throughout its life cycle and have risk management plans been implemented before the risk occurs, during the risk occurrence, and after the risk? This obviously requires close coordination between the risk manager and the continuity planner.
- *Individual and Portfolio:* The most sophisticated organizations will look at each risk individually, as well as in aggregate or in portfolio. Viewing risks in a portfolio can help identify risks that are natural hedges

against themselves, and risks that amplify each other. Knowledge of how risks interact as a portfolio can increase the ability of the organization to effectively manage the risks at the most reasonable cost.

- *Qualitative and Quantitative:* Most organizations will progress from being able to qualitatively assess risks to being able to quantify risks. In general, the more quantifiable the information about the risk, the more treatment options available to the organization.

The Role of Continuity Planning

From the enterprise view, business continuity planning is an integral element of the risk functionality as mentioned earlier. The main message is that the control functions should be organized and exercised in a planned manner for the good of the enterprise.

A well-constructed and implemented enterprisewide approach to continuity planning enables an organization to deal effectively with a major business disruption. Continuity planning is a process that minimizes the impact on an organization's time-critical business processes given significant disruptive events such as power outages, natural disasters, accidents, acts of sabotage, or other such occurrences. The CP process is intended to help management develop cost-effective approaches to ensuring continuity during and after an interruption of time-critical processes, supporting systems, and resources. An effective planning structure will address the information required and steps involved in recovering and maintaining time-critical business processes — the lifeblood of an organization. Continuity planning services should be designed to assist in the development, implementation, and maintenance of effective continuity plans focused on the unique needs of the organization.

The CP process also includes assessing and improving the overall Crisis Management Planning (CMP) infrastructure of the organization. CMP focuses on assisting the organization to develop an effective and efficient enterprisewide emergency and disaster response capability. This response capability includes forming appropriate management teams and training team members in reacting to serious company emergency situations (i.e., hurricane, earthquake, flood, fire, serious hacker or virus damage, etc.).

The continuity planning approach consolidates three traditional continuity-planning disciplines as follows:

1. IT disaster recovery planning (DRP). Traditional disaster recovery planning addresses the restoration planning needs of the organization's IT infrastructures, including centralized and decentralized IT capabilities, and includes both voice and data communications network support services.
2. Business continuity planning (BCP). Traditional BCP addresses continuity of an organization's business operations (i.e., Accounting, Procurement, HR, etc.) should they lose access to their supporting resources (i.e., IT, communications network, facilities, external agent relationships, etc.).
3. Crisis management planning (CMP). CMP focuses on assisting the organization to develop an effective and efficient enterprisewide emergency and disaster response capability. This response capability includes forming appropriate management teams and training their members in reacting to serious company emergency situations (i.e., hurricane, earthquake, flood, fire, serious hacker or virus damage, etc.) to at least minimize but avoid (hopefully) a disaster. CMP also encompasses response to life-safety issues for personnel during a crisis or response to disaster. Nowhere is the need for effective risk management capabilities more evident than at a time of managing a crisis. In light of the recent headline incidents of corporate meltdowns, global terrorism, and a rapidly changing business environment, boards of directors and senior management must now take the time to reassess their organizations' crisis and enterprise risk management (ERM) capabilities.

The key components of the continuity planning development methodology are discussed next.

Assessment Phase

- *Business impact assessment (BIA):* During this process, an organization's business objectives and processes are examined to determine the impact of loss or interruption of service on the overall business. The goal of the BIA is to prioritize business processes and assign the recovery time objective (RTO) for their recovery and the recovery of their support resources. An important outcome of this activity is the mapping of time-critical processes to their support resources (i.e., IT applications, networks, facilities, third parties, etc.).

- *CP process current state assessment:* This process involves analyzing the organization's environment to gauge the health and vitality of the continuity planning process. This process also involves identifying or determining how the organization values the CP process and measures its success (an often-overlooked process and one that frequently leads to the failure of the CP process).
- *Risk management review (RMR):* During this process, potential risks and vulnerabilities are assessed and strategies and programs are developed to mitigate or eliminate those risks. Using traditional qualitative risk assessment approaches that focus on the security of physical, environmental, and information capabilities of the organization can support this process. In general, the RMR should identify or discuss seven basic areas:

  1. Potential threats
  2. Physical security
  3. Recoverability of time-critical processes and support resources
  4. Single points of failure
  5. Problem and change management
  6. Business interruption and extra-expense insurance
  7. A critical system off-site storage program

Design Phase

- *Leading practices/benchmarking services:* This optional component encompasses reviewing the performance of industry and peer benchmarking studies to determine leading practices, which can then be used to help establish the most appropriate Future State Vision for the organization's CP infrastructure.
- *Recovery strategy visioning:* This interactive, facilitated process includes developing an appropriate and measurable CP process. Major organization stakeholders can use this technique to develop the best possible overall CP process by encouraging input and buy-in.
- *Recovery strategy development:* This practice involves facilitating a workshop or series of workshops designed to determine and document the most appropriate recovery alternative to CP challenges (i.e., determining whether a hot site is needed for IT continuity purposes; whether additional communications circuits should be installed in a networking environment; whether additional workspace is needed in a business operations environment, etc.) using the information derived from the business impact assessments. From these facilitated workshops, the CP development team works with the organization teams to create a business case documenting the optimal recovery alternative solutions.
- *Continuity plan development:* During plan development, the recovery team members are selected, assigned, and formally documented. The detailed activities and tasks associated with the recovery of time-critical processes (or IT infrastructure components, etc.) are detailed and assigned to recovery team members. All the inventory information needed by the recovery team members is also collected and documented, including data, software, telecommunications, people, space, documentation, offsite workspace, equipment, etc.
- *CP testing, maintenance, training, and measurement:* During this process, the CP development team works with the organization management to design appropriate CP testing, maintenance, training, and measurement strategies and guidelines.

Implement Phase

- *Plan testing:* During plan testing, the CP development team works with business unit leaders to simulate potential disasters and test continuity plans for effectiveness. Any necessary adjustments and modifications are incorporated into the plan.
- *CP process implementation:* During this phase, the development team will work with the organization to deploy the continuity plans that have been developed, and to implement long-term testing, maintenance, training, and measurement strategies, as determined in the Design Phase.
- *Continuity and crisis management plan implementation:* During this phase, the initial versions of the continuity and crisis management plans are implemented across the enterprise environment.

Measure Phase

The continuity plan and process review and maintenance phase involves the regular review and maintenance of the continuity and crisis management plans.

Other Techniques for Improving CP Efficiencies

In combination with the introduction of ERM disciplines in improving the CP function, traditional CP Process Improvement, Organizational Change Management, and Balanced Scorecard techniques can also be used to assist in improving the efficiencies of continuity planning business processes.

CP Process Improvement

Harrington et al., in *Business Process Improvement Workbook*,[5] point out that applying process improvement approaches can often cause trouble unless the organization manages the change process. They state that

> …approaches like reengineering only succeed if we challenge and change our paradigms and our organization's culture. It is a fallacy to think that we can change the processes without changing the behavior patterns or the people who are responsible for operating these processes.

The Need for Organizational Change Management

The plans may be ready for the company, but the company may not be ready for the plans. Organizational change management concepts, including the identification of people enablers and barriers, and the design of appropriate implementation plans that change behavior patterns, play an important role in shifting the CP project approach to one of CP process improvement.

> There are a number of tools and techniques that are effective in managing the change process, such as pain management, change mapping, and synergy. The important thing is that every BPI program must have a very comprehensive change management plan built into it, and this plan must be effectively implemented.[5]

How Can We Measure Success? The Balanced Scorecard Concept

A complement to the CP Process Improvement approach is the establishment of meaningful measures or metrics that the organization can use to weigh the success of the overall CP process. This concept was mentioned briefly when discussing development of metrics that fit the culture of the organization. Traditional CP measures have included:

- How much money is spent on hot sites?
- How many people are devoted to CP activities?
- How many adverse audit comments have been brought to management's attention?

Instead, the focus should be on measuring the CP process contribution to achieving the overall goals of the organization, as mentioned in the ERM discussion. This focus helps us to:

- Identify agreed-upon CP development milestones
- Establish a baseline for execution
- Validate CP process delivery
- Establish a foundation for management satisfaction to successfully manage expectations

The *CP Balanced Scorecard* includes a definition of the:

- Value Statement
- Value Proposition
- Metrics and assumptions on reduction of CP risk
- Implementation Protocols
- Validation Methods

Following this Balanced Scorecard[6] approach, and aligning development of the scorecard with the ERM business and risk drivers mentioned earlier, the organization could define what the future-state of the CP process should look like. This future-state definition should be co-developed by the organization's top management and those responsible for development of the CP process infrastructure. Current State/Future State Visioning is a technique that can also be used for developing expectations for the Balanced Scorecard. Once the future-state vision is defined, the CP process development group can outline the CP process implementation critical success factors in the areas of:

- Growth and innovation
- Customer satisfaction
- People
- Process quality
- Financial state

These measures must be uniquely developed based on the specific organization's culture and environment.

Next Steps

What can the CP professional do within his organization to begin considering the feasibility of shifting the continuity planning processes under the ERM umbrella? One suggestion might be to identify the Enterprise Risk Committee or other suitable risk management organizational components within the company and initiate discussions relative to some of the issues raised in this chapter. In addition, depending on the industry group your organization is in, there may well be industry leading practices or examples of other organizations that have undertaken this course of action. You may well be able to profit from the experiences of others. There are professional societies such as the Risk and Insurance Managers Society, Inc. (http://www.rims.org/) and the Institute of Internal Auditors (http://www.theiia.org) where additional information can be obtained on this subject.

Summary

The failure of organizations to measure the success of their CP implementations has led to what seems like an endless cycle of plan development and decline. The chief reason for this cycle is that a meaningful set of CP measurements that complement the organization's business drivers have not been adopted. Because these measurements are lacking, expectations, reasonable or otherwise, of both executive management and those responsible for CP often go unfulfilled. Statistics gathered in the Contingency Planning and Management/KPMG Continuity Planning Survey support this assertion.

A true understanding of business objectives and their value-added contributions to overall business goals is a powerful motivator for achieving success on the part of the CP manager. There are many value drivers of strategic (competitive forces, value chains, key capabilities, dealing with future value, business objectives, strategies and processes, performance measures, etc.), financial (profits, revenue growth, capital management, sales growth, margin, cash tax rate, working capital, cost of capital, planning period and industry-specific subcomponents, etc.), and operational value (customer or client satisfaction, quality, cost of goods, etc.) that the CP professional should focus on, not only during the development of successful continuity planning strategies, but also when establishing performance measurements.

This chapter has introduced the role of continuity planning business processes in supporting an enterprise view of risk management, and to highlight how, working in harmony, the ERM and CP functions can provide measurable value to the enterprise, people, technologies, processes, and mission. It is incumbent upon continuity planning managers and enterprise risk managers to search for a way to merge efforts to create a more effective and efficient risk management structure within the enterprise.

Acknowledgment

Special thanks go to Mark Carey, President, DelCreo, Inc., for his valuable contributions to this chapter.

Business Continuity in the Distributed Environment

Steven P. Craig

This chapter describes the process of business recovery planning with an emphasis on the considerations for LANs and the components that comprise the LAN. The considerations of this chapter can be applied to companies of any size with a recovery scope from operational to catastrophic events.

INTRODUCTION

Today's organizations, in their efforts to reduce costs, are streamlining layers of management while implementing more complex matrices of control and reporting. Distributed systems have facilitated the reshaping of these organizations by moving the control of information closer to its source, the end user. In this transition, however, secure management of that information has been placed at risk. Information Technology Departments must protect the traditional system environment within the computer room plus develop policies, standards, and guidelines for the security and the protection of the company's information base. Further, the information technology staff must communicate these standards to all users to enforce a strong baseline of controls.

In these distributed environments, information technology personnel are often asked to develop system recovery plans outside the context of an overall business recovery scheme. Recoverability of systems, however, should be viewed as only one part of business recovery. Information Systems, in and of themselves, are not the lifeblood of a company; inventory, assets, processes, and people are all essential factors that must be considered in the business continuation design. The success of business continuity planning rests on a company's ability to integrate systems recovery in the greater overall planning effort.

BUSINESS RECOVERY PLANNING — THE PROCESS

Distinctive areas must be addressed when formulating a company's business disaster recovery plan that follow the stages of the scientific process, namely; the statement of the problem, development of an hypothesis, and testing of the hypothesis. Most importantly, as with any scientifically developed process, the Disaster Recovery Planning Process development is iterative! The testing phase of this process identifies whether or not the plan will work in practice, not just in theory. It is imperative that the plan and its assumptions be tested, tested, and re-tested. The important distinction about disaster recovery planning, and the importance of its viability, is what is at stake — namely the survivability of the business!

The phases of a viable disaster recovery plan process are

- Awareness and Discovery
- Risk Assessment
- Mitigation
- Preparation
- Testing
- Response and Recovery

Some of these phases may be combined, depending on the size of the company and the extent of exposure to risk. However, these phases are distinct and discussed more in length in the following sections.

Awareness and Discovery

Awareness begins when a recovery planning team can identify both possible threats and plausible threats to business operations. The more pressing issue for an organization in terms of business recovery planning is that of plausible threats. These threats must be evaluated by recovery planners and their planning efforts, in turn, will depend on these criteria:

1. The business of the company.
2. The area of the country in which the company is located.
3. The company's existing security measures.
4. The level of adherence to existing policies and procedures.
5. Management's commitment to existing policies and procedures.

Awareness is also education! Part of the awareness process consists of instructing all employees on what exposures exist for the company and themselves; what measures have been taken to minimize those exposures; and what their individual roles are in complying with those measures.

Pertaining to systems and information: what exposures are there; what information is vital to the organization; and, what information is proprietary and confidential to the business? Also with respect to systems,

another question that needs to be addressed is, when is an interruption considered to be catastrophic as opposed to operational? Again, this needs to be answered on a company-by-company basis. In an educational environment the systems being down for two to three days may not be considered catastrophic, however, in a process control environment (e.g. chemicals or electronics) a few minutes of down time might be considered catastrophic.

Discovery is determining the extent of the exposure and the extent of recovery planning and of the security measures that should be taken. Based on the response to the awareness question, what is plausible; there are more questions to be asked: what specific operations would be impacted by the exposures; what measures are in place or could be put in place to minimize those exposures; and, what measures could be taken to remove the exposure?

Risk Assessment

Risk assessment is a decision process that weighs the cost of implementing preventative measures against the risk of loss from not taking any action. There are qualitative and quantitative approaches to risk analysis of which there are full text references written on the subject. Typically for the systems environment, in terms of outright loss, two major cost factors arise. The first is the loss from not conducting business due to system down time. The second is the replacement cost of the equipment. The unavailability of systems for an extended period of time is the easiest intuitive sell, as it is readily understandable by just about everyone in today's organizations as to how much they rely on systems.

The cost to replace systems and information, however, is often not well understood, at least not from a catastrophic loss point of view. In many instances, major organizations, when queried on insurance coverage for systems, come up with some surprising results. There will typically be coverage for mainframes and mid-range systems, as well as coverage for the software for these environments, but when it comes to the workstations or the network servers they are deemed as not worth enough to insure. Another gaping hole is the lack of coverage for the information itself. The major replacement cost for a company is the recreation of its information base.

Further, the personal computer (PC), no matter how it is configured or what it is hooked up to or how extensive the network, is still perceived to be a stand alone unit from the risk assessment point of view. Even though many companies have retired their mainframes and fully embraced an extensive client/server architecture to fully manage their businesses, and fully comprehend the impact of the loss of its use, they erroneously look at

the replacement cost of the unit rather than the distributed system as the basis of risk.

Risk Assessment is the control point of the recovery planning process. The amount of exposure a company believes it has, or is willing to accept, determines how much additional effort the company will put forth on this process. Quite simply, a company with no plan is taking on the full risk of exposure, assuming that nothing, at least nothing severe, will ever happen to them. Companies that have developed plans have decided on the extent of risk assumption in two ways: (1) they have identified their "worst case scenario"; (2) they have made decisions based on how much they will expend in offsetting that scenario through mitigation, contingency plans, and training. Risk Assessment is the phase required to gel a company's management perspective, which in turn supports the goal to develop and maintain a company-wide contingency plan.

Mitigation

Mitigation has two primary objectives: lessen the exposures and minimize possible loss. History teaches us several lessons in this area. You can be sure that companies in Chicago now think twice about installing data centers in the basement of buildings after the underground floods of 1992. Bracing of key computer equipment and of office furniture has become popular in California due to the potential injuries to personnel and the threat of loss of assets from earthquakes. And, forward thinking companies in the South and Southern Atlantic states are installing systems far from the exterior of the buildings and windows because of the potential damage due to hurricanes.

Once again, from a more operational perspective, you can read story after story in the trade journals about back-up schemes gone awry, if there was a back-up performed at all! Although it is a simple concept, to make a back up copy of key data and systems, it is a difficult one to enforce in a distributed systems environment. To wit, as systems have been distributed and the end-user has been empowered, the regimen of daily or periodic back ups has diminished. The end-user has been empowered with the tools but not given the responsibility that goes along with the use of those tools. I recently went into a company, one of the leaders in the optical disk drive market, and found that it did perform daily backups to optical disk (using its own product) of its accounting and manufacturing systems; but they never rotated the media and never thought to take it off site! Any event impacting the hardware (e.g., fire, theft, earthquake) would have also destroyed the "only backup" and the means of business recovery for this premier company.

Preparation

This phase of your disaster planning process delineates what must be done in addition to the mitigation taken, should an event occur. Based on the perception of what could happen; who will take what actions? Are alternates identified for key staff members that may have been injured as a result of the event? Can the building be occupied, if not, where will temporary operations be set up? What supplies, company records, etc., will be required to operate from a temporary facility? What computer support will be required at the temporary location? Will a hot site be used for systems and telecommunications? What vendors and services providers need to be contacted; and further, do you have access to their off-hours phone numbers, emergency numbers, or home phone numbers? These are all questions that need to be addressed, contingencies established, and the plans documented as an integral part of your disaster preparedness process.

Testing

As mentioned above, the testing phase proves out the viability of your planning efforts. If there are omissions in your plan, or invalid assumptions, or inadequately postulated solutions... this is where you want to find these things out! Not at the time of an actual event! Additionally, organizations do not remain static; the elements of change within an organization and its environment dictate a reasonable frequency of testing. This is the phase of your plan you must afford to reiterate until you are comfortable with the results and that your plans will work in time of crisis. Section 3.0 covers testing more in-depth and proposes a testing strategy made available by the use of distributed systems.

Response and Recovery

Most of us carry auto insurance, home insurance, professional liability insurance and life insurance, yet we hope we'll never have to use it or rely on it. Well, this is the phase of your contingency plan you hope you never have to use! This part of your plan details what individuals will take on specific roles as part of predetermined teams, trained to address the tasks of: emergency response, assessment of damage, clean-up, restoration, alternate site start-up, emergency operations center duties and whatever else managing through your crisis might demand.

Every phase of the planning process, prior to this phase, is based on normalcy. The planning effort is based on what is perceived to be plausible. Responses are envisioned to cover those perceptions, and are done so under rational conditions. Remember that people are an integral part of the response and recovery effort. Dealing with a catastrophic crisis is not a normal part of everyday life or of someone's work load.

You can expect very different reactions from individuals, you may think you knew well, under severe stress. A simple example, you may have experienced yourself, is being trapped in an elevator for several minutes. Within a couple of minutes, individual's personalities, anxieties, and fears start to surface. Some will begin to panic, others will start taking control of the situation. Here again, testing the plan may afford you some insight as to how your team members will react. Ideally you will be able to stage some tests that will involve "role playing" so as to give your team members a sense of what they may be exposed to and the conditions they will have to work under.

DEPARTMENTAL PLANNING

Time and time again I will be asked to help a company develop its business resumption plan, only to be asked to focus just on the systems and ignore everything else; for the most obvious reason — cost. As it turns out, if a company receives an action item as a result of an audit, it is typically a part of an EDP audit and thus only targeted at the systems of a company. In turn, the company focuses only on the audit compliance, thus viewing disaster recovery as an expense, rather than the view of being an investment in business continuity.

Having a plan which addresses data integrity and systems survivability is a good start, but there is a lot more to consider. Depending on the nature of the business, telecommunications availability, as an example, may be much more important than systems availability. In a manufacturing environment, if the building and equipment were to be damaged, getting the systems up and running would not necessarily be the most important priority.

A company's Business Continuation Plan is, in fact, a compilation of its departmental plans. It is essential that each department identify its own processes and subsequent priorities of those processes. Overall company-wide operating and recovery priorities are then established by the company's management based on the input supplied by the departments. Information Technology, as a service department to all other departments, is subsequently in a much better position to plan recovery capacity and required system availability based on their inputs, priorities, and departmental recovery schedules.

INFORMATION TECHNOLOGY'S ROLE

Information Technology should not be responsible for creating the individual departmental plans for the rest of the company, but it can and indeed needs to take a leadership role in the departmental plan development. Information Technology has generally been the department that has the best appreciation and understanding of information flow throughout

the organization. It is therefore in the best position to identify and assess the following areas.

Inter-Departmental Dependencies

Many times in reviewing a company's overall plan and its departmental plans and their subsequent priorities, conflicts in the priorities will arise. This occurs because the departments tend to develop their plans on their own without the other departments in mind. One department may down play the generation of certain information, knowing it has little importance to its own operations, but it might be a vitally important input to the operation of another department. Information Technology can typically identify these priority discrepancies simply by being able to review each of the other department's plans.

External Dependencies

During the discovery process, recovery planners should determine with what outside services end-user departments are linked. End-user departments often tend to think of external services as being outside the scope of their recovery planning efforts, despite the fact that dedicated or unique hardware and software are required to use the outside services. At a minimum, make sure the departmental plans include the emergency contact numbers for the services and any company account codes that would permit linkage to the service from a recovery location. Also inquire as what provisions the outside service provider may have to assist your company in its recovery efforts.

Outsourced Operations

A 1990s trend in corporate strategic directions has been the outsourcing of entire department operations. The idea is to focus the company's resources on what it does best, and outsource the functions that it believed other companies could better handle as part of their expertise and focus. The idea sounds good in theory, but in practice this has been a mixed bag of tricks. The bottom line of this strategic direction was that it would add to the bottom line. Based on what is being published on the subject, the savings may only be a short-term result, and in fact be very costly in the long run. From a contingency planning perspective, what happens if the idea does not work; how does a company rebuild an Information Systems Department from scratch?

With respect to recovery planning, this is a key area that requires involvement at the earliest stages possible, including the review of contract wording and stipulations. This is an area in which the contractor has to be an integral partner, with as much ownership and jointly owned risk as the acquiring company. In many disasters, the Information Systems staffs

are the first responders for business recovery; will the contractor be as willing to take on this role? The recovery planner needs to validate that the on-site outsourced contractors are as well trained on response and recovery as the other internal departments. The area of systems is so integral to the recovery capability of the other departments that is it imperative that the outsourced information systems personnel be well versed in the recovery needs and response priorities of all of the departments they are there to support.

Collectively, the outsourcer may have considerably more resources available to it than the customer; however, it must be agreed to contractually that the contractor will bring its resources to bear in the event of the customer's catastrophe. Normally these outsourced arrangements start off with the greatest of intentions, but once things get under way and the conditions of systems, documentation, and operations are established — anything outside the scope of the contract is doable, but with incremental cost. Costs were what was intended to be cut when the outsourcing direction was decided upon, upping these costs will be a tough sell. So the recovery planner has to be involved early in the development of any such outsourcing contract and be sure to protect the company's contingency planning interests.

Internal and External Exposures

Stand-alone systems acquired by departments for a special purpose are often not linked to a company's networks. Consequently, they are often overlooked in terms of data security practices.

For example, a mortgage company funded all of its loans via wire transfer from one of three standalone systems. This service was one of the key operations of the company. Each system was equipped with a modem and a uniquely serialized encryption card for access to the wire service. As you might guess, these systems were not maintained by Information Technology; there were no data or system back-ups maintained by the end-user department; and, each system was tied to a distinct phone line. Any mishap involving those three systems could have potentially put this department several days, if not weeks, in arrears in funding its loans. A replacement encryption card and linkage establishment would have taken as much as a month under catastrophic conditions to re-establish.

As a result of this discovery, a secondary site was identified and a standby encryption card, an associated alternate phone line and a disaster recovery action plan were filed with the wire service. This one finding and its resolve more than justified the expense of the entire planning effort.

Another external exposure was identified for the same company during the discovery process dealing with power and the requirements of its UPS

capabilities. The line of questioning was on the sufficiency of battery back up capacity and whether an external generator should be considered as well for longer term power interruption. An assumption had been made that even in the event of an area wide disaster that power would probably be restored within 24 hours. The company had 8 hours of battery capacity which would suffice for the main operational shift of the company. At first I was in agreement with them, knowing that the county's power utility company had a program of restoring power on a priority basis for the larger employers of the county. When I mentioned this observation to them, I was corrected! They were in a special district and actually acquired their power from the city; and as a business would have power restored only after all the emergency services and city agencies were restored. The restoration period was unknown! The assumption of power restoration within 24 hours was revised and an external generator was added to the uninterruptable power supply system.

Systems themselves should not be the only type of exposure looked for. In a recent client discovery walk-through, a protracted construction project was underway. The existing computer room (on the eighth floor of a twenty story high rise) was being remodeled to house the company's latest generation of computers and telecommunications equipment. The room had originally been designed with standalone air conditioners, a UPS system, secured entry and a raised floor. Sprinklers had been eliminated from the room to avoid potential water damage and a Halon fire suppression system had been installed.

As a result of the construction, the computer equipment was temporarily moved to the adjoining computer technician's room. As you might guess, the technician's room had none of the protections that had been developed for the computer room. However, while there were short-term exposures (for length of the construction period) this was a known calculated risk. The actual exposure discovered was the computer room itself. During construction, the Halon fire suppression system and alarms had been turned off, as well as the stand alone air conditioning systems within. In addition a considerable amount of packing material had been accumulated within the room, so much so that it was stacked from floor to ceiling. The room was hot, from the lack of air conditioning. This was a fire waiting to happen. A fire needs fuel, oxygen, and heat _ all three readily existed in the room. If a fire were to start there were no active fire suppression capabilities within the room and with the alarms being turned off, it would have been well under way before the other building detection systems would have been alerted. A fire located here would have easily knocked out the central computing capability and telecommunications for the entire corporation as well as potentially destroying several floors of this corporate tower. Transition periods can be the times of greatest vulnerability for any

company, as existing detection and protection systems are temporarily shut down. The recovery planner needs to know that the planning process is reiterative, if the assumptions of the plan change, a review of all of the process steps is in order.

Apprise Management of the Risk

It is entirely management's decision on how much risk they are willing to take or deem what risks are unacceptable. However, as Information Technology identifies the various risks, it is their responsibility to make management aware of those risks. This holds true across the board on all security issues, be they system survivability issues (disaster recovery) or confidentiality or system integrity issues.

A company having its key system client files breached from the outside or a sales representative's laptop stolen with those key client files contained within, can be potentially more devastating to a company's operations than a prolonged power outage.

Apprise Management of Mitigation Cost

I find a tremendous amount of frustration in Information Technology departments these days, as departments have been "right-sized" and yet have to manage more complex systems than ever before. Many of the things that you will uncover will have such an obvious risk that obtaining approval for your mitigation campaigns should be relatively easy to obtain. Other system related topics are more intangible or in some cases deemed as being a "nuisance" are admittedly a tougher sell.

To cope with today's organizational demands and yet still feel "good" about the job it is performing, the Information Technology personnel responsible for this planning effort has to adapt to the changing times, anticipate the risks, and present to management the mitigation options and their associated costs; knowing that management will make a decision with the company's best interest in mind.

Policies

The best approach to begin an implementation of a system or data safeguard strategy is to first define and get approval from management on the policy or standard operating procedure that requires the safeguard be established. In assisting a community college in putting together a disaster recovery plan for its central computing operations, we discovered numerous departments had isolated themselves from the networks supported by the Information Technology group. The reason for this departure was the belief that the servers were always crashing, which was a cause for concern some three years ago, but no longer true. Yet to date, these

departments including Accounting, were processing everything locally on their hard drives with no back-ups whatsoever! This practice, now three years old, needed to be dispelled, as a disaster such as a fire in the Accounting Department would severely disrupt if not cause a cessation of the college's operations altogether. One of the other satellite campuses of the district, went entirely its own route and set up its own network with no ties to the central computing facility, and you guessed it, absolutely no back-ups at all!

We subsequently went back to the fundamentals; distribute the responsibility for data integrity along with the distributed system capability. A college policy statement on data integrity was made to the effect:

The recoverability and correctness of digitized data, which resides on college owned computer systems and media, is the responsibility of the individual user. The ultimate responsibility of ensuring the data integrity for each departmental workstation rests with the department/division administrator.

Information Technology will provide the guidelines for data back-ups. Adherence to these guidelines by the users of the college owned workstations is mandatory.

Establish Recovery Capability

Based on the inputs from the departments of the company and the company's overall priorities, Information Technology is challenged with designing an intermediate system configuration that is adequately sized to permit the company's recovery, immediately following the event. This configuration whether it be local, at an alternate company site, or a hot site needs to initially sustain the highest priority applications, yet be adaptable to expand to address other priorities; depending on how long it make take to re-occupy the company's facilities and fully restore all operations back to normal. You'll need to consider, for example, that the key Client/Server Applications may be critical to company operations whereas office automation tools may not.

Restore Full Operational Access

Information Technology's plan also needs to address the move back from an alternate site and what resources will be required to restore and resume full operations. Depending on the size of the enterprise and the disaster being planned for, this could include hundreds to thousands of end-user workstations. At a minimum, this step will be as complex as a move of your company to a new location.

PLANNING FOR THE DISTRIBUTED ENVIRONMENT

First and foremost, what are your marching orders? What is the extent to which your plan is to cover? Is it just the servers? Is it just the computers directly maintained by the Information Technology Department? Or is it the entire enterprise's systems and data that your are responsible for? Determining the extent of recovery is your first step, i.e., defining the scope of the project. The project scope, the overall company priorities, and the project funding will bracket the options you have in moving forward. But what follows in the next sections are some of the basics no matter what your budget. As you read through them, you'll find many of these ideas are founded in sound operational management, as they should be.

Protecting the LAN

There are two primary reasons why computer rooms are built: one, to provide special environmental conditions; and two, for control. Environmental conditions include: air conditioning, fire rated walls, dry sprinkler systems, special fire abatement systems (Halon, FM-200), raised flooring, cable chase-ways, equipment racking, equipment bracing, power conditioning, and continuous power (UPS systems), etc. "Control" includes a variety of factors, namely: access, external security, and internal security. All these aspects of protection (mitigation steps taken to offset the risk of fire, theft, malicious tampering, etc.,) were built-in benefits of the computer room. Yet if one walks around company facilities today, they will find servers and all sorts of network equipment on desk tops in open areas, on carts with wheels, in communication closets that are unlocked or with no conditioned power — yes, they're truly distributed and open! What's on those servers or accessible through those servers… just about anything and everything important to the company.

Internal Environmental Factors. A computer room is a viable security option, though there are some subtleties to designing one specifically for a client/server environment. If the equipment is to be all rack mounted, racking can be suspended from the ceiling, which still yields clearance from the floor avoiding possible water damage. Notably, the cooling aspects of a raised floor design, plus its ability to hide a morass of cabling are no longer needed in a distributed environment.

Conditioned power requirements have inadvertently modified computer room designs as well. If an existing computer room has a shunt trip by the exit but standalone battery backup units are placed on servers, planners must review their computer room emergency shutdown procedures. The idea of the shunt trip was to "kill all power" in the room, so that if operational personnel had to leave in a hurry, they would be able to come back later and reset systems in a controlled sequence. However, when

there are individual battery back-up units that sustain equipment in the room, the equipment connected to them will continue to run, even after the shunt is thrown, until the batteries run out!

Rewiring the room for all wall-circuits to run off the master UPS, in proper sequence with the shunt trip, is one way to resolve this conflict. However if the computer room houses mainframe, mid-range, and client/server equipment a different strategy might be required. Many of the client/server systems are designed to "begin" an orderly shut down once the cut over to battery power has been detected. This is not the case with all mid-range and mainframe systems.

There are instances when it would be better to allow an orderly shut down to occur, a short term power outage for example. While other times an instant shut off of all power would be required, as in the case of a fire or an earthquake.

The dilemma rests with the different requirements of the system platforms; the solution lies in the wiring of the room. One option is to physically separate the equipment into different rooms and wire each room according to the requirements of the equipment it contains. Another solution is a two-stage shunt approach: a red shunt would immediately shut off all power, as was always intended; a yellow shunt would cut all power except from the UPS, allowing the servers to initiate an orderly shut down on their own.

Room placement within the facility is also a consideration as pointed out earlier. If designing a room from scratch, identify an area with structural integrity, avoid windows, and eliminate overhead plumbing.

Alternate fire suppression systems are still a good protection strategy for all the expensive electronics and the operational, on-site tape back-ups within a room. If these types of systems are beyond your budget, consider multiple computer rooms (companies with a multiple building campus environment or multiple locations can readily adapt this as a recovery strategy). Equip the rooms with sprinklers; and keep some tarpaulins handy to throw over the equipment to protect the equipment from incidental water damage (a broken sprinkler pipe for example). A data safe may also be a worthwhile investment for the back-up media maintained on-site. However, if you go through the expense of using a safe, train your personnel to keep it closed! Eight out of ten site visits where a data safe is used, I'll find the door ajar (purely as a convenience). The safe only provides the protection to your media when it is sealed. If the standard practice is to keep it closed, then the personnel won't have to second guess, under the influence of adrenaline, whether or not they shut it as they evacuated the computer room.

If your company occupies several floors within a building and you maintain communication equipment (servers, hubs, modems, etc.) within the

closets; then treat them as a miniature computer room as well. Keep the doors to the closets locked and equip the closet with power conditioning and adequate ventilation.

Physical Security. The other aspect of a secured computer room was "control." Control (both internal and external to the company) of access to the equipment, cabling, and back-up media. Servers out in the open are prime targets for a range of mishaps from "innocent" tampering to outright theft. A thief, in stealing a server, not only gets away with an expensive piece of equipment but a potentially great amount of information; which, if the thief realizes it, may be several times more valuable and marketable than the equipment.

I mentioned earlier a college satellite campus that had no back-ups of the information contained within its network. I had explained to that campus administration, which by the way kept their servers out in the open of their administration office area that was in a temporary trailer, that a simple theft (equipment with a street value of $2000) would challenge their viability of continuing to operate as a college. All their student records, transcripts, course catalogs, instructor directories, financial aid records and more were maintained on their servers. With no back-ups to rely on and their primary source of information evaporated they would be faced with literally thousands of hours to re-construct their information bases.

Property Management. Knowing what and where the organization's computer assets (hardware, software, and information) are, at any moment in time, is critical to your recovery efforts. This may sound blatantly obvious, but remember we're no longer talking about the assets just within the computer room. Information Technology needs to be aware of: every workstation used throughout the organization, whether it is connected to a network or not (this includes portables); what its specific configuration is; what software resides on it; and, what job function it supports. This is readily doable, if all hardware/software acquisitions and installations are run through your department; and , the company's policies and procedures support your control (meaning that all departments and all personnel willingly adhere to the policies and procedures), and your property management inventory is properly maintained. Size is a factor here. If you manage an organization with a single server and fifty workstations, you may not deem this too large a task; however, if you support several servers and several hundred workstations, then you'll appreciate the amount of effort this can entail.

Data Integrity. Information is the one aspect of a company's systems that cannot be replaced, if lost or destroyed, simply by ordering another copy or another component. You can have insurance, "hot-site" agreements or

quick replacement arrangements for hardware and global license agreements for software, but your data integrity process is entirely up to you! You, as the Information Technology Specialist and the Disaster Recovery Planner are the individual that needs to insure the company's information will be recoverable when needed. It all goes back to the risk of loss as to how extensive a data integrity program you need to devise; from policies, to frequency of back-ups, to storage locations, to retention schedules, to the periodic verification that the back-ups are being done correctly. If you are just starting your planning process, this should be the first area you focus your mitigation efforts on. None of the other strategies you'll implement will count if there is no possible recovery of the data.

Network Recovery Strategies

As Information Technology your prime objective with respect to systems contingency planning is system survivability. This means that you have provisions in place, albeit limited capacity, to continue to support the company's system needs for priority processing through the first few hours immediately following the disaster.

Fault Tolerance vs. Redundancy. To a degree what we're striving for is fault tolerance of the company's critical systems. Fault tolerance, means that no single point of failure will stop the system. This is many times built in as part of the operational component design of the system. Examples include: mirroring of disks, use of RAID systems, shadowed servers, and UPS's to multiple T1's for wide area communications. Redundancy, duplication of key components, is the basis of fault tolerance. Where fault tolerance can not be built in, a quick replacement or repair program needs to be devised. Moving to an alternate site, either one of your company's, or a facility that is under contract for emergency support, i.e., a hot site, is a quick replacement strategy.

Alternate Sites and System Sizing. Once the priorities of a company are fully understood, sizing the amount of system capacity required to support those priorities, in the first few hours, through the first few days and weeks after a disaster can be accomplished. If you plan for you own recovery site, using another company location, or establish a contract with a "hot-site" service provider, you will want to adequately size the immediate recovery capacity. This is extremely important, as most hot-site service providers will not allow you to modify your requirements once you've declared a disaster.

The good news with respect to distributed systems, is that the hot-site service providers offer you options for recovery: from using their recovery center; to bringing self-contained vans to your facility, equipped with your required server configuration; to shipping you replacement equipment for what was lost, assuming your facility is still operable.

Adequate Backups with Secure Off-site Storage. This process must be based on established company policies that identify vital information and detail how its integrity will be managed. The work flow of the company and the volatility of its information base will dictate the frequency of back-ups. At a minimum, backup should occur daily for servers; and, weekly or monthly for key files of individual workstations.

Workstation based information continues to be one of the greatest vulnerabilities for most companies. There is so much vital information stored locally on these workstations with little or no backup. If individuals have taken the precaution of creating backups, they are typically stored right next to the workstations, leaving the company exposed to any type of catastrophic disaster. The recovery planner must insist that the company proactively address this issue through policy and through providing the means for effective workstation backups.

Planners must decide when and how often to take back-ups off-site. Depending on a company's budget, off-site could be the building next door, a bank safety deposit box, the network administrator's house, the branch office across town, or a secure media vault at a storage facility maintained by a company that's in the business of "off-site" media storage. Once the company meets the objective of separating the backup copy of vital data from its source, it must address the accessibility of the off-site copy.

The security of the company's information is also of vital concern. Security has several facets: if at a branch office, where do they safeguard the copy; if at the network administrator's house where is it kept; and what about the exposure to the media during transit? There are off-site storage companies that intentionally used unmarked, nondescript vehicles to transport your company's backup tapes to and from storage. This makes a lot of sense as your information is valuable and in your attempt to secure it you don't want to be advertising who you are using and where your storing your complete system backups.

Several products have come to market (1998) which will assist the LAN Administrator with these backup issues. Several of the products offer highly compressed, encrypted backups of workstations and other servers. The compression techniques require very little in the way of bandwidth, so they even work very effectively in remote backups of laptops using the Internet. The concept of vaulting, running mirrored data centers in separate locations, has been implemented by larger corporations who traditionally had the means to invest in the communications capabilities and the system redundancy. This type of capability is now made possible through these new tools. It is possible today to either work with off-site storage vendors to remotely backup at their facility or if the company has multiple locations,

to readily implement vaulting at the client/server level. Either way recovery options are facilitated via dial-up access to key recovery systems and data.

Adequate LAN Administration. Keeping track of everything the company owns, with respect to its hardware, software, and information bases is fundamental to your company's recovery effort. The best aid in this area is a solid audit application that is periodically run on all workstations. This assists you in maintaining an accurate inventory across the enterprise as well as providing you a tool for monitoring software acquisitions and hardware configuration modifications. The inventory may be extremely beneficial for insurance loss purposes. It also provides you with accurate records for license compliance and application revision maintenance.

Personnel. The all too often overlooked area of systems recovery planning is the system's personnel. Will there be adequate system personnel resources to handle the complexities of response and recovery. What if a key individual is impacted by the same catastrophic event that destroys the systems? This event could cause a single point of failure.

An option available to the planner is to an "emergency staffing contract." A qualified systems engineer hired to assist on a key project that you never seems to get completed (e.g., the network system documentation) may be a cost-effective security measure. Once that project is completed to satisfaction, the company can consider structuring a contractual arrangement that, for example, retains the engineer for one to three days a month to continue to work on documentation and other special projects. The contract could also stipulate coverage for staff vacations and sick days and should guarantee the engineer will be available on an as needed basis should the company experience an emergency. The advantage of this concept, is that you maintain an effective resource that is well trained and versed on your company's systems should you need to rely on them during an emergency; you have coverage for the company during employee's personal leaves; and, you have your systems documented!

TESTING

The timeless adage with regards to a business's success being "location, location, location," is adapted here. The pro forma success of a business's recovery plan will be most influenced by the extent of the "testing, testing, testing" of its plan! Testing and training are the reiterative and necessary components of the planning process that keep the plan up-to-date and maintain the viability of recovery.

Tests can be conducted in a variety of ways; from desk checking, reading through the plan and thinking though the outcome, to full parallel system testing, setting up operations at a hot site or alternate location and have

the users run operations remotely. The full parallel system test does generally prove out that the hot site equipment and remote linkages work but doesn't necessarily test the feasibility of the user-department's plans, as it is a system test. Full parallel testing is also generally staged with a limited amount of time which adds the pressure of "getting it done" and "passing" because of the time restriction.

Advantages of the Distributed Environment for Testing

Distributed client/server systems because of their size and modularity permit a readily available, modifiable, and affordable system set up for testing. They allow for a testing concept that I coin, "cycle testing."

For those of you with a manufacturing background, this draws a direct parallel to cycle counting; a process whereby inventory is categorized by value and counted several times a year rather than a one time physical inventory. With cycle counting, inventory is counted all year long, with portions of the inventory being selected to be counted either on a random basis or on a pre-selected basis. Inventory is further classified into categories, such that the more expensive or critical inventory items are counted more frequently, and the less expensive items less frequently. The end result is the same as taking a one time physical inventory, in that by the end of a calendar year, all the inventory has been counted. However, the cycle counting method has several advantages: (1) Operations do not have to be completely shut down, while the inventory is being taken; (2) Counts are not done under the pressure of "getting it done" which can result in more accurate counts; (3) Errors in inventories are discovered and corrected as a part of the continuous process.

The parallels to cycle testing are straightforward. Response and recovery plan tests can be staged with small manageable groups, so as not to be disruptive to company operations. Tests can be staged by a small team of facilitators and observers, on a continual basis. Tests can be staged and debriefings held with out the pressure of "getting it done"; allowing the participants the time to fully understand their role and critically evaluate their ability to respond to the test scenarios and make necessary corrections to the plan. Any inconsistencies or omissions in a department's plan can be discovered and resolved directly amongst the working participants.

Just as the more critical inventory items can be accounted for on a more frequent basis, so can the crucial components required for business recovery, i.e., systems and telecommunications. With the wide spread use of LANs and client/server systems throughout companies today, the Information Systems department is afforded more opportunity to work with the other departments in testing their plans and… getting it right!

SUMMARY

Developing a business recovery plan is not a one time, static task. It is a process that requires the commitment and cooperation of the entire company. In order to perpetuate the process, Business Recovery Planning must be a company stipulated policy as well as a company sponsored goal. The organizations that adopt this company culture oriented posture are the ones whose plans are actively maintained and tested, and whose employees are well trained and poised to proactively respond to a crisis. The primary objective of developing a Business Resumption Plan is the survivability of the business.

An organization's Business Resumption Plan is, in fact, an orchestrated collection of its Departmental Response and Recovery Plans. Information Technology's plan is also a departmental plan, however, in addressing the overall coordination of the departmental plans, Information Technology is typically in the best position to facilitate the other departments' development of their plans. With respect to the continuing trend of distributed processing permeating throughout organizations, Information Technology can be of particular help in identifying the organization's inter-departmental information dependencies and external dependencies for information access and exchange.

There are some basic protective security measures that should be fundamental to Information Technology's plan, no matter what the scope of disasters being planned for. From operational mishaps, to industrial espionage, to area-wide disasters, you'll want to make sure the Information Technology plan addresses:

1. an adequate back-up methodology with off-site storage;
2. sufficient physical security mechanisms for the servers and key network components;
3. sufficient logical security measures for the organization's information assets, and
4. adequate LAN/WAN administration, including up-to-date inventories of equipment and software.

Lastly, in support of an organization's goal to have its Business Resumption Planning process in place to facilitate its quick response to a crisis, the plan must be sufficiently and reiteratively tested and the key team members sufficiently trained. When testing is routinely built into the planning process, it becomes the feedback step that keeps: the plan current; the response and recovery strategies properly aligned; and, the responsible team members postured to respond. Once a plan is established, testing is the key process step that keeps the plan viable. Plan viability equates to business survivability!

The Changing Face of Continuity Planning

Carl Jackson, CISSP, CBCP

To one degree or another, the information security professional has always had responsibility for ensuring the availability and continuity of enterprise information. While still the case, specialization within the availability discipline has resulted in the growth of the continuity planning (CP) profession and the evolution into full-time continuity planners by many former information security specialists. Aside from the growth and reliance upon E-business by most major worldwide companies, the events of September 11, 2001, and even the Enron meltdown have served to heighten awareness for increased planning and advanced arrangements for ensuring availability. The reality is that continuity planning has a changing face, and is simply no longer *recovery planning as usual*. This chapter focuses on some of the factors to be considered by continuity planning professionals who must advance their skills and approaches to keep up with swiftly evolving current events.

REVOLUTION

Heraclitus once wrote, "There is nothing permanent except change." The continuity planning profession has evolved from the time when disaster recovery planning (DRP) for mainframe data centers was the primary objective. Following the September 11 attacks and the subsequent calls for escalating homeland security in the United States, the pace of change for the CP profession has increased dramatically from just a few months prior to the attacks. In looking back, some of us who have been around awhile may reminisce for the good ole' days when identification of critical applications was the order of the day. These applications could be easily plucked from a production environment to be plopped down in a hot site somewhere, all in the name of preventing denial of access to information assets. In retrospect, things were so simple then — applications stood alone, hard-wired coax connectivity was limited and limiting, centralized change control ruled, physical security for automated spaces solved a multitude of

sins, and there were less than half a dozen vendors out there that could provide assistance. Ah, those were the days!

The kind of folks who performed disaster recovery tasks in those times were fairly technical and were usually associated with the computer operations side of the house. They tended to understand applications and disk space and the like, and usually began their disaster recovery planning projects by defining, or again redefining, critical applications. Of course, the opinion of the computer operations staff about what constituted a critical application and that of the business process owner many times turned out to be two different things.

Of late, especially since September 11, we have seen the industry shift from a focus strictly on computer operations and communications recovery planning to one where business functionality and processes are considered the start and endpoint for proper enterprisewide availability. This is the point where many continuity planners began to lose their technical focus to concentrate on understanding business process flow and functional interdependencies so that they could map them back to supporting resources that included IT and communications technologies. Some of us simply lost our technological edge, due to the time it took to understand business processes and interdependencies, but we became good at understanding business value-chain interrelationships, organizational change management, and process improvement/reengineering.

Exhibit 42-1 depicts the evolution of industry thinking relative to the passage from technical recovery to business process recovery. It also reflects the inclination by continuity planners to again focus on technologies for support of Internet-based business initiatives.

As organizations move operations onto the Web, they must ensure the reliability and availability of Web-based processes and technologies. This includes the assurance that trading partners, vendors, customers, and employees have the ability to access critical B2B (business-to-business) and B2C (business-to-customer) resources. This has been identified in recent security surveys (sources include Gartner Research, IDC, and Infonetics) that suggest the worldwide marketplace for Internet security solutions will reach somewhere around $20 billion by 2004. Included within the scope of the security solutions marketplace are myriad products that facilitate detection, avoidance, mitigation of, and recovery from adverse events.

THE LESSONS OF SEPTEMBER 11

For the past decade or so, continuity planners have been shifting the emphasis to business process planning as the starting point for any meaningful continuity planning exercise. The pace has accelerated within the

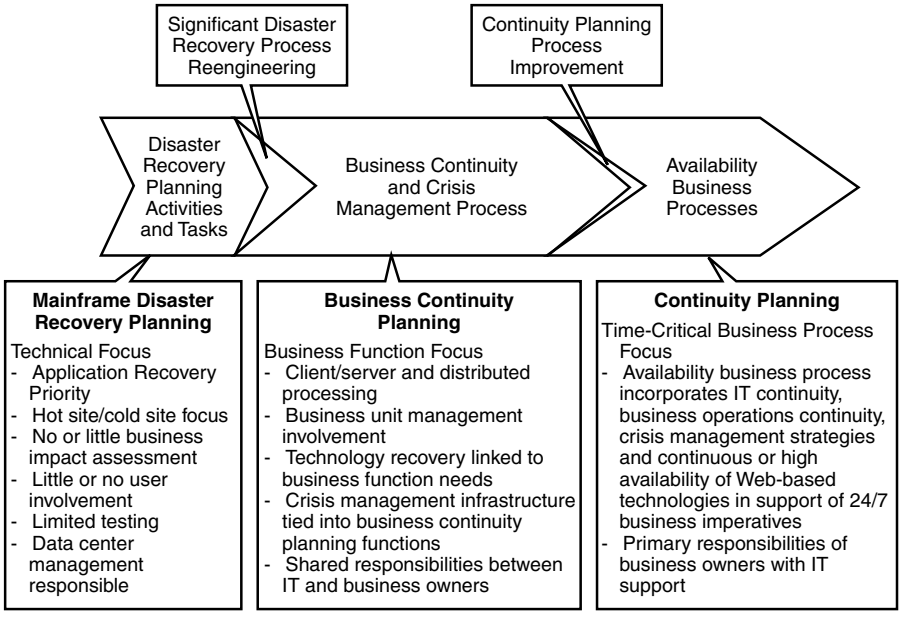


Exhibit 42-1. Evolution from technical recovery to business process recovery.

past five years, with E-business considerations driving shorter and shorter recovery time windows. But something happened following the September 11 attacks in the United States that appeared to redouble the speed of shifting focus for many of us.

We have all lived through much since the attacks of September 11. Our horror turned to shock and then grief for those souls lost on that day, and continues in military and related activities the world continues to undertake in response to these atrocities. As continuity planning professionals, we have a very unique view of events such as these because our careers so closely relate to mitigation and recovery from disruptions and disasters.

Call to Arms

The September 11 attacks raised the awareness level for the need for appropriate recovery planning in the United States and indeed the rest of the world. The U.S. Attorney General's call for companies to revisit their security programs in light of the terrorist attacks on U.S. properties should also serve to put executive management on notice — as if they needed any more incentives — that it may be time to rethink investments in their security and continuity planning programs.

There are no signs that the potential for disruptions caused by terrorist activities will be over anytime soon. In fact, it was recently made public

that the U.S. Government has activated its own continuity plans by establishing off-site operations for all three branches of government at secret locations outside of the Washington, D.C. area. These contingency plans were originally prepared during the Eisenhower administration in anticipation of nuclear attack during the Cold War, but they were thankfully never needed — until now. It is more than interesting to think that these long-prepared contingency plans had to be activated some 50 years later! I wonder if the folks who suggested that these plans be developed in the first place had to worry about cost justification or return on investment?

A Look at the Aftermath

The extent of the damage to the WTC complex alone was staggering. Even six months following the attacks, companies displaced by them continue to struggle. *The Wall Street Journal* reported on March 15, 2002, that of the many large companies impacted by September 11, numerous ones remain either undecided about moving back or have decided not to move back into the same area (see Exhibit 42-2). The graphic illustrates the destroyed and damaged buildings and lists some of the large companies located there.

This event displaced well over 10,000 employees of the hundreds of companies involved. It is estimated that in excess of 11 million square feet of space have been impacted.

There were many lessons learned from these tragic events. There are two areas that stick most in my mind as significant. First, it was the bravery of the people in reacting to the event initially and within a short period of time following the events; and second, it was the people who had to execute under duress on the many recovery teams that reacted to help their organizations survive. It was the people who made it all happen, not just the hot sites or the extra telecommunications circuits. That lesson, above all, must be remembered and used as a building block of future leading practices.

The Call for Homeland Security

From the mailroom to the executive boardroom, calls abound for increased preparations of your organization's responsibility in ensuring homeland security. Following September 11, continuity planners must be able to judge the risk of similar incidents within their own business environments. This includes ensuring that continuity planning considerations are built in to the company's policies for dealing with homeland security. Planners cannot neglect homeland security issues for their own organization, but they must also now be aware of the preparations of public- and private-sector partner organizations. Once understood, planners must interleave these external preparations with their own continuity and crisis management planning actions. In addition, continuity planners may want to

Some of the Biggest Firms

RETURNED	
Bank of New York	100 Church, 101 Barclay
Merrill Lynch	WFC 2,4

PLANS TO RETURN	
American Express	WTC 7, WFC 3, 40 Wall
Deloitte Touche Tomatsu	WFC 1, 2
Port Authority	WTC 1

PERMANENTLY RELOCATED	
Empire Blue Cross	WTC 1
Keefe Bruyette	WTC 2
Lehman Brothers	WFC 1
Marsh & McLennan	WTC 1
Morgan Stanley	WTC 2

Map of Trade Center Area and Location of Damage

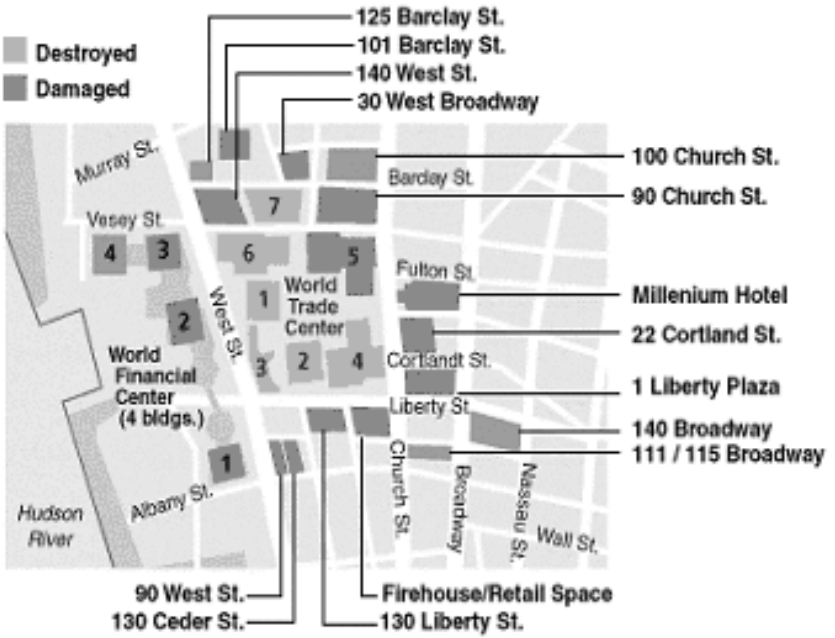


Exhibit 42-2. Plans to move back to Ground Zero. (Source: *The Wall Street Journal,* March 15, 2002.)

RED ALERT

The Bush administration unveiled a color-coded, five-level warning system for potential terrorist attacks. In the future, Attorney General Ashcroft will issue higher states of alerts for regions, industries, and businesses that may be the specific targets of terrorists.

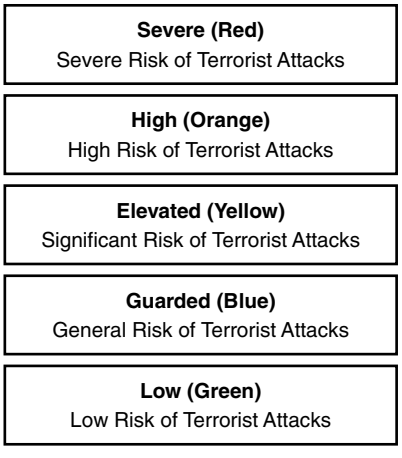


Exhibit 42-3. Alert system offered by the Office of Homeland Security. (Source: Office of Homeland Security.)

consider adoption, for crisis management purposes, of an alert system similar to the one offered by the Office of Homeland Security (see Exhibit 42-3).

The Importance of Education, Training, and Awareness

The results of the 2000/2001 CPM/KPMG Business Continuity Study Benchmark Report show that dismal attention has been paid by many companies to training, education, and awareness. When asked, "Do employees get sufficient disaster recovery/business continuity planning training?", of those answering the survey, 75 percent responded with *no* for the year 1999; and 69.5 percent said *no* for the year 2000. Unfortunately, I doubt that these percentages have improved to any significant degree, even since September 11.

People Must Be the Focus

People are important! Whether it is a life safety issue, or their participation in the recovery after the event, it is people who are most impacted by the disruption; and it is people who will have to recover following the disruption. All one has to do is look at case studies of the companies that had to recover following the attacks on the World Trade Center. For instance, in one sad case, all of the people who had participated in the most recent hot site test perished in the attack.

Planners simply should not allow haphazard education, training, and awareness programs to continue. These programs must be designed to teach the people how to protect themselves and the organization and to periodically refresh the message. *The single largest lesson that must be learned from September 11 is that the people must be the focus of all crisis management and continuity planning activities — not technology.* There is absolutely no question that technologies and their recovery requirements are vital, but technologies and processes are things that can be reconstructed or replaced. People cannot, as demonstrated by the loss of approximately 3000 souls on September 11.

What about Executive Protection and Succession Plans?

Not typically considered as part of the continuity planning responsibility, the events of September 11 call attention to the need for organizational management to revisit dated executive protection and succession plans and to test enterprise crisis management plans by challenging old assumptions based upon pre-September 11 thinking.

Business Process Continuity versus IT DRP

Another lesson learned was that, while many companies impacted by the events were able to recover automated operations, the vast majority of them were seriously disabled from a business process/operations standpoint. Their inability to physically transport people and supplies — given aircraft groundings — to off-site locations suitable for recovering business processes and supporting infrastructures (i.e., mail room operations, client/server configurations, purchasing, HR, back-office operations, etc.) illustrated that the practice of only preparing for IT recovery had resulted in a serious shortfall of preparations.

Security and Threats Shifting

There were many, many more companies seriously impacted than those located directly in the WTC buildings. Businesses all over the country, and indeed the world, that had critical dependencies upon the WTC-based companies were also injured by the event. Subsequent severe travel restrictions and the resulting economic downturn affected countless other organizations. Our highly interconnected world is much different than our world of just a few short non-Internet years ago. There are no islands in the global economy; and because the United States is the largest economic engine in that financial system, and because each U.S. company plays a role in that engine, it seems really rather shortsighted for major companies to not be making availability-related investments. Our risks have changed and shifted focus in addition to the ones mentioned above.

Others include:

- *Nuclear power plant security.* Recent media reports indicate that the U.S. Nuclear Regulatory Commission is unsure how many foreign nationals or security guards are employed at nuclear reactors and does not require adequate background checks of nuclear reactor employees that would uncover terrorist ties. There are 21 U.S. nuclear reactors located within five miles of an airport, 96 percent of which were not designed to withstand the crash of a small airplane.
- *Airport security.* It was recently reported (Fox News, March 25, 2002) that, according to a confidential February 19, 2002, Transportation Department memo, the department ran tests of security at 32 airports around the country that continued to be found lacking.
- *Border security.* There is focused attention on the increased security needs and staffing levels of border security staff along both the Canadian and Mexican borders to the United States, and President Bush is calling for consolidation of the INS and Customs Department.
- *Food and water supply security.* In connection with concerns over bioterrorism, Homeland Security is calling for consolidation of rival U.S. agencies responsible for food and water safety.
- *Internet security.* The U.S. Government is attempting to persuade industry to better protect the Internet from threats of cyber-crime and cyber-terrorism.
- *Travel security.* Key personnel residences and travel to unstable international destinations must be monitored and controlled appropriately.

Reassess Risk

As enterprise risk is assessed, through either traditional risk analysis/assessment mechanisms or through business impact assessments, understanding potential impacts from these expanded threats is essential and prudent. We must consider the impact of functionality loss that may occur either inside or outside our walls. These types of potential impacts include the direct ones, like those listed above, and those impacts that might disrupt an external entity that our organization relies upon — a supply-chain partner, key vendor, outsourcer, parent or subsidiary company, etc. Now is the time to go back and seriously consider the last time your organization performed a comprehensive risk assessment/business impact assessment, and think about updating it. Organizations change over time and should be reevaluated frequently.

THE LESSONS OF ENRON

Speaking of reliance on key external relationships, the Enron situation and its repercussions among the supply-chain partners, outsourcers, vendors, and supplier relationships continue to ripple through several industry

groups. Understanding your organization's reliance upon primary supply-chain partners and assorted others is crucial in helping you anticipate the breadth and scope of continuity and crisis management planning efforts — if for no other reason than for you to say that these issues were considered during preparations and not merely ignored. Granted, there is no question that, given the global level of the Enron-related events, it would have been challenging for those with internal continuity planning responsibilities to anticipate the extent of the impacts and to appropriately prepare for all contingencies. But in hindsight, it will be incumbent upon those who have responsibility for preparing continuity and crisis management plans to be at least aware of the potential of such events and be prepared to demonstrate some degree of due diligence.

COMPUTER FORENSIC TEAMS

The composition of crisis management and continuity planning teams is changing as well. Virus infestations, denial-of-service attacks, spoofing, spamming, content control, and other analogous threats have called for the inclusion of computer forensic disciplines into development of continuity planning infrastructures. Forensic preparations include understanding the procedures necessary to identify, mitigate, isolate, investigate, and prosecute following such events. It is necessary to incorporate enterprise forensic teams, legal resources, and public relations into continuity planning and crisis management response teams.

THE INTERNET AND ENTERPRISE CONTINUOUS AVAILABILITY

With growing Internet business process reliance on supporting technologies as the motivating force, continuity planners must once again become conversant and comfortable with working in a technical environment — or at least comfortable enough to ensure that the right technical or infrastructure personnel are involved in the process. The terminology currently used to describe this Internet resource availability focal point is *continuous* or *high availability*.

Continuous availability (CA) is a building-block approach to constructing resilient and robust technological infrastructures that support high-availability requirements. In preparing your organization for high availability, focusing on *automated applications* is only a part of the problem. On this topic Gartner Research writes:

> Replication of databases, hardware servers, Web servers, application servers, and integration brokers/suites help increase availability of the application services. The best results, however, are achieved when, in addition to the reliance on the system's infrastructure, the design of the application itself incorporates considerations for continuous availability. Users looking to achieve continuous availability

for their Web applications should not rely on any one tool but should include the availability considerations systematically at every step of their application projects.

— Gartner Group RAS Services
COM-12-1325, 29 September 2000

Implementing CA is easier said than done. The key to achieving 24/7 or near-24/7 availability begins with the process of determining business process owner needs, vulnerabilities, and risks to the network infrastructure (e.g., Internet, intranet, extranet, etc.). As part of considering implementation of continuous availability, continuity planners should understand:

- The resiliency of network infrastructures as well as the components thereof
- The capability of their infrastructure management systems to handle network faults
- The network configuration and change control practices
- The ability to monitor network availability
- Infrastructure single points of failure
- The ability of individual network components to handle capacity requirements, among others

Among the challenges facing continuity planners in CA are:

- Ensuring that time-critical business processes are identified within the context of the organization's Web-based initiatives
- Making significant investments in terms of infrastructure hardware, software, management processes, and consulting
- Obtaining buy-in from organizational management in the development, migration, and testing of CA processes
- Keeping continuous availability processes in line with enterprise expectations for their organization's continuity and crisis management plans
- Ensuring CA processes are subjected to realistic testing to assure their viability in an emergency

FULL-SCOPE CONTINUITY PLANNING BUSINESS PROCESS

The evolution from preparing disaster recovery plans for mainframe data centers to performing full-scope continuity planning and, of late, to planning for the continuous operations of Web-based infrastructure begs the question of process improvement. Reengineering or improving continuity planning involves not only reinvigorating continuity planning processes but also ensuring that Web-based enterprise needs and expectations are identified and met through implementation of continuous availability disciplines. Today, the continuity planning professional must

possess the necessary skill set and expertise to be able to effectively manage a full-scope continuity planning environment that includes:

- *IT continuity planning.* This skill set addresses the recovery planning needs of the organization's IT infrastructures, including centralized and decentralized IT capabilities, and includes both voice and data communications network support services. This process includes:
  - Understanding the viability and effectiveness of off-site data backup capabilities and arrangements
  - Executing the most efficient and cost-effective recovery alternative, depending upon recovery time objectives of the IT infrastructure and the time-critical business processes it supports
  - Development and implementation of a customized IT continuity planning infrastructure supported by appropriately documented IT continuity plans for each primary component of the IT infrastructure
  - Execution of IT continuity planning testing, maintenance, awareness, training, and education programs to ensure long-term viability of the plans, and development of appropriate metrics that can be used to measure the value-added contribution of the IT infrastructure continuity plans to the enterprise people, process, technologies, and mission
- *Business operations planning.* This skill set addresses recovery of an organization's business operations (i.e., accounting, purchasing, etc.) should they lose access to their supporting resources (i.e., IT, communications network, facilities, external agent relationships, etc.). This process includes:
  - Understanding the external relationships with key vendors, suppliers, supply-chain partners, outsourcers, etc.
  - Executing the most efficient and cost-effective recovery alternative, depending upon recovery time objectives of the business operations units and the time-critical business processes they support
  - Development and implementation of a customized business operations continuity plan supported by appropriately documented business operations continuity plans for each primary component of the business units
  - Execution of business operations continuity plan testing, maintenance, awareness, training, and education programs to ensure long-term viability of the plans
  - Development of appropriate metrics that can be used to measure the value-added contribution of the business operations continuity plans to the enterprise people, processes, technologies, and mission
- *Crisis management planning.* This skill set addresses development of an effective and efficient enterprisewide emergency/disaster response capability. This response capability includes forming appropriate

management teams and training their members in reacting to serious company emergency situations (i.e., hurricane, earthquake, flood, fire, serious hacker or virus damage, etc.). Key considerations for crisis management planning include identification of emergency operations locations for key management personnel to use in times of emergency. Also of importance is the structuring of crisis management planning components to fit the size and number of locations of the organization (many small plans may well be better than one large plan). As the September 11 attacks fade somewhat from recent memory, let us not forget that people responding to people helped save the day; and we must not ever overlook the importance of time spent on training, awareness, and education for those folks who will have responsibilities related to continuity following a disruption or disaster. As with IT and business operations plans, testing, maintenance, and development of appropriate measurement mechanisms is also important for long-term viability of the crisis management planning infrastructure.

- *Continuous availability.* This skill set acknowledges that the *recovery time objective* (RTO) for recovery of infrastructure support resources in a 24/7 environment has shrunk to *zero* time. That is to say that the organization cannot afford to lose operational capabilities for even a very short period of time without significant financial (revenue loss, extra expense) or operational (customer service, loss of confidence) disruptions. CA focuses on maintaining the highest possible uptime of Web-based support infrastructures, of 98 percent and higher.
- *The importance of testing.* Once developed and implemented, the individual components of the continuity plan business process must be tested. What is more important is that the people who must participate in the recovery of the organization must be trained and made aware of their roles and responsibilities. Failure of companies to do this properly was probably the largest lesson learned from the September 11 attacks. Continuity planning is all about people!
- *Education, training, and awareness.* Renewed focus on practical personnel education, training, and awareness programs is called for now. Forming alliances with other business units within your organization with responsibility for awareness and training, as well as utilizing continuity planning and crisis management tests and simulations, will help raise the overall level of awareness. Repetition is the key to ensuring that, as personnel turnover occurs, there will always be a suitable level of understanding among remaining staff.
- *The need to measure results.* The reality is that many executive management groups have difficulty getting to the bottom of the value-add question. What degree of value does continuity planning add to the enterprise people, processes, technology, and mission? Great question. Many senior managers do not seem to be able to get beyond the *financial justification* barrier. There is no question that justification of investment

in continuity plan business processes based upon financial criteria is important, but it is not usually the financial metrics that drive recovery windows. These metrics must be both quantitative and qualitative. It is the *customer service and customer confidence* issues that drive short recovery time frames, which are typically the most expensive. Financial justifications typically only provide support for them.

Implementation of an appropriate measurement system is crucial to success. Companies must measure not only the financial metrics but also how the continuity planning business process adds value to the organization's people, processes, technologies, and mission. These metrics must be both quantitative and qualitative. Focusing on financial measures alone is a lopsided mistake!

CONCLUSION

The growth of the Internet and E-business, corporate upheavals, and the tragedy of September 11 and subsequent events have all contributed to the changing face of continuity planning. We are truly living in a different world today, and it is incumbent upon the continuity planner to change to fit the new reality.

Continuity planning is a *business process*, not an event or merely a plan to recover. Included in this business process are highly interactive continuity planning components that exist to support time-critical business processes and to sustain one another. The major components include planning for:

- IT and communications (commonly referred to as disaster recovery planning)
- Business operations (commonly referred to as business continuity planning)
- Overall company crisis management
- And, finally, for those companies involved in E-business — continuous availability programs

In the final analysis, it is incumbent upon continuity planning professionals to stay constantly attuned to the changing needs of our constituents, no matter the mission or processes of the enterprise. The information security and continuity planning professional must possess the necessary skill set and expertise to effectively manage a full-scope continuity planning environment. Understanding the evolution and future focus of continuity planning as it supports our information security responsibilities will be key to future successes. As Jack Welch has said, "Change before you have to."

ABOUT THE AUTHOR

Carl Jackson, CISSP, CBCP, brings more than 25 years of experience in the areas of business continuity planning, information security, and IT internal control reviews and audits. As the vice president, continuity planning, for QinetiQ-Trusted Information Management Corporation, he is responsible for the continued development and oversight of QinetiQ-TIM (U.S.) methodologies and tools in the enterprisewide business continuity planning arena, including network and E-business availability and recovery.

References

1. Contingency Planning and Management/KPMG 2002 Business Continuity Planning Survey, *Contingency Planning and Management Magazine,* 2003.
2. *Enterprise Risk Management: Trends and Emerging Practices,* The Institute of Internal Auditors Research Foundation, 2001.
3. "What Is the Balanced Scorecard," www.balancedscorecard.org/basics/bsc1.html
4. Bennett Stewart, "About EVA," www.sternstewart.com/evaabout/whatis.php
5. H. James Harrington, Erick K.C. Esseling, and Harm Van Nimwegen, *Business Process Improvement Workbook,* McGraw-Hill, New York, 1997.
6. Robert S. Kaplan and David P. Norton, *Translating Strategy Into Action: The Balanced Scorecard,* HBS Press, 1996.

137

Restoration Component of Business Continuity Planning

John Dorf, ARM and Martin Johnson, CISSP

Everyone understands the importance of developing a business continuity plan (BCP) to ensure the timely recovery of mission-critical business processes following a damaging event. There are two objectives, however, and often, the second objective is overlooked return to normal operations as soon as possible. The reason for the urgency to return to normal operations is that backup and work-around procedures are certainly not "business as usual." Backup capabilities, whether due to the loss of primary premises or primary data, probably only include those business activities that are critical to getting by. The longer a company must operate in this mode, the more difficult the catch-up will be. There are several steps that can be taken in advance to prepare for the timely, efficient return to normalization. The purpose of this chapter is to discuss the steps and resources to ensure total recovery. In addition, it is important to understand how to handle damaged equipment and media in order to minimize the loss associated with a disaster.

Restoration includes the following:

1. Handling damaged equipment and media in order to minimize the loss
2. Salvaging hard copy and electronic media
3. Performing damage assessment and the resulting disposition of damaged facilities and equipment
4. Determining and procuring appropriate property insurance
5. Identifying internal and external resources to perform restoration activities
6. Developing, maintaining, and testing your restoration plan

This chapter will help you understand the issues related to each of these items and be a resource for developing the necessary information for inclusion in your BCP program.

The more time that passes before the salvation of hardcopy and electronic media, the greater the chance that the data or archival records will be permanently lost. However, if you rush to handle, move, dry, etc., media and do not do so in the correct manner, you may worsen the situation. Therefore, to ensure minimizing the damage you must act quickly and correctly to recover data and restore documents. This also applies to the facilities and infrastructure damage.

Having telephone numbers for restoration companies is not enough. The primary reason is in the event of a regional problem like flooding, ice storms, etc., you will have to wait for those companies that have advance commitments from other companies.

Another important issue associated with restoration is insurance. It is imperative you understand what is covered by your insurance policy and what approval procedures must be completed before any restoration work is performed. There are many stories about how insurance companies challenged claims because of disagreements concerning coverage or restoration procedures. Challenges from insurance carriers can hold up restoration for extended periods of time. Following are two examples showing the importance and magnitude of effort involved with restoration after a disaster.

> The 1993 World Trade Center bombing illustrates the potential magnitude of a clean-up effort. Over a 16-day period, 2700 workers hired by a restoration contractor, working round the clock in three shifts, cleaned over 880,000 square feet of space in the twin towers and other interconnected facilities. Ninety percent of the floors in the 110-story towers had light amounts of soot, while 10 percent suffered heavier damage.
>
> In 1995, Contra Costa County, California, suffered almost $15 million in arson-related fire damage to four county courthouses over a three-week period. In all, 124,000 files had to be freeze-dried and restored at an estimated cost of $50 per document.

A good restoration program will not guarantee you will not have a problem with your insurance carrier. The following is an example of how a disagreement between an insured and insurer can delay restoration of your business:

> In 1991, a 19-hour fire at One Meridian Plaza in Philadelphia destroyed eight of the 38 floors in the building. It took 6 years of legal maneuvering to settle the claim between the building owners and the insurers. Each party disagreed with the other over the extent of the restoration. For most of the 6 years, the parties' difference amounted to almost $100 million. The owners believed that the floors above the 19th floor had to be torn down because the steel beams supporting the structure had moved 4 inches and could not be certified as safe. The insurance company disagreed and argued that the building could be repaired without tearing down the floors. The owner and insurer also disagreed over the extent of environmental cleanup caused by the fire. Eventually, the matter was settled out of court for an undisclosed sum.

Understanding the Issues

For all damaged or destroyed property a company must understand when it needs to try to restore the property, and when the property can just be replaced. A critical issue concerning restoration is really the handling of documents and electronic media. Handling of the physical damage is more easily accomplished and more straightforward. The handling of vital records, however, is more difficult. The vital records may only be needed if an original contract is challenged, or is needed from a corporate entity standpoint. How a company deals with this exposure is not an easy determination. Some companies build facilities that are protected from most hazards to critical documents and data. The issue concerning having both a protected environment and duplication becomes a business issue; how much insurance is enough? Therefore, any time a company only has a single copy of vital documents and data, it must develop a strategy of what it would do if those records are damaged. This is a dilemma for many companies where duplicate copies cannot be maintained. Insurance companies have millions of pages of archived contracts and other legal documents that may not be feasibly copied. Other industries such as financial services handle equity certificates and other legal tender that perhaps cannot be copied as a normal course of business.

A company should develop a restoration plan in conjunction with performing a vital records review. In this way, the restoration of business-critical items can be assessed along with the alternatives of providing replication. Insurance coverage must be evaluated and coordinated with the restoration plan and other components of business continuity planning.

How to Select Restoration Service Providers

It is not difficult to find a service provider to clean up the rubble following a flood or fire. It is much more difficult to find a service provider that knows how to dry the soaked documents to best ensure their usability. It also takes a lot of expertise to handle fire-damaged documents and magnetic media to restore information.

The normal care for selecting any critical supply chain partner should be used. For a restoration company, however, you don't have the ability to ask for a pilot program. There are many sources of information to identify restoration companies, including local, state, and federal agencies. In addition, the Internet is an excellent source for both planning information, and resources.

Your own insurance carrier is also a good source of service provider information. Additionally, many insurance carriers have a partnership with recovery firms so that a firm is authorized to do certain work and

deal directly with the insurance carrier to ensure there are no misunderstandings about the work to be performed.

Where Does Insurance Coverage Fit into Your Restoration Program?

The subjects of restoration and insurance are closely intertwined as, in most cases, property insurers are expected to pay for the majority of the cost of any restoration. The settlement of a property insurance claim can be a complex, time-consuming, and vexing issue, even for a seasoned insurance professional. The insured often do not understand their coverage and routinely overestimate the amount of the loss or assume that a claim is covered when it is not. Insurers and their representatives may communicate poorly with the insured as to the nature of the coverage, the information required to adjust the claim, and the timetable to be expected. Both sides need to cooperate and communicate clearly so that reasonable expectations are established quickly and conflicts can be resolved in a timely manner.

The discussion on insurance includes a brief overview of standard commercial property insurance policies and common problems during the claim settlements process.

Property Insurance Overview

Property insurance can be purchased with many options, which serve to tailor the standard policy language to the specific needs of the policyholder. Therefore, it is important that business owners take the time to review their needs with their insurance agent, broker, or advisor, so that the resulting insurance purchase reflects those needs before a loss occurs. This will help avoid future misunderstandings with the insurance company in the event of a claim.

Property insurance can be purchased on either a Named Perils or All Risk form. The All Risk form covers all causes of loss that are not specifically excluded in the policy and provides broader protection to the insured than a Named Perils form. Under a Named Perils form, the insured bears the responsibility of proving that damage to the property was caused by one of the enumerated causes of loss. Use of the All Risk form shifts the burden of proof onto the insurer to prove that a particular loss was not covered by the policy. Insurers avoid the use of the phrase "All Risk" and use the phrase "Special Form" to describe this same coverage.

The property policy valuation clause is a second area of frequent misunderstanding by policyholders. That is, if a loss occurs, on what basis will the policyholder be compensated for the loss or damage to the property? Insurers offer two basic valuation choices: actual cash value (ACV) or replacement cost coverage. ACV is defined as the cost to repair or replace the lost or damaged property with property of like kind and quality less physical depreciation. For example, suppose that a commercial refrigerator purchased five years ago and expected to have a useful working life of ten years is burned up in a fire. Assuming that the refrigerator had been well maintained up to the time of the loss, the insurance company adjuster might offer to settle the claim for 50 percent of the cost today of a new refrigerator of similar design, quality, and capacity. It should be noted that the lost or damaged property will be valued as of the date of the loss and not on the basis of the original cost.

Replacement cost valuation means that the policyholder will be compensated on the basis of new for old. That is, the policyholder is entitled to compensation on the basis of the cost to repair or replace the lost or damaged property with property of like kind and quality with no deduction for physical depreciation. As noted above, the determination of the replacement cost of the damaged or lost property takes place as of the actual date of loss.

Regardless of whether ACV or replacement cost valuation is chosen, the policyholder needs to make sure that the amount of insurance purchased accurately reflects the current replacement cost value of the insured property. This is necessary to avoid a coinsurance penalty being applied that could reduce any loss adjustment.

If replacement cost coverage is chosen, then in the event of loss or damage to the covered property, the insured must actually repair or replace the lost or damaged property. Otherwise, the insurance company is usually only required to reimburse the insured on an ACV basis.

Finally, the insurance company will never pay more than the applicable amount of insurance that has been purchased by the policyholder. This last provision underscores the need for business owners to adequately assess the replacement cost value of their property at the time the policy is placed.

We have not included an in-depth discussion of the topics of Business Interruption or Extra Expense insurance in our discussion of property insurance because it is beyond the scope of this chapter. These coverages

go hand in hand with adequate property insurance coverage. Business interruption coverage pays for lost earnings and continuing expenses during the period of time the business is shut down. Extra expense coverage pays for the additional costs to maintain business during the shut-down period. The absence or insufficiency of either of these coverages can jeopardize the survival of the business that is jeopardized because of a lack of financial resources during the restoration period. Detailed records of all expenditures to maintain the operations of the business (extra expense) should be kept and included in the claim. The business interruption portion of the claim will be based on the lost earnings of the business as compared with periods preceding the loss.

In addition to standard property insurance coverage, business owners should discuss with their insurance advisors the need for additional insurance coverage in the following areas:

- Boiler and machinery
- Valuable papers
- Accounts receivable
- Electronic data processing (EDP)

Property insurance policies exclude coverage for damage caused by:

- Explosion of steam boilers, steam pipes, steam engines, or steam turbines
- Artificially generated electric current, including electric arcing, that affects electrical devices, appliances, or wire
- Mechanical breakdown, including rupture or bursting caused by centrifugal force

Such damage may be covered under boiler and machinery insurance policies. Boiler and machinery policies have many characteristics similar to property policies. In the event of a loss, these insurers often provide assistance in the repair or replacement of the damaged equipment. They also provide statutorily required inspection services.

Valuable papers coverage under a standard commercial property insurance policy is limited to $2500. Valuable papers coverage may be important for businesses where the destruction of documents would cause the business to suffer a monetary loss or to expend large sums in reconstructing the documents. The limit of insurance under a standard property policy can be increased to meet a desired need. The ISO (Insurance Services Office) valuable papers form defines valuable papers and records as “inscribed, printed, or written documents, manuscripts, or records.” Money and securities, data processing programs, media, and converted data are not covered. Coverage for loss or destruction to money and securities can be found in Crime Insurance policies. Data processing programs, media, and data can be covered under EDP policies. Care needs to be exercised in estimating the cost of reconstructing documents so that adequate limits of insurance can be purchased.

If Accounts Receivable records are damaged by an insured cause of loss, this type of coverage will pay the business owner amounts due from customers that he is unable to collect as a result of the damage to his records, collection expenses in excess of normal collection costs, and other reasonable expenses incurred to reestablish records of accounts receivable. This coverage can be purchased as an endorsement to a commercial property insurance policy. Again, care must be exercised in setting an adequate amount of insurance.

Electronic data processing (EDP) coverage is a must for organizations that rely heavily on data processing or electronic means of information storage. EDP coverage can provide All Risk coverage for equipment and data, software and media, including the perils of electrical and magnetic injury, mechanical breakdown, and temperature and humidity changes, which are important to computer operations. In addition, the coverage can include the cost of reproducing lost data, which is not available under a standard commercial property insurance policy.

Property Insurance Claims Settlement Process

Exhibit 137.1 provides a broad overview of the claim settlement process. The exhibit underscores the importance of complete and well-organized documentation and open communication during the claim settlement process. These two factors are major reasons why claims settlements are delayed or even end up in litigation. The items shown in this table are important steps to include in your restoration plan.

The claims settlement process is adversarial by its nature. The insured party is intent on maximizing its potential recovery under its insurance policy, while the insurance company is trying to minimize its exposure

EXHIBIT 137.1 Overview of the Claim Settlement Process

- Report the event to the property insurance company immediately. Depending on the specific items damaged and the nature of the damage, it may be appropriate to notify the boiler and machinery insurer as well.
- Prevent further damage to covered property.
- Obtain property repair/replacement estimates or appraisals and prepare and document the claim. If business interruption and/or extra expense are going to be claimed, extensive additional documentation may be needed. (If a business interruption loss exceeds $1 million, the insured should consider hiring accountants experienced in documenting such claims.)
- Submit documentation to the insurance company adjuster and cooperate with the adjuster in his investigation and adjustment of the claim.
- Request authorization to proceed with repairs or the purchase of major items.
- If appropriate, request a partial payment of the claim from the insurance company.
- Negotiate the final claim settlement with the insurance company adjuster.
- Submit a sworn proof of loss to the insurance company.
- Receive claim settlement.

to the insured's claim. This does not mean that the claim settlement process must be nasty or unpleasant. The parties should work together in good faith in arriving at a reasonable settlement of a claim. The insurance carrier will be less likely to raise substantive issues if it believes that the insured is not trying to take advantage of the situation. Likewise, if the insurer establishes reasonable ground rules at the beginning of the process, it should expect the insured to be forthcoming with the information requested in a timely manner. Although it is usually in the insured's best interests to provide complete and well-organized documentation, the insured should not overwhelm the insurance company and should only provide the documentation necessary to substantiate the amounts requested, keeping ancillary documentation available in the event that the insurance carrier requests additional information.

The insurance adjuster is an individual assigned by the insurance company to handle a claim on its behalf. The adjuster may be an employee of the insurance company or may work for an independent firm hired by the insurance company. Adjusters will be the key contact between the insurer and the insured. Their responsibilities include determining the cause of a loss, the nature and scope of damage to the property, whether the policy covers the damages claimed, to what extent property should be repaired or replaced and the corresponding cost, and finally the amount that the insurance carrier is willing to pay in settlement of the claim. The adjuster also acts as a quarterback in determining whether other specialists need to become involved.

Depending on the size and complexity of the claim, the insurance carrier may selectively involve accountants, lawyers, and other specialists in the claim settlement process. These specialists are working on behalf of the insurance carrier and not the insured. Although the insured should not be unduly alarmed if the insurance company employs such specialists, the insured may be well advised to consider employing his own specialists to work on his behalf in calculating the claim in order to be on a more equal footing with the insurance company.

The agent or broker who placed the insurance can provide guidance and assistance to the insured in handling the claim. This should be expected, because the broker or agent has received compensation to arrange the insurance. Smaller brokers sometimes lack the capability to be of much assistance in a claim situation.

The responsibilities of the policyholder in the event of a loss are spelled out in most insurance policies. They include prompt notification of the insurer, protecting the covered property from further damage, providing detailed inventories of the damaged and undamaged property, allowing the insurance company to inspect the damaged property, take samples, and examine the pertinent records of the company, providing a sworn proof of loss, cooperating with the insurer in the investigation and settlement of the claim, and submitting to examination under oath concerning any matter relating to the insurance or the claim.

Willis Corroon, a large multinational insurance broker, recommends that the following steps be taken immediately following a loss:

- Make sure that the loss area is safe to enter.
- Report the claim to the agent and to the insurer.
- Restore fire protection.
- Take immediate action to minimize the loss.
- Protect undamaged property from loss.

- Take photographs of the damage.
- Identify temporary measures needed to resume operations and maintain safety and security, and the costs of those measures.
- Consult with engineering, operations, and maintenance personnel as well as outside contractors for an initial estimate of the scope and cost of repairs.
- Make plans for repairing the damage.

What Is Included in a Restoration Plan?

After a disaster such as a fire or hurricane, the natural inclination is to assume that documents, computer records, equipment and machinery, and high-tech computers and other data processing equipment that appear to be unusable or severely damaged should be scrapped and replaced. However, before anything is done, experts should be brought in to assess the damage and determine short- and long-term courses of action. The short-term course of action is intended to stabilize the situation at the disaster location so as to prevent further damage from occurring. The long-term strategy is to determine which items can be salvaged and repaired and which should be replaced.

Although notification to the insurance company should be one of the first steps taken after a disaster has occurred, do not wait for the insurance adjuster to show up before implementing stabilization procedures. It is a common insurance policy requirement that the insured take steps to prevent additional damage from occurring after a disaster. Such post-loss disaster mitigation should be part of a comprehensive business continuity plan. If no plan exists, then common sense should prevail.

Your restoration plan should include the following:

- Ensure life safety at the disaster location.
- Reactivate fire protection and other alarm/life safety systems.
- Establish security at the site to keep out intruders, members of the public, the press, as well as employees who should not be allowed in the disaster area unless they are directly involved in damage assessment or mitigation efforts.
- Cover damaged roofs, doors, windows, and other parts of the structure.
- Arrange for emergency heat, dehumidification, or water extraction.
- Separate damaged components that may interfere with restoration, but do not dispose of these components because restoration experts and the insurance adjuster will want to inspect them.
- Take photographs or videotape of the disaster site as well as damaged and undamaged property.
- Bring in experts in document/records restoration and qualified technical personnel to work on computer and communications equipment and systems, machinery and furniture, wall and floor coverings, and structural elements.
- Maintain a log of all steps taken after a disaster, noting time, location, what has been done, who did it, as well as work orders and invoices of all expenditures relating to the disaster.

After the disaster site has been secured and stabilized and the extent of damage assessed, contracts should be negotiated with qualified restoration contractors. The insurance company adjuster may be able to recommend qualified contractors. The adjuster should be consulted before any contracts are awarded.

The extent of the restoration possibly depends on the type of property damaged, the nature of the damage, and the extent and speed of post-disaster damage minimization. Another factor is the level of expertise brought in to assess and recommend restoration strategies as well as the quality of the restoration contractors brought in to do the work.

Following are some generalized comments on the restoration of paper documents, magnetic media (computer disks and tape), and electronic equipment and machinery.

Water damage is one of the most prevalent forms of damage to paper-based documents. Restoration efforts need to begin immediately if documents are to be saved. Water should be pumped out of the area as quickly as possible. The area also needs to be vented to allow air to circulate. Cool temperatures will help preserve water-soaked documents until actual restoration work can begin. Bringing in a freezer unit such as a refrigerated trailer (capable of being held at 0 degrees F) to store the documents will help slow down mold damage. Before

freezing, documents should be cleaned and handled with extreme care. Documents should be kept in blocks (i.e., not pulled apart) as this will prevent additional deterioration. Documents that are not thoroughly soaked can be dried using dehumidification. Freeze-drying water-soaked documents will produce good results. Sterilization and application of a fungicidal buffer will help prevent further mold damage. Dehumidification and freeze-drying can take from one to two weeks to be completed.

Damaged computer tapes and diskettes need to be restored within 72 to 96 hours of a disaster to be effective. Water-damaged diskettes can be opened and dried using isopropyl alcohol and put into new jackets. Then the information is transferred onto new disks. Tapes can be freeze-dried or machine-dried using specialized machinery. The data on the tapes is then transferred to new media. Soot- and smoked-damaged diskettes need to be cleaned by hand, and then data transfer can take place.

Equipment and machines need to be evaluated on a case-by-case basis. There are specialist firms that can evaluate and recommend repair/restoration strategies for equipment. These firms may also do the repairs, or they may recommend shipping the damaged equipment to the manufacturer or utilize other shops to do the restoration. In general, insurance companies will not authorize replacement of damaged equipment with new or refurbished equipment unless the cost to repair the item exceeds 50 percent of its replacement cost. Smoke, soot, and other contaminants can be removed from equipment and replacement parts when damaged parts cannot be adequately cleaned. Occasionally, the original manufacturer may balk at substantially repairing damaged equipment, claiming that the repair will prove inadequate or will void the manufacturer's warranty. They are usually interested in selling new equipment. In such cases, insurance companies may be able to purchase replacement warranties (to replace the original manufacturer's warranty) from a warranty replacement company to satisfy the insured. The replacement warranty will be for the period of time remaining on the original manufacturer's warranty.

What Are the Costs for a Restoration Program?

The costs associated with restoration are more "at time of disaster" costs and would be covered by insurance. Having a thorough restoration strategy and plan will help to scope the insurance needed, and may even save money for those who are over-insured due to the lack of knowledge.

The primary cost of a program are the people resources necessary to develop and maintain the capability.

An approach to matching insurance needs with the potential cost to restore data and infrastructure is to start with your insurance carrier. Determine the types of restoration covered with different policies and then compare the coverage with restoration company estimates. Costs are usually based on square feet, type of media, etc.

Restoration of critical equipment is usually procured through the source of the equipment. This may include staged replacement parts or quick-ship components. Sometimes there is an incremental charge to maintenance fees to guarantee expedited service or replacement.

Ensuring Provider Can and Will Perform at Time of Disaster

Restoration is a service not dissimilar to maintenance for critical IT and facility operations. In the event of an emergency, any delay can cause a significant financial impact. You should view restoration in this same light. Therefore, expend the same diligence you would to selecting a service provider for ensuring business continuation, to selecting one for ensuring timely business resumption.

Testing Your Restoration Plan

Once a restoration plan has been implemented, it should be tested as part of a company's BCP program. The purpose of testing will be to validate that the plan:

1. Meets the business needs in terms of timeframe
2. Reduces the exposure to the loss of documents and data to an acceptable level
3. Remains in compliance with insurance requirements
4. Is current and the level of detail is sufficient to ensure a timely, efficient recovery

Testing is a primary means of keeping the restoration plan current. Regular tests with varying scope and objectives prevent the program from becoming too routine. As with any testing program, you start out simple and build on successes. Initially, it may involve contacting your service providers and verifying the following:

- You would be able to reach them at any hour, on any day
- They should be able to respond within the expected timeframes

Other tests may involve your restoration team members' awareness of the plan, ability to perform the tasks, and coordination with other "recovery and return to normal" activities.

In some cases, a company's need for restoration services actually diminishes. As IT solutions become more robust and the need for nonstop processing increases, more and more companies employ remote, replicated data. In this case, if the primary copy of data is lost, a second, equally current copy is available. Therefore, if a company had services for the restorations of electronic media, it may not be necessary.

Restoration Plan without a BCP Plan

Even if your company does not have a BCP program, it is still prudent to have ready resources to provide restoration services if needed. A company that does not understand the need for a BCP program will not allocate resources to develop a restoration strategy. A fallback would be to coordinate with your insurance carrier an understanding the critical nature of your vital records and single points of processing failure in order to procure the appropriate resources to get the job done.

Conclusion

A restoration strategy is one that can be implemented relatively easily and at minimal cost. Have your insurance carrier explain the types of hazards and restoration techniques, and if in a bind, work with the approved service partners.

Because time is of the essence when it comes to recovering damaged vital records and sensitive equipment, a BCP team should be assigned specific restoration responsibilities. Restoration should be a close second when it comes to recovering your business following a disaster.

Getting Support for Your Restoration Program

The most difficult task in developing a restoration capability and plan is to get internal manpower resources approved to help with the work. There may be some reluctance to go to management and suggest there is a need to prepare for the potential damage to critical property after management has spent money supposedly to eliminate the risk.

Everyone has seen news reports of damage due to floods, fires, and explosions. What most people do not know is that there is significant technology available to recover the critical data from damaged vital records. In addition, there are service providers who will guarantee replacement equipment within preestablished timeframes for a fixed subscription fee.

The important task is for the owner of critical business data and processing equipment to educate himself and his management that preplanning can significantly reduce the impact from potential loss of data.

Next Steps to Planning for Restoration

Below is an outline of steps to be performed to design and implement a restoration strategy to further protect a company's informational and physical assets.

I. Assess the needs
   A. What insurance coverage currently exists for the recovery and restoration of vital records following an event?
   B. What are the coverage options available for restoration of archival data and documents, as well as data needed to fully recover business processing?
   C. What are the business risks in terms of single copies of vital records?

D. What are the business risks associated with the loss of equipment and facilities?

II. Develop a restoration strategy
   A. Identify alternatives to either eliminate single points of failure or reduce the impact of lost or damaged property.
   B. Perform a cost/benefit analysis of viable alternatives.
   C. Obtain approval and funding for appropriate alternatives.
   D. Implement the preventive and restoration strategies.

III. Develop a restoration plan and ongoing quality assurance
   A. Incorporate restoration into the existing BCP program.
   B. Assign restoration roles and responsibilities.
   C. Coordinate restoration with the risk management department and other BCP efforts.
   D. Develop ongoing plan maintenance tasks and schedules.
   E. Perform periodic tests of restoration capability.

138

Business Resumption Planning and Disaster Recovery: A Case History

Kevin Henry, CISA, CISSP

Business resumption and disaster recovery planning is probably the part of information security that is easiest to overlook and postpone. Perhaps that is because few people actually enjoy preparing a business resumption plan. Like insurance, it is something one hopes is never needed; and because it is an inexact science at best, one is rarely sure that it has been completed correctly. More often, however, no one intentionally delays business resumption planning; it just does not happen — because of other job pressures, deadlines, and more seemingly urgent demands on one's time.

It is estimated that fewer than 50 percent of all firms have a reliable, complete, and current business resumption and disaster recovery plan in place.[1] For that reason, many firms are looking at two initiatives to address the lack of viable business resumption plans. The first is establishing a risk manager position within the corporation, a position with the primary responsibility of coordinating the development of business resumption and disaster recovery plans. The second initiative is to build business resumption and disaster recovery plan funding and timelines into every project. This is intended to force the development of plans prior to the project wrapping up and the team members dispersing. The effectiveness of these initiatives will ultimately depend on the leadership of senior management to enforce the mandate of the risk managers and require the completion of these tasks prior to project closure.

Because no organization ever wants to experience either a partial or full interruption of business operations, there is a silver lining in every cloud. The experience of having handled — and survived — a disaster can have a long-term benefit to a company. This chapter examines an actual case history of a computer system failure and the events that contributed to this becoming a disaster. In this particular instance, the business plan was implemented and, as it always seems to be, it was not a complete solution; however, it allowed a measure of the business process to continue to operate.

A business resumption plan is designed to provide an alternate method of continuing business operations in the event that the "normal" processes have been disrupted. A business resumption plan must address all types of scenarios that could disrupt the business process. These can be computer failures, but they are often other internal or external incidents that prevent an operation from continuing its usual practices. Some of these other disruptions may be environmental, such as fire (even if in a nearby structure) or flood, or they may be other external issues such as labor disruptions, gas leaks, or power failures. One notable computer system failure was caused by a watermain break some distance from the data processing site. When the water supply to the air conditioning unit was stopped, the air conditioning unit shut down and the data center overheated within a very short time.

One primary purpose of a business resumption plan and disaster recovery plan is to reduce the likelihood of a disaster occurring. This is a natural by-product of the initial stages of a properly developed business

resumption plan. As the business resumption team begins to examine the area that it is developing a plan for, that team will create an awareness of the risks a system or corporation is exposed to. This will also locate and identify the weaknesses that could lead to an operational failure. These weaknesses might be found in a system, a process, hardware, software, lack of training, personnel issues that have not been addressed, or some form of environmental or external threat. Following that, the purpose of the plan is to set up a framework for the business process to be able to resume its usual operations in an alternate manner. The implementation speed of a business resumption plan is primarily dependent on the importance of the system. A critical system (such as 911, hospitals, or air traffic control) must have a plan that can be operational within seconds or minutes, while a less-critical system may be able to slowly come up to speed, over a period of days or even weeks.

An excellent example of a successful business resumption scenario was the ability of United Airlines to continue its operations despite a fire that shut down its operational control center for three weeks in 1999. Despite controlling 2500 flights a day from that site, United was able to resume processing at its backup site in less than one hour, with the result that only one flight had to be canceled and a handful of other flights experienced minor delays. Fortuitously, this backup site was just in the final stages of acceptance testing as part of the development of a new business resumption plan.

Once a disaster has struck, the primary intent of the business groups is to resume operations with as little operational impact on critical systems as possible. Simultaneously, the disaster recovery plan implementation is beginning. The first goal of the disaster recovery plan is to prevent further damage. This means, first and foremost, ensuring personal safety. Then the disaster recovery plan splits into three areas: cleanup of the damaged site (salvage and repair), supporting the alternate business operations, and transition back to normal process.

The ultimate goal of the business resumption and disaster recovery plan is achieved when business operations are able to resume their normal or predisaster state. Failure to be able to maintain or resume operations in a timely manner results in a devastating statistic of nearly 50 percent total business failure.

To be effective, a business resumption and disaster recovery plan must be fully documented. Every responsibility and task, all software and hardware, communications links, and security requirements must be written out and available immediately when required. It is not sufficient to rely on personnel with a wealth of experience or understanding of the operations to be available for consultation in the middle of the disaster. When properly documented, any two people reading the document will reach the same conclusions and take the same actions. When this can be proven to be the case, then one can be assured that the documentation is thorough and clear.

A Case History

This case history is an actual sequence of events experienced by Serv-co (a fictitious name). There is a tremendous amount of information to be learned from this disaster — both to see the sequence of events that led up to and contributed to the disaster itself, and the lessons learned through the handling of the disaster.

Serv-co had a payments processing system (see Exhibit 138.1) that handled all of the incoming payments to the company — mailed checks, Internet payments, and payments handled by agents of Serv-co, including local banks and independent agents and representatives. The payment processing system handled in excess of 25,000 payments daily. The incoming payments were handled at three separate workstations (see Exhibit 138.1). The workstation operators would enter the payment amount and account number into the workstation. Once a thousand payments had been entered, the file was closed and transmitted to a central server. Attached to the file were control totals to assist in verification of file integrity and error detection. Once a day, the area manager would log on to the server and group all of the day's transaction files into one large file. Once some preliminary balancing had been done, the manager would establish a communications link to the legacy mainframe system that handled all customer account management and invoicing. The manager would transmit the cumulative file to the legacy system. Once received by the mainframe system, batch processes would be run that posted all of the payment activity to the individual customer accounts.

Unfortunately, one day the payment processing system failed.

The failure of the payment processing system happened, as most failures seem to do, on a Friday afternoon in mid-summer when most people's minds are already at the beach. The area manager called the support vendor and reported a strange error code that had been encountered when she tried to transfer the day's payments summary file to the mainframe system.

Being late on a Friday, it was agreed that the support company, referred to as Maint Group, would come out to Serv-co's location first thing Monday morning to investigate and correct the problem. This was not

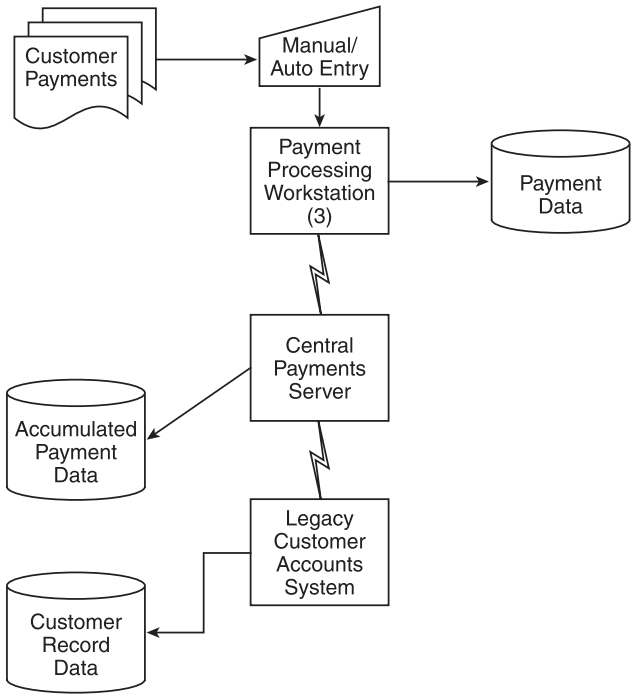


EXHIBIT 138.1 Payments system layout.

considered a serious problem. In the past, it had happened that minor system failures or file errors and imbalances would delay the posting of customer payments to their accounts by a day or two.

With his usually cheerful greeting, Maint Group's technician arrived early Monday morning to repair the problem. It should be noted that Maint Group was not the original vendor of the equipment; Maint Group had assumed the maintenance contract when the original vendor failed and went out of business. Within moments, the helpful grin of the technician faded as he realized that despite his years of experience with this equipment, he had never encountered this error condition. At this point, the value of a large vendor with a network of offices and a second-tier support group became apparent. Although this was not a common error, the technician was able to obtain assistance through contact with another branch (see Exhibit 138.2).

The error turned out to be a hard drive disk failure requiring the replacement of the hard drive. Here the first major deficiencies of the Serv-co payment processing system became clear. When Serv-co had purchased the system, instead of purchasing a server-class machine, the system server only included one hard drive and one power supply. Because of this, Serv-co's own Information Systems Standards Group had refused to accept the maintenance and oversight of the system.

EXHIBIT 138.2 Vendor Selection

When making a new purchase of hardware or software, or making the decision about in-house support or outsourced maintenance agreements, the choice of vendor is critical. Many times, the number of firms to choose from may be limited, especially when proprietary products are involved. However, where possible, a company or agency should ensure that it retains sufficient skills in-house to be able to perform or oversee basic updates and tasks. This is a safeguard against vendor failure or vendor labor disruption. Also, the choice of a large vendor may be more expensive than a local, smaller vendor. The larger vendor may have more technical and equipment support available for that price, whereas the smaller vendor may be able to provide faster or a more-personal level of service. Ensure that the choice of vendor includes a review of whether the vendor has adequate support systems in place to deal with abstract or custom problems and has ready access to spare components; although there may be a higher cost for such support, it can be critical in a disaster scenario.

Since the original purchase, the Payment Processing group had also moved to a new location. This meant a move of their workstations and server to a new facility. The equipment was located in a secure room; however, no provisions had been made for a proper power supply (UPS), nor were the proper environmental conditions provided for the equipment. This included mounting the server itself on a shelf over a desk. In addition, no secure and organized storage facility was provided for the backup tapes. The Payments Processing workgroup had several employees who had a keen interest in computers and were enthusiastic about looking after the equipment; however, without proper training and knowledge, they were unable to identify some of the basic deficiencies in the setup of the system.

As with many corporations, Serv-co had undergone some major restructuring a few years earlier. As part of this, several of the employees who were most knowledgeable about the system were released from Serv-co as part of a downsizing initiative. Because there was little or no documentation for the system, much of the practical knowledge of the system departed with these individuals (see Exhibit 138.3).

Back to the case history. The technician now had to find a new hard drive for the server. Because the equipment was now more than 12 years old, it was obsolete and piece parts were becoming increasingly difficult to find. In fact, Maint Group had sent a note to Serv-co two years earlier, indicating that the hard drive for this system was manufacturer discontinued and had exceeded its life expectancy. Maint Group recommended immediate replacement of the equipment. As a part of this notice, Maint Group also indicated that because of these limitations, it would only be able to continue to support the equipment on a "best effort" basis.

The technician was able to locate a hard drive in another city and arrangements were made to courier the hard drive to Serv-co for delivery first thing the next morning (Tuesday).

Tuesday morning the package arrived; the drive it contained was not the same one indicated by the label on the outside of the package. (Obviously, whenever a critical delivery of this type is required, the sender should take the necessary steps to verify the contents of the delivery.)

At this point, Serv-co had begun the transition from a minor inconvenience to a major disaster. Every day that passed caused an increase in the number of customers who have made payments to Serv-co and they received bills that did not reflect those payments. Moreover, these bills assessed the customers with an invalid late payment charge. This began to cause increased workload for the Customer Service Representatives and lead to poor customer relations and possibly even unwelcome media attention. By the end of this disaster, more than 15,000 customers had been affected.

Maint Group located two more hard drives in other parts of the country and arranged to have both sent to Serv-co for delivery the next morning. However, Wednesday morning arrived with no deliveries. Because of a labor disruption at the airline, the packages had been bumped off their flights and consequently did not arrive.

Thursday morning a replacement hard drive arrived and, with a great sense of relief, the technician began to install it. Once installed, the technician asked the local manager for the copies of the system backups so that he could begin to load the operating system onto the new drive. The manager reached across the shelf and passed the technician a stack of old tape cartridges. For several years since the downsizing of the "computer support" person for the group, the manager had faithfully been taking daily backups and storing them on these tapes. What she did not realize was that all she was backing up were the daily transaction files, not the operating system. Serv-co had no viable backup copy of its operating system.

The Maint Group technician called his technical support personnel and was told that a generic copy of the operating system was available, but that it would not contain any customization that had been built into the

EXHIBIT 138.3 Documentation

Documentation is perhaps the most critical resource in a disaster situation. When properly prepared, documentation allows all personnel involved to understand their tasks and responsibilities and how those tasks fit into the other activities surrounding the disaster. Ideally, documentation should be written in a clear, standard format so that no time or effort is lost trying to understand the flow of the documents. This means that any two people who read the documents will come to the same conclusion and undertake the same actions.

Documentation must be written for all processes and tasks surrounding a system, especially the routine or mundane daily tasks. Often, it is these tasks that no one knows how to do, or forgets, when the "expert" is sick or on vacation.

operating system by the original vendor (who, as one remembers, had since gone out of business). This generic copy was installed but it was not useable in its current state. Maint Group immediately began the task of writing patches to the operating system to meet the requirements of the Serv-co application. These patches were promised to be ready by the following Tuesday.

At this point, the customer impact had become critical and Serv-co began to examine its business continuity program. As a proper program should, it reflected the critical time factors that applied to this group. Management had accepted that payments processing was not as critical as some other services provided by Serv-co and rightfully had designed the plan to allow for a few days' delay before business process resumption. The business resumption plan prescribed a manual work-around of entering the payments into financial spreadsheets. These spreadsheets would then be FTP'd to the legacy mainframe systems and the batch processes adapted to read the new files. This was a tremendously labor-intensive operation, and a call went out to the various departments within Serv-co to provide personnel to work over the long holiday weekend to input these payments.

Because of the manual effort involved, more personnel were also required to examine the completed spreadsheets to detect errors. In fact, of the many spreadsheets created, only one was found to be totally error-free. The local Payments Processing manager called the Risk Management group to alert them of the implementation of their business continuity plan and was advised to "keep them posted." This was a breakdown in the role of the Risk Management group. With their knowledge of crisis management and process flow and their familiarity with contacting other groups such as Human Resources, Legal, and Corporate Communications, they could have provided a substantial level of assistance in handling this disaster. But like so many departments, Risk Management was short staffed due to vacations. Without this assistance and coordination, the local manager in Payments Processing was soon overwhelmed with calls from other groups for scheduling and recovery operations. The demands of this activity on the manager's time and the time of the other people in her group further impacted their ability to respond to the business needs. The other result of the lack of input from Risk Management was that proper communication with the unions on the property were not established and, instead of receiving support for their recovery efforts, the manager was soon faced with several grievances pertaining to people from the wrong jurisdiction doing another bargaining unit's work. This may not have been avoidable, depending on the overall tone of labor/management relations, but proper communication and involvement may have prevented further animosity and stress in an already tense situation.

On Tuesday morning, the Maint Group technician arrived with the patches for the operating system. Once installed, these patches provided some functionality but many of the error-detection and balancing controls were absent. Also, the server was unable to establish a communications link with the mainframe. The last time this link had been set up, it had taken two technicians three days to determine the correct settings. Once again, the documentation was missing, and with it, this critical piece of information. Fortunately, a copy of the configuration was found in the recycling bin by a LAN support person who had been doing an inventory of communications links several months earlier.

Over the next week, Serv-co was able to catch up on its payments processing, but the cost in manpower and goodwill was extensive.

It is noteworthy that at the time of this failure, Serv-co had already bought a replacement system but it had not yet been delivered by the vendor. This process had started more than two years earlier with the notification of the obsolete equipment, but it had encountered several hurdles along the way. Management had twice sent the purchase proposal back to the Payment Processing department to explore other options (such as outsourcing) and less-expensive solutions. This delayed the replacement long enough for the existing equipment to finally fail.

Once again, however, the Payments Processing area had purchased the replacement equipment without the input and oversight of the Information Systems Standards group. As a result, the new equipment was similar to the old equipment in that it only had a single hard drive and a single power supply. It was also designed as a stand-alone system and plans had not been made to back it up to the corporate enterprise storage system. In fact, the Information Systems Standards group had once again declared that it would not support the new system, and its only concern with the project was that the interface to its legacy systems would work correctly.

So, what did Serv-co learn from this disaster? And what can the reader learn? A lot.

Professional Support

Ensure that all systems are installed with the oversight of information systems (IS) professionals and according to corporate standards. The active involvement of the IS staff in the procurement and support of stand-alone systems will prevent many minor errors from turning into major disasters. If the corporation does not have the standards it needs to develop them, this will also prevent further holes from developing in the security infrastructure through incompatible equipment. The more standard the equipment is, the easier it is to have in-house knowledge and keep the correct operating system patches up to date. Standard equipment also allows for easier load sharing and minimizes single points of failure. As a part of this, all companies should ensure that they have knowledgeable support for all of their systems. Especially when a system has been developed by an outside contractor, ensure that the knowledge of the system is not lost at the completion of the project. Once this disaster was resolved, Serv-co's Payments Processing and IS departments began to cooperate and redesign the replacement system. This included a regular backup to the enterprise storage system and the purchase of server-class equipment.

Backups

It goes without saying that proper backups must be done on all operating systems. Often, it is configurations (communications, routers, etc.) and rule bases (firewalls) that are overlooked. In all cases, backups should be done often enough to ensure that a processing cycle can be rebuilt if necessary. There are many examples of situations in which a system has only kept two or three generations of certain files. In the event of a failure (especially when the failure was related to application program change), the on-call programmer tries to rerun the job. If the subsequent rerun fails, it could happen that the last good backup has already been aged off and deleted before the problem is corrected. It is also important to ensure that all legal requirements for backups are met, such as long-term retention of financial records.

There are many different types of backup media available these days, including various tape products and CDs. The latest documentation on CDs indicates that they have a life expectancy even in adverse conditions of up to 200 years. In that case, the lifespan of the product is not the problem; the challenge is to ensure that any encryption keys are securely stored and available, and the software needed to read the CDs is also available the day that the data is required.

When recording backups, always ensure that the backup copy is readable. One company recently attempted to recover from a disk-head write failure only to discover that four of its 20 newly purchased tape cartridges were faulty. When it comes to the point of needing to recover from a backup, if the backup is faulty, the extent of the problem grows exponentially.

Equipment Aging

More and more of the equipment in use in corporations and agencies these days has already exceeded its lifespan. This is especially true for hard drives, power supplies, and tapes. A regular inventory of all equipment should be taken and the equipment specifications reviewed to ensure that the equipment is still reliable.

Dependencies

Many systems and business processes are not even aware of the other systems that depend on them, and that they themselves depend on for processing. Detailed data flow diagrams showing all internal and external system dependencies should be drawn up so that if a system fails, it is immediately apparent who else has been affected. This is especially important for financial systems and areas subject to regulatory requirements where the absence of a file may not be noticed but could have significant impact on processing or legal penalties.

Encryption

If the system has any form of encryption, it is necessary to keep all keys in a secure place for retrieval. Often, once a system has been operating for some period of time, the keys are forgotten; and when the system experiences a failure, it can be extremely dangerous if the keys are unavailable. Whenever an employee is using

encryption for company documents or files, a copy of the keys should be retained in a secure, trusted location. It has happened that the loss of an employee through accident or termination has left a company unable to recover critical files. In one recent case, an employee who was about to be terminated for inappropriate behavior was able to hold the company "hostage" by refusing to disclose his keys and the administrative passwords to several key systems.

Vendor Failure

One of the most prevalent characteristics of the entire information processing field has to be vendor change. On a nearly daily basis, vendors are opening, closing, merging, or changing business direction. When this is accompanied by the rapid replacement of one technology with a newer product, this can have a significant impact on business resumption plans. Information systems professionals need to be continuously aware of the state of their vendor support network. A list of vendor phone numbers and contact lists must be kept together with the business resumption plan, and many plans should also include a commitment from vendors to supply new equipment on a priority basis in the event of a major failure.

Vendor-supplied software should be kept in escrow (held in trust by a third party) so that it is available if the vendor is unable to meet its maintenance or upgrade contractual conditions.

When purchasing new equipment, the risk is always whether it will continue to be manufactured and supported. More than one company has been unable to obtain a maintenance agreement for equipment that it had recently purchased because the vendor moved to a new line of business and abandoned a certain product line.

When selecting a vendor, the decision must be made whether to go with a possibly higher-priced vendor that has a large network of support and spare equipment availability, or with a smaller or local vendor and mitigate the risk through the purchase of spare parts or retaining greater in-house expertise.

BCP: Up to Date

To get a comprehensive and complete business resumption plan set up is difficult, but the effort does not stop there. A corporation, department, or agency still needs to identify the person responsible for the plan on an ongoing basis. Plans need to be reviewed at least once a year and after any major change in departmental structure. This responsibility should be built into the job description of the person who will maintain the business resumption plan and represent the department on the corporate Risk Management team. If the department does routine job reviews, the adherence to this responsibility should also be reviewed.

Union

Unions are a fact of life in many companies and agencies these days, and that places certain legal restrictions on the employees and managers. In most jurisdictions, it is illegal to negotiate a separate agreement with an individual who is represented by a union. Often in a crisis, a manager has attempted to negotiate a separate pay or compensation agreement directly with employees. This may seem practical but it can also be illegal and unenforceable. The business resumption plan must include a method of contacting a union representative for a unionized group that could be involved or affected by a business interruption. Hopefully, through prompt communication, the union can be available to assist in the recovery and personnel coordination activity, rather than add increased complexity to the disaster through labor disruption.

Whether or not there is a union on the property, the Human Resources department should be involved in the recovery efforts to ensure that any applicable labor codes or laws are being followed.

Risk Management Involvement

Many corporations and agencies now have a Risk Management group that has overall responsibility for coordinating the departmental plans, liason with external and internal groups, and leadership in a crisis. This group needs to have unrestricted access to the senior management of the company and must have the mandate to assist or lead in any business disruption. Without this mandate, Risk Management groups often have

difficulty obtaining the subject matter experts (SMEs) to assist in a crisis because a manager in another group has refused to release them from their regular duties.

The focus areas of this group in a crisis are communication, collaboration, control, and coordination (the 4 Cs). With a properly set up group, a corporation will avoid the "Alexander Haig syndrome" of competing groups unsure of who is in charge and delivering conflicting statements.

One of the members of this group, on an as-needed basis, should be a member of the Health and Safety group of the company. This is to ensure that proper attention is being paid to the health issues, both mental and physical, of individual workers in a crisis scenario.

In a disaster, the Risk Management group should also ensure that all advertising campaigns related to the company or the disaster are halted or amended, and that a separate individual or organization is monitoring the media and providing feedback on how the corporation's statement or message is being received in the community.

Two factors that can be missed in many Risk Management groups are housekeeping and security of the Emergency Operations Center (EOC) and the site of the failure during disaster recovery efforts. Limiting access to the EOC and keeping it clean and uncluttered will aid in the smooth operation of the center.

The EOC should have separate access lines for the families of employees that are involved in the recovery operation so that they can pass on messages or receive updates. The understanding and resolution of family issues are critical to the involved individuals being able to focus on the recovery efforts. In addition, the company should have a telephone line with an answering machine that provides regular updates to other employees not directly related to the crisis. This can also be used to relay worksite and reporting information to the employees.

A disaster recovery operation often includes the disbursement of funds that exceeds the normal limit of local managers. A chain of command that accelerates the approval process or grants an increased spending limit on a temporary basis should be developed. There also needs to be a payroll process or provision for advance funds to be released to the families and individuals affected by the crisis.

The Risk Management group should have a list of the major customers of a company so that calls can immediately be made to these firms indicating that the company is still operational and outlining revised contact methods. This may prevent the loss of contracts or eroded confidence by the client community.

Downsizing

Downsizing has had a devastating effect on information systems security. It has led to the amalgamation of many functions, thereby removing separation of duties, and it has led to many individuals assuming responsibility for many tasks for which they have not received adequate training or experience. This is where inadequate documentation can harm a corporation. Often, many of the little jobs that were being done and the reasons for those actions are lost once a person has been released. Support people are especially vulnerable to downsizing because the benefit and importance of their work is not realized.

Downsizing also impacts morale and loyalty to the corporation. It has been estimated that a downsizing initiative deprives a corporation of four week's worth of productivity. Increased attention to security risks and possible malicious behavior must be included in the activity of the information systems security professional at this time. Most estimates are that 10 percent of an employee base will take advantage of an opportunity to defraud a corporation at any time. During a period of downsizing, this will usually rise to approximately 30 percent.

Documentation

Although documention was previously discussed, it is timely to add one further comment. Following any failure or test of the business resumption plan or a disaster recovery effort, review all documentation promptly to record all improvements and amendments to the documentation. Ensure that only the latest version of documentation is available (this can be accomplished by numbering the documents).

Partial Processing: Who Gets Priority

During a disaster every department wants priority service. This is not the time to make these decisions or to try to juggle multiple tasks. An integral part of developing business resumption and disaster recovery plans is to determine which areas of the company get first attention. In many plans, the plan does not include enough hardware or processing power to recover all business processes. Ensure that the correct ones are the ones that are recovered. Once a plan has been developed, have all managers sign off on it so that they realize and accept who will get first priority in the event of an incident.

Multiple Disasters: Be Aware of Other Disasters that May Impact the Primary Recovery Site

A daily task of the Risk Management group, and all business resumption planners, has to be monitoring of ongoing events that could affect a corporation's business processes or disaster recovery plans. For example, a corporation should attempt never to be surprised by an event at a neighboring facility or an environmental hazard that affects its ability to operate. This includes an awareness of ongoing disasters that may be affecting its disaster plan. An example of this was experienced following the World Trade Center bombing. A few weeks later, another company that lost its data center due to a structural failure (heavy snow load on the roof) was unable to move into its contracted hot site as planned because it was already in use by companies displaced from the World Trade Center. If that company had been attentive to this, it could have realized that this would have an effect on its disaster recovery plans and taken measures to arrange for an alternate site if necessary prior to its own failure.

A disaster recovery plan may be needed for an extended period of time; recent ice storms, for example, have disrupted commercial power for some firms for several weeks. Despite the fact that they were able to initially resume business operations, they were unable to continue because they only planned for providing alternate power for a few days.

Summary

Information systems security professionals have become keys players in the whole field of business resumption planning and disaster recovery. This is a radical departure from the normal duties of most information systems security personnel. Rather than a strictly technical or systems understanding, it requires them to gain an understanding of the entire business process and how they can support and enable those processes in a disaster scenario. The knowledgeable and professional advice that information systems security professionals provide will also significantly enhance the ability of most organizations, corporations, and agencies to prepare for and react to any incidents that could impair their business processes or threaten their very survival in a competitive and fast-moving marketplace.

References

1. Quantum Corporation, Disaster Readiness of BCP Professionals, *Disaster Recovery Journal,* 13, 1.

139

Business Continuity Planning: A Collaborative Approach

Kevin Henry, CISA, CISSP

Business continuity planning (BCP) has received more attention and emphasis in the past year than it has probably had cumulatively during the past several decades. This is an opportune time for organizations to leverage this attention into adequate resourcing, proper preparation, and workable business continuity plans. Business continuity planning is not glamorous, not usually considered to be fun, and often a little mundane. It can have all the appeal of planning how to get home from the airport at the end of an all-too-short vacation.

This chapter examines some of the factors involved in setting up a credible, useful, and maintainable business continuity program. From executive support through good leadership, proper risk analysis and a structured methodology, business continuity planning depends on key personnel making business-oriented and wise decisions, involving user departments and supporting services.

Business continuity planning can be defined as preparing for any incident that could affect business operations. The objective of such planning is to maintain or resume business operations despite the possible disruption. BCP is a preincident activity, working closely with risk management to identify threats and risks and reducing the likelihood or impact of any of these risks occurring. Many such incidents develop into a crisis, and the focus of the effort turns to crisis management. It is at this time that the value of prior planning becomes apparent.

The format of this chapter is to outline the responsibilities of information systems security personnel and information systems auditors in the BCP process. A successful BCP program is one that will work when needed and is built on a process of involvement, input, review, testing, and maintenance. The challenge is that a BCP program is developed in times of relative calm and stability, and yet it needs to operate in times of extreme stress and uncertainty. As we look further into the role of leadership in this chapter, we will see the key role that the leader has in times of crisis and the importance of the leader's ability to handle the extreme stress and pressures of a crisis situation.

A significant role of the BCP program is to develop a trained and committed team to lead, manage, and direct the organization through the crisis.

Through this chapter we will examine the aspects of crisis development, risk management, information gathering, and plan preparation. We will not go into as much detail about the plan development framework because this is not normally a function of IT or security professionals, yet understanding the role and intent of the business continuity program coordinator will permit IT professionals to provide effective and valued assistance to the BCP team.

So what is the purpose of the BCP program? It is to be prepared to meet any potential disruption to a business process with an effective plan, the best decisions, and a minimization of interruption.

A BCP program is developed to prepare a company to recover from a crisis — an event that may have serious impact on the organization, up to threatening the survival of the organization itself. Therefore, BCP is a process that must be taken seriously, must be thorough, and must be designed to handle any form of crisis that may occur. Let us therefore look at the elements of a crisis so that our BCP program will address it properly.

The Crisis

A crisis does not happen in isolation. It is usually the combination of a number of events or risks that, although they may not be catastrophic in themselves, in combination they may have catastrophic results. It has sometimes been said that it takes three mistakes to kill you, and any interruption in this series of events may prevent the catastrophe from taking place. These events can be the result of preexisting conditions or weaknesses that, when combined with the correct timing and business environment, initiate the crisis. This can be called a "catalyst" or "crisis trigger."

Once the crisis has begun, it evolves and grows, often impacting other areas beyond its original scope and influence. This growth of the crisis is the most stressful period for the people and the organization. This is the commencement of the crisis management phase and the transition from a preparatory environment to a reactionary environment. Decisions must be made on incomplete information amid demands and pressure from management and outside groups such as the media and customers. An organization with an effective plan will be in the best position to survive the disaster and recover; however, many organizations find that their plan is not adequate and are forced to make numerous decisions and consider plans of action not previously contemplated. Unfortunately, most people find that Rudin's Law begins to take effect:

> When a crisis forces choosing among alternatives, most people will choose the worst possible one.
>
> **— Rudin's Law**

Let us take a closer look at each of these phases of a crisis and how we can ensure that our BCP program addresses each phase in an effective and timely manner.

Preexisting Conditions

In a sporting event, the opposition scores; when reviewing the video tapes later, the coach can clearly see the defensive breakdowns that led to the goal. A player out of position, a good "deke" by the opponent (used in hockey and soccer when an opposing player fools the goalie into believing that he is going in one direction and yet he actually goes in a different direction, thereby pulling the goaltender out of position and potentially setting up a good opportunity to score), a player too tired to keep pace — each contributing to the ability of the unwanted event to occur. Reviewing tapes is a good postevent procedure. A lot can be learned from previous incidents. Preparations can be made to prevent recurrence by improvements to the training of the players, reduction of weakness (maybe through replacing or trading players), and knowledge of the techniques of the opponents.

In business we are in a similar situation. All too often organizations have experienced a series of minor breakdowns. Perhaps they never became catastrophes or crises, and in many cases they may have been covered up or downplayed. These are the best learning events available for the organization. They need to be uncovered and examined. What led to the breakdown or near-catastrophe, what was the best response technique, who were the key players involved — who was a star, and who, unfortunately, did not measure up in times of crisis? These incidents uncover the preexisting conditions that may lead to a much more serious event in the future. Examining these events, documenting effective response techniques, listing affected areas, all provide input to a program that may reduce the preexisting conditions and thereby avert a catastrophe — or at least assist in the creation of a BCP that will be effective.

Other methods of detecting preexisting conditions are through tests and audits, interviewing the people on the floor, and measuring the culture of the organization. We often hear of penetration tests — what are they designed to do? Find a weakness before a hostile party does. What can an audit do? Find a lack of internal control or a process weakness before it is exploited. Why do we talk to the people on the floor? In many cases, simply reading the policy and procedure manuals does not give a true sense of the culture of the organization. One organization that recently received an award for its E-commerce site was immediately approached by several other organizations for a description of its procedure for developing the Web site. This was willingly provided — except that in conversation with the people involved, it was discovered that in actual fact the

process was never followed. It looked good on paper, and a lot of administrative time and effort had gone into laying out this program; but the award-winning site was not based on this program. It was found to be too cumbersome, theoretical, and, for all intents and purposes, useless. Often, merely reviewing the policy will never give the reader a sense of the true culture of the organization. For an effective crisis management program and therefore a solid, useable BCP program, it is important to know the true culture, process, and environment — not only the theoretical, documented version.

One telecommunications organization was considering designing its BCP for the customer service area based on the training program given to the customer service representatives. In fact, even during the training the instructors would repeatedly say, "This may not be the way things will be done back in your business unit, this is the ideal or theoretical way to do things; but you will need to learn the real way things are done when you get back to your group." Therefore, a BCP program that was designed according to the training manual would not be workable if needed in a crisis. The BCP needs to reflect the group for which it is designed. This also highlighted another risk or preexisting condition. The lack of standardization was a risk in that multiple BCP programs had to be developed for each business operation, and personnel from one group may not be able to quickly assume the work or personnel of another group that has been displaced by a crisis. Detecting this prior to a catastrophe may allow the organization to adjust its culture and reduce this threat through standardization and process streamlining.

One of the main ways to find preexisting conditions is through the risk analysis and management process. This is often done by other groups within and outside the organization as well — the insurance company, the risk management group, internal and external audit groups, security, and human resources. The BCP team needs to coordinate its efforts with each of these groups — a collaborative approach so that as much information is provided as possible to design and develop a solid, workable BCP program. The human resources group in particular is often looking at risks such as labor difficulties, executive succession, adequate policy, and loss of key personnel. These areas also need to be incorporated into a BCP program.

The IT group plays a key role in discovering preexisting conditions. Nearly every business process today relies on, and in many cases cannot operate without, some form of IT infrastructure. For most organizations this infrastructure has grown, evolved, and changed at a tremendous rate. Keeping an inventory of IT equipment and network layouts is nearly impossible. However, because the business units rely so heavily on this infrastructure, no BCP program can work without the assistance and planning of the IT group. From an IT perspective, there are many areas to be considered in detecting preexisting conditions: applications, operating systems, hardware, communications networks, remote access, printers, telecommunications systems, databases, Internet links, stand-alone or desktop-based systems, defense systems, components such as anti-virus tools, firewalls, and intrusion detection systems, and interfaces to other organizations such as suppliers and customers.

For each component, the IT group must examine whether there are single points of failure, documented lists of equipment including vendors, operating version, patches installed, users, configuration tables, backups, communications protocols and setups, software versions, and desktop configurations. When the IT group has detected possible weaknesses, it may be possible to alert management to this condition as a part of the BCP process in order to gain additional support for new resources, equipment, or support for standardization or centralized control.

The risk in many organizations is the fear of a "shoot the messenger" reaction from management when a potential threat has been brought to the attention of management. We all like to hear good news, and few managers really appreciate hearing about vulnerabilities and recommendations for increased expenditures in the few moments they have between budget meetings. For that reason, a unified approach using credible facts, proposals, solutions, and costs, presented by several departments and project teams, may assist the IT group in achieving greater standards of security and disaster preparedness. The unfortunate reality is that many of the most serious events that have occurred in the past few years could have been averted if organizations had fostered a culture of accurate reporting, honesty, and integrity instead of hiding behind inaccurate statistics or encouraging personnel to report what they thought management wanted to hear instead of the true state of the situation. This includes incidents that have led to loss of life or financial collapse of large organizations through city water contamination, misleading financial records, or quality-of-service reporting.

It is important to note the impact that terrorist activity has had on the BCP process. Risks that had never before been seriously considered now have to be contemplated in a BCP process. One of the weaknesses in some former plans involved reliance on in-office fireproof safes, air transit for key data and personnel, and proximity to high-risk targets. An organization not even directly impacted by the actual crisis may not be able to get access to its location because of crime-scene access limitations, clean-up activity, and infrastructure

breakdowns. Since the terrorist actions in New York, several firms have identified the area as a high-risk location and chosen to relocate to sites outside the core business area. One firm had recently completed construction of a new office complex close to the site of the terrorist activity and has subsequently chosen to sell the complex and relocate to another area.

On the other hand, there are several examples of BCP programs that worked properly during the September 11, 2001, crisis, including tragic incidents where key personnel were lost. A BCP program that is properly designed will operate effectively regardless of the reason for the loss of the facility, and all BCP programs should contemplate and prepare for such an event.

Crisis Triggers

The next step in a crisis situation is the catalyst that sets off the chain of events that leads to the crisis. The trigger may be anything from a minor incident to a major event such as a weather-related or natural disaster, a human error or malicious attack, or a fire or utility failure. In any event, the trigger is not the real problem. An organization that has properly considered the preconditions that may lead to a crisis will have taken all precautions to limit the amount of damage from the trigger and hopefully prevent the next phase of the crisis — the crisis expansion phase — from growing out of control. Far too often, in a *post mortem* analysis of a crisis, it is too easy to focus on the trigger for the event and look for ways to prevent the trigger from occurring — instead of focusing on the preconditions that led to the extended impact of the crisis.

When all attempts have been made to eliminate the weaknesses and vulnerabilities in the system, then attention can be given to preventing the triggers from occurring.

Crisis Management/Crisis Expansion

As the crisis begins to unfold, the organization transitions from a preparatory stage, where the focus is on preventing and preparing for a disaster, to a reactionary stage, where efforts are needed to contain the damage, recover business operations, limit corporate exposure to liability and loss, prevent fraud or looting, begin to assess the overall impact, and commence a recovery process toward the ultimate goal of resumption of normal operations. Often, the organization is faced with incomplete information, inadequate coordinating efforts, complications from outside agencies or organizations, queries and investigations by the media, unavailability of key personnel, interrupted communications, and personnel who may not be able to work together under pressure and uncertainty.

During a time of crisis, key personnel will rise to the occasion and produce the extra effort, clarity of focus and thought, and energy and attitude to lead other personnel and the organization through the incident. These people need to be noticed and marked for involvement in future incident preparation handling. Leadership is a skill, an art, and a talent. Henry Kissinger defines leadership as the ability to "take people from where they are to places where they have never been." Like any other talent, leadership is also a learned art. No one is born a perfect leader, just as no one is born the world's best golfer. Just as every professional athlete has worked hard and received coaching and guidance to perfect and refine his ability, so a leader needs training in leadership style, attention to human issues, and project planning and management.

One of the most commonly overlooked aspects of a BCP program is the human impact. Unlike hardware and software components that can be counted, purchased, and discarded, the employees, customers, and families impacted by the crisis must be considered. No employee is going to be able to provide unlimited support — there must be provisions for rest, nourishment, support, and security for the employees and their families.

The crisis may quickly expand to several departments, other organizations, the stock market, and community security. Through all of this the organization must rapidly recognize the growth of the disaster and be ready to respond appropriately.

The organization must be able to provide reassurance and factual information to the media, families, shareholders, customers, employees, and vendors. Part of this is accomplished through knowing how to disseminate information accurately, representing the organization with credible and knowledgeable representatives, and restricting the uncontrolled release of speculation and rumor. During any crisis, people are looking for answers, and they will often grasp and believe the most unbelievable and ridiculous rumors if there is no access to reliable sources of information. Working recovery programs have even been interrupted and halted by the spread of inaccurate information or rumors.

Leadership is the ability to remain effective despite a stressful situation; remain composed, reliable, able to accept criticism (much of it personally directed); handle multiple sources of information; multitask and delegate; provide careful analysis and recommendations; and inspire confidence. Not a simple or small task by any means.

In many cases the secret to a good BCP program is not the plan itself, but the understanding of the needs of the business and providing the leadership and coordination to make the plan a reality.

Some organizations have been dismayed to discover that the people who had worked diligently to prepare a BCP program, coordinating endless meetings and shuffling paperwork like a Las Vegas blackjack dealer, were totally unsuited to execute the very plans they had developed.

The leader of a disaster recovery team must be able to be both flexible and creative. No disaster or crisis will happen "by the book." The plan will always have some deficiencies or invalid assumptions. There may be excellent and creative responses and answers to the crisis that had not been considered; and, although this is not the time to rewrite the plan, accepting and embracing new solutions may well save the organization considerable expense, downtime, and embarrassment. One approach may be the use of wireless technology to get a LAN up and running in a minimal amount of time without reliance on traditional cable. Another example is the use of microwave to link to another site without the delay of waiting for establishment of a new T1 line. These are only suggestions, and they have limitations — especially in regard to security — but they may also provide new and rapid answers to a crisis. This is often a time to consider a new technological approach to the crisis — use of Voice-over-IP to replace a telecommunications switch that has been lost, or use of remote access via the Internet so employees can operate from home until new facilities are operational.

Business resumption or business continuity planning can be described as the ability to continue business operations while in the process of recovering from a disaster.

The ability to see the whole picture and understand hidden relationships among processes, organizations, and work are critical to stopping the expansion of the crisis and disaster. Determining how to respond is a skill. The leaders in the crisis must know who to call and alert, on whom to rely, and when to initiate alternate processing programs and recovery procedures. They need to accurately assess the extent of the damage and expansion rate of the crisis. They need to react swiftly and decisively without overreacting and yet need to ensure that all affected areas have been alerted.

The disaster recovery team must be able to assure the employees, customers, management team, and shareholders that, despite the confusion, uncertainty, and risks associated with a disaster, the organization is competently responding to, managing, and recovering from the failure.

Crisis Resolution

The final phase of a crisis is when the issue is resolved and the organization has recovered from the incident. This is not the same as when normal operations have recommenced. It may be weeks or years that the impact is felt financially or emotionally. The loss of credibility or trust may take months to rebuild. The recovery of lost customers may be nearly impossible; and when data is lost, it may well be that no amount of money or effort will recover the lost information. Some corporations have found that an interruption in processing for several days may be nearly impossible to recover because there is not enough processing time or capacity to catch up.

The crisis resolution phase is a critical period in the organization. It pays to reflect on what went well, what lessons were learned, who were the key personnel, and which processes and assumptions were found to be missed or contrarily invalid. One organization, having gone through an extended labor disruption, found that many job functions were no longer needed or terribly inefficient. This was a valuable learning experience for the organization. First, many unnecessary functions and efforts could be eliminated; but second, why was the management unable to identify these unnecessary functions earlier? It indicated a poor management structure and job monitoring.

The Business Continuity Process

Now that we have examined the scenarios where we require a workable business continuity plan, we can begin to explore how to build a workable program. It is good to have the end result in mind when building the

program. We need to build with the thought to respond to actual incidents — not only to develop a plan from a theoretical approach.

A business continuity plan must consider all areas of the organization. Therefore, all areas of the organization must be involved in developing the plan. Some areas may require a very elementary plan — others require a highly detailed and precise plan with strict timelines and measurable objectives. For this reason, many BCP programs available today are ineffective. They take a standard one-size-fits-all approach to constructing a program. This leads to frustration in areas that are overplanned and ineffectiveness in areas that are not taken seriously enough.

There are several excellent Web sites and organizations that can assist a corporation in BCP training, designing an effective BCP, and certification of BCP project leaders. Several sites also offer regular trade journals that are full of valuable information, examples of BCP implementations, and disaster recovery situations. Some of these include:

- *Disaster Recovery Journal,* www.drj.com
- Disaster Recovery Institute Canada, www.dri.ca
- Disaster Recovery Information Exchange, www.drie.org
- American Society for Industrial Security, www.asisonline.org
- Disaster Recovery Institute International, www.dr.org
- Business Continuity Institute, www.thebci.org
- International Association of Emergency Managers, www.nccem.org
- Survive — The Business Continuity Group, www.survive.com

There are also numerous sites and organizations offering tools, checklists, and software to assist in establishing or upgrading a BCP program.

Regardless of the Web site accessed by a BCP team member, the underlying process in establishing a BCP program is relatively the same.

- Risk and business impact analysis
- Plan development
- Plan testing
- Maintenance

The Disaster Recovery Institute recommends an excellent ten-step methodology for preparing a BCP program. The *Disaster Recovery Journal* Web site presents a seven-step model based on the DRI model, and also lists the articles published in its newsletters that provide education and examples of each step. Regardless of the type of methodology an organization chooses to use, the core concepts remain the same. Sample core steps are:

- Project initiation (setting the groundwork)
- Business impact analysis (project requirements definition)
- Design and development (exploring alternatives and putting the pieces together)
- Implementation (producing a workable result)
- Testing (proving that it is a feasible plan and finding weaknesses)
- Maintenance and update (preserving the value of the investment)
- Execution (where the rubber meets the road — a disaster strikes)

As previously stated, the intent of this chapter is not to provide in-depth training in establishing a BCP program. Rather, it is to present the overall objectives of the BCP initiative so that, as information systems security personnel or auditors, we can provide assistance and understand our role in creating a workable and effective business continuity plan.

Let us look at the high-level objectives of each step in a BCP program methodology.

Project Initiation

Without clearly defined objectives, goals, and timelines, most projects flounder, receive reduced funding, are appraised skeptically by management, and never come to completion or delivery of a sound product. This is

especially true in an administrative project like a BCP program. Although the awareness has been raised about BCP due to recent events, this attention will only last as long as other financial pressures do not erode the confidence that management has in realizing worthwhile results from the project.

A BCP project needs clearly defined mandates and deliverables. Does it include the entire corporation or only a few of the more critical areas to start with? Is the funding provided at a centrally based corporate level or departmentally? When should the plans be provided? Does the project have the support of senior management to the extent that time, resources, and cooperation will be provided on request as needed by the BCP project team?

Without the support of the local business units, the project will suffer from lack of good foundational understanding of business operations. Therefore, as discussed earlier, it is doubtful that the resulting plan will accurately reflect the business needs of the business units.

Without clearly defined timelines, the project may tend to take on a life of its own, with never-ending meetings, discussions, and checklists, but never providing a measurable result.

Security professionals need to realize the importance of providing good support for this initial phase — recommending and describing the benefits of a good BCP program and explaining the technical challenges related to providing rapid data or processing recovery. As auditors, the emphasis is on having a solid project plan and budget responsibility so that the project meets its objectives within budget and on time.

Business Impact Analysis

The business impact analysis (BIA) phase examines each business unit to determine what impact a disaster or crisis may have on its operations. This means the business unit must define its core operations and, together with the IT group, outline its reliance on technology, the minimum requirements to maintain operations, and the maximum tolerable downtime (MTD) for its operations. The results of this effort are usually unique to each business unit within the corporation. The MTD can be dependant on costs (costs may begin to increase exponentially as the downtime increases), reputation (loss of credibility among customers, shareholders, regulatory agencies), or even technical issues (manufacturing equipment or data may be damaged or corrupted by an interruption in operations).

The IT group needs to work closely during this phase to understand the technological requirements of the business unit. From this knowledge, a list of alternatives for recovery processing can be established.

The audit group needs to ensure that proper focus is placed on the importance of each function. Not all departments are equally critical, and not all systems within a department are equally important. E-mail or Internet access may not be as important as availability of the customer database. The accounting department — despite its loud objections — may not need all of its functionality prioritized and provided the same day as the core customer support group. Audit can provide some balance and objective input to the recovery strategy and time frames through analysis and review of critical systems, highest impact areas, and objective consideration.

Design and Development

Once the BCP team understands the most critical needs of the business from both an operational and technology standpoint, it must consider how to provide a plan that will meet these needs within the critical timeframes of the MTD. There are several alternatives, depending on the type of disaster that occurs, but one alternative that should be considered is outsourcing of some operations. This can be the outsourcing of customer calls such as warranty claims to a call center, or outsourcing payroll or basic accounting functions.

Many organizations rely on a hot site or alternate processing facility to accommodate their information processing requirements. The IT group needs to be especially involved in working together with the business units to ensure that the most critical processing is provided at such a site without incurring expense for the usage of unnecessary processing or storage capability.

The audit group needs to ensure that the proper cost/benefit analysis has been done and that the provisions of the contract with the hot site are fulfilled and reasonable for the business needs.

The development of the business continuity plan must be reviewed and approved by the managers and representatives in the local business groups. This is where the continuous involvement of key people within these groups is beneficial. The ideal is to prepare a plan that is workable, simple, and timely. A plan that is too

cumbersome, theoretical, or unrelated to true business needs may well make recovery operations more difficult rather than expedite operational recovery.

During this phase is it noticed that, if the BCP process does not have an effective leader, key personnel will begin to drop out. No one has time for meaningless and endless meetings, and the key personnel from the business units need to be assured that their investment of time and input to the BCP project is time well spent.

Implementation of the Business Continuity Plan

All of the prior effort has been aimed at this point in time — the production of a workable result. That is, the production of a plan that can be relied on in a crisis to provide a framework for action, decision making, and definition of roles and responsibilities.

IT needs to review this plan to see its role. Can IT meet its objectives for providing supporting infrastructures? Does IT have access to equipment, backups, configurations, and personnel to make it all happen? Does IT have the contact numbers of vendors, suppliers, and key employees in off-site locations? Does the business unit know who to call in the area for support and interaction?

The audit group should review the finished product for consistency, completeness, management review, testing schedules, maintenance plans, and reasonable assumptions. This should ensure that the final product is reliable, that everyone is using the same version, that the plan is protected from destruction or tampering, and that it is kept in a secure format with copies available off-site.

Testing the Plans

Almost no organization can have just one recovery strategy. It is usual to have several recovery strategies based on the type of incident or crisis that affects the business. These plans need to be tested. Tests are verification of the assumptions, timelines, strategies, and responsibilities of the personnel tasked with executing a business continuity plan. Tests should not only consist of checks to see if the plan will work under ideal circumstances. Tests should stress the plan through unavailability of some key personnel and loss of use of facilities. The testing should be focused on finding weaknesses or errors in the plan structure. It is far better to find these problems in a sterile test environment than to experience them in the midst of a crisis.

The IT staff should especially test for validity of assumptions regarding providing or restoring equipment, data links, and communications links. They need to ensure that they have the trained people and plans to meet the restoration objectives of the plan.

Auditors should ensure that weaknesses found in the plans through testing are documented and addressed. The auditors should routinely sit in on tests to verify that the test scenario is realistic and that no shortcuts or compromises are made that could impair the validity of the test.

Maintenance of the BCP (Preserving the Value of the Investment)

A lot of money and time goes into the establishment of a good BCP program. The resulting plans are key components of an organization's survival plan. However, organizations and personnel change so rapidly that almost any BCP is out of date within a very short timeframe. It needs to be defined in the job descriptions of the BCP team members — especially the representatives from the business units — to provide continuous updates and modifications to the plan as changes occur in business unit structure, location, operating procedures, or personnel.

The IT group is especially vulnerable to outdating plans. Hardware and software change rapidly, and procurement of new products needs to trigger an update to the plan. When new products are purchased, consideration must be given to ensuring that the new products will not impede recovery efforts through unavailability of replacements, lack of standardization, or lack of knowledgeable support personnel.

Audit must review plans on a regular basis to see that the business units have maintained the plans and that they reflect the real-world environment for which the plans are designed. Audit should also ensure that adequate funding and support is given to the BCP project on an ongoing basis so that a workable plan is available when required.

Conclusion

A business continuity plan is a form of insurance for an organization — and, like insurance, we all hope that we never have to rely on it. However, proper preparation and training will provide the organization with a plan that should hold up and ease the pressures related to a crisis. A good plan should minimize the need to make decisions in the midst of a crisis and outline the roles and responsibilities of each team member so that the business can resume operations, restore damaged or corrupted equipment or data, and return to normal processing as rapidly and painlessly as possible.

140

The Business Impact Assessment Process

Carl B. Jackson, CISSP, CBCP

The initial version of this chapter was written for the 1999 edition of the *Handbook of Information Security Management*. Since then, Y2K has come and gone, E-commerce has seized the spotlight, and Web-based technologies are the emerging solution for almost everything. The constant throughout these occurrences is that no matter what the climate, fundamental business processes have changed little. And, as always, the focus of any business impact assessment is to assess the time-critical priority of these business processes. With these more recent realities in mind, this chapter has been updated and is now offered for your consideration.

The objective of this chapter is to examine the business impact assessment (BIA) process in detail and focus on the fundamentals of a successful BIA.

There is no question that business continuity planning (BCP) is a business process issue, not a technical one. Although each critical component of the enterprise must participate during the development, testing, and maintenance of the BCP process, it is the results of the business impact assessment (BIA) that will be used to make a case for further action.

Why perform a business impact assessment? The author's experiences in this area have shown that all too often, recovery strategies, such as hot sites, duplicate facilities, material or inventory stockpiling, etc., are based on emotional motivations rather than the results of a thorough business impact assessment. The key to success in performing BIAs lies in obtaining a firm and formal agreement from management as to the precise maximum tolerable downtimes (MTDs), also referred to in some circles as recovery time objectives (RTOs), for each critical business process. The formalized MTDs/RTOs, once determined, must be validated by each business unit, then communicated to the service organizations (i.e., IT, Network Management, Facilities, HR, etc.) that support the business units. This process helps ensure that realistic recovery alternatives are acquired and recovery measures are developed and deployed.

There are several reasons why a properly conducted and communicated BIA is so valuable to the organization. These include: (1) identifying and prioritizing time-critical business processes; (2) determining MTDs/RTOs for these processes and associated supporting resources, (3) raising positive awareness as to the importance of business continuity, and (4) providing empirical data upon which management can base its decision for establishing overall continuous operations and recovery strategies and acquiring supporting resources. Therefore, the significance of the BIA is that it sets the stage for shaping a business-oriented judgment concerning the appropriation of resources for recovery planning and continuous operations. (E-commerce — see below).

The Impact of the Internet and E-Commerce on Traditional BCP

Internet-enabled E-commerce has profoundly influenced the way organizations do business. This paradigm shift has dramatically affected how technology is used to support the organization's supply chain, and because of this, will also have a significant effect on the manner in which the organization views and undertakes business continuity planning. It is no longer a matter of just preparing to recover from a serious disaster or disruption. It is now incumbent upon technology management to do all it can to avoid any kind of outage whatsoever.

EXHIBIT 140.1 Continuous Availability/Recovery Planning Component Framework

Continuous Operations/Availability Disciplines	Traditional Recovery/BCP Disciplines
Current state assessment	Current state assessment
Business impact assessment	Business impact assessment
Leading practices/benchmarking	Leading practices/benchmarking
Continuous operations strategy development	Recovery strategy development
Continuous operations strategy deployment	Recovery plan development/deployment
Testing/maintenance	Testing/maintenance
Awareness/training	Awareness/training
Process measurement/metrics/value	Process measurement/metrics/value

The technical disciplines necessary to ensure continuous operations or E-availability include building redundancy, diversity, and security into the E-commerce-related supply-chain technologies (e.g., hardware, software, systems, and communications networks) (see Exhibit 140.1).

This framework attempts to focus attention on the traditional recovery planning process components as well as to highlight those process steps that are unique to the continuous operations/E-availability process.

The BCP professional must become conversant with the disciplines associated with continuous operations/E-availability in order to ensure that organizational E-availability and recovery objectives are met.

The BCP Process Approach

The BIA process is only one phase of recovery planning and E-availability. The following is a brief description of a six-phase methodological approach. This approach is commonly used for development of business unit continuity plans, crisis management plans, technological platform, and communications network recovery plans.

- Phase I — Determine scope of BCP project and develop project plan. This phase examines business operations and information system support services, in order to form a project plan to direct subsequent phases. Project planning must define the precise scope, organization, timing, staffing, and other issues. This enables articulation of project status and requirements throughout the organization, chiefly to those departments and personnel who will be playing the most meaningful roles during the development of the BCP.
- Phase II — Conduct business impact assessment. This phase involves identification of time-critical business processes, and determines the impact of a significant interruption or disaster. These impacts may be financial in terms of dollar loss, or operational in nature, such as the ability to deliver and monitor quality customer service, etc.
- Phase III — Develop recovery/E-availability strategies. The information collected in Phase II is employed to approximate the recovery resources (i.e., business unit or departmental space and resource requirements, technological platform services, and communications networks requirements) necessary to support time-critical business processes and sub-processes. During this phase, an appraisal of E-availability/recovery alternatives and associated cost estimates are prepared and presented to management.
- Phase IV — Perform recovery plan development. This phase develops the actual plans (i.e., business unit, E-availability, crisis management, technology-based plans). Explicit documentation is required for execution of an effective recovery process. The plan must include administrative inventory information and detailed recovery team action plans, among other information.
- Phase V — Implement, test, and maintain the BCP. This phase establishes a rigorous, ongoing testing and maintenance management program.
- Phase VI — Implement awareness and process measurement. The final and probably the most crucial long-term phase establishes a framework for measuring the recovery planning and E-availability processes against the value they provide the organization. In addition, this phase includes training of personnel in the execution of specific continuity/recovery activities and tasks. It is vital that they are aware of their role as members of E-availability/recovery teams.

BIA Process Description

As mentioned above, the intent of the BIA process is to assist the organization's management in understanding the impacts associated with possible threats. Management must then employ that intelligence to calculate the maximum tolerable downtime (MTD) for time-critical support services and resources. For most organizations, these resources include:

1. Personnel
2. Facilities
3. Technological platforms (traditional and E-commerce-related systems)
4. Software
5. Data networks and equipment
6. Voice networks and equipment
7. Vital records
8. Data
9. Supply chain partners

The Importance of Documenting a Formal MTD/RTO Decision

The BIA process concludes when executive management makes a formalized decision as to the MTD it is willing to live with after analyzing the impacts to the business processes due to outages of vital support services. This includes the decision to communicate these MTD decision(s) to each business unit and support service manager involved.

The Importance of a Formalized Decision

A formalized decision must be clearly communicated by senior management because the failure to document and communicate precise MTD information leaves each manager with imprecise direction on: (1) selection of an appropriate recovery alternative method; and (2) the depth of detail that will be required when developing recovery procedures, including their scope and content.

The author has seen many well-executed BIAs with excellent results wasted because senior management failed to articulate its acceptance of the results and communicate to each affected manager that the time requirements had been defined for recovery processes.

BIA Information-Gathering Techniques

There are various schools of thought regarding how best to gather BIA information.Conducting individual one-on-one BIA interviews is popular, but organizational size and location issues sometimes make conducting one-on-one interviews impossible. Other popular techniques include group sessions, the use of an electronic medium (i.e., data or voice network), or a combination of all of these. Exhibit 140.2 is a BIA checklist. The following points highlight the pros and cons of these interviewing techniques:

1. *One-on-one BIA interviews.* In the author's opinion, the one-on-one interview with organizational representatives is the preferred manner in which to gather BIA information. The advantages of this method are the ability to discuss the issues face-to-face and observe the person. This one-on-one discussion will give the interviewer a great deal of both verbal and visual information concerning the topic at hand. In addition, personal rapport can be built between the interviewee and the BIA team, with the potential for additional assistance and support to follow. This rapport can be very beneficial during later stages of the BCP development effort if the person being interviewed understands that the BCP process was undertaken to help them get the job done in times of emergency or disaster. The disadvantages of this approach are that it can become very time-consuming, and can add time to the critical path of the BIA process.
2. *Group BIA interview sessions or exercises.* This type of information-gathering activity can be very efficient in ensuring that a lot of data is gathered in a short period of time and can speed the BIA

EXHIBIT 140.2 BIA Dos Checklist

BIA To Dos

- Customize the BIA information-gathering tools questions to suit the organization's customs/culture.
- Focus on time-critical business processes and support resources (i.e., systems, applications, voice and date networks, facilities, people, etc.).
- Assume worst-case disaster (day of week, month of year, etc.).
- Assume no recovery capability exists.
- Obtain raw numbers in orders of magnitude.
- Return for financial information.
- Validate BIA data with BIA participants.
- Formalize decision from senior management so lower-level managers (MTD timeframes, scope, and depth of recovery procedures, etc.) can make precise plans.

Conducting BIA Interviews

- When interviewing business unit personnel, explain that you are here to get the information you need to help IT build their recovery plan. But emphasize that the resulting IT recovery is really theirs, and the recovery plan is really yours. One is obtaining their input as an aid in ensuring that MIS constructs the proper recovery planning strategy.
- Interviews last no longer that 45 minutes to 1 hour and 15 minutes.
- The number of interviewees at one session should be at best one, and at worst two to three. More than that and the ability of the individual to take notes is questionable.
- If possible, at least two personnel should be in attendance at the interview. Each should have a blank copy of the questionnaire on which to take notes.
- One person should probably not perform more than four interviews per day. This is due to the requirement to successfully document the results of each interview as soon as possible and because of fatigue factors.
- Never become confrontational with the interviewees. There is no reason that interviewees should be defensive in their answers unless they do not properly understand the purpose of the BIA interview.
- Relate to interviewees that their comments will be taken into consideration and documented with the others gathered. And that they will be requested to review, at a later date, the output from the process for accuracy and provide their concurrence.

process tremendously. The drawback to this approach is that if not conducted properly, it can result in a meeting of a number of people without very much useful information being obtained.

3. *Executive management mandate.* Although not always recommended, there may be certain circumstances where conducting only selected interviews with very high-level executive management will suffice for BIA purposes. Such situations might include development of continuous operations/E-availability strategies where extremely short recovery timeframes are already obvious, or where times for development of appropriate strategies for recovery are severely shortened (as in the Y2K recovery plan development example). The level of confidence is not as high in comparison to performing many more exhaustive sets of interviews (at various levels of the organization, not just with the senior management group), but it does speed up the process.
4. *Electronic medium.* Use of voice and data communications technologies, videoconferencing, and Web-based technologies and media are becoming increasingly accepted and popular. Many times, the physical or geographical size and diversity, as well as the structural complexity of the organization, lends itself to this type of information-gathering technique. The pros are that distances can be diminished and travel expenses reduced. The use of automated questionnaires and other data-gathering methods can facilitate the capture of tabular data and ease consolidation of this information. Less attractive, however,

is the fact that this type of communication lacks the human touch, and sometimes ignores the importance of the ability of the interviewer to read the verbal and visual communications of the interviewee. *Note:* Especially worrisome is the universal broadcast of BIA-related questionnaires. These inquiries are sent to uninformed groups of users on a network, whereby they are asked to supply answers to qualitative and quantitative BIA questions without regard to the point or nuance of the question or the intent of the use of the result. Such practices almost always lend themselves to misleading and downright wrong results. This type of unsupported data-gathering technique for purposes of formulating a thoughtful strategy for recovery should be avoided.

Most likely, an organization will need to use a mix of these suggested methods, or use others as suited to the situation and culture of the enterprise.

The Use of BIA Questionnaires

There is no question that the people-to-people contact of the BIA process is *the* most important component in understanding the potential a disaster will have on an organization. People run the organization, and people can best describe business functionality and their business unit's degree of reliance on support services. The issue here, however, is deciding what is the best and most practical technique for gathering information from these people.

There are differing schools of thought regarding the use of questionnaires during the BIA process. The author's opinion is that a well-crafted and customized BIA questionnaire will provide the structure needed to guide the BIA and E-availability project team(s). This consistent interview structure requires that the same questions be asked of each BIA interviewee. Reliance can then be placed on the results because answers to questions can be compared to one another with assurance that the comparisons are based on the same criterion.

Although a questionnaire is a valuable tool, the structure of the questions is subject to a great deal of customization. This customization of the questions depends largely on the reason why the BIA is being conducted in the first place.

The BIA process can be approached differently, depending on the needs of the organization. Each BIA situation should be evaluated in order to properly design the scope and approach of the BIA process. BIAs are desirable for several reasons, including:

1. Initiation of a BCP process where no BIA has been done before, as part of the phased implementation methodology
2. Reinitiating a BCP process where there was a BIA performed in the past, but now it needs to be brought up to date
3. Conducting a BIA in order to incorporate the impacts of a loss of E-commerce-related supply-chain technologies into the overall recovery strategies of the organization
4. Conducting a BIA in order to justify BCP activities that have already been undertaken (i.e., the acquisition of a hotsite or other recovery alternative)
5. Initiating a BIA as a prelude to beginning a full BCP process for understanding or as a vehicle to sell management on the need to develop a BCP

Customizing the BIA Questionnaire

There are a number of ways that a questionnaire can be constructed or customized to adapt itself for the purpose of serving as an efficient tool for accurately gathering BIA information. There are also an unlimited number of examples of BIA questionnaires in use by organizations. It should go without saying that any questionnaire — BIA or otherwise — can be constructed so as to elicit the response one would like. It is important that the goal of the BIA be in the mind of the questionnaire developers so that the questions asked and the responses collected will meet the objective of the BIA process.

BIA Questionnaire Construction

Exhibit 140.3 is an example of a BIA questionnaire. Basically, the BIA questionnaire is made up of the following types of questions:

EXHIBIT 140.3 Sample BIA Questionnaire

Introduction
Business Unit Name:

Date of Interview:

Contact Name(s):

Identification of business process and/or business unit (BU) function:

Briefly describe the overall business functions of the BU (with focus on time-critical functions/processes, and link each time-critical function/process to the IT application/network, etc.) and understanding of business process and applications/networks, etc. interrelationships:

Financial Impacts
Revenue Loss Impacts Estimations (revenue or sales loss, lost trade discounts, interest paid on borrowed money, interest lost on float, penalties for late payment to vendors or lost discounts, contractual fines or penalties, unavailability of funds, canceled orders due to late delivery, etc.):

Extraordinary expense impact estimations (acquisition of outside services, temporary employees, emergency purchases, rental/lease equipment, wages paid to idle staff, temporary relocation of employees, etc.):

Operational Impacts
Business interruption impact estimations (loss of customer service capabilities, inability to serve internal customers/management/etc.):

Loss of confidence estimations (loss of confidence on behalf of customers/shareholders/regulatory agencies/employees, etc.):

Technological Dependence
Systems/business functions/applications reliance description (attempt to identify specific automated systems/processes/applications that support BU operations):

Systems interdependencies descriptions:

State of existing BCP measures:

Other BIA-related discussion issues:

First question phrased: "What else should I have asked you that I did not, relative to this process?"

Other questions customized to environment of the organization, as needed:

- *Quantitative questions.* These are the questions asked the interviewee to consider and describe the economic or financial impacts of a potential disruption. Measured in monetary terms, an estimation of these impacts will aid the organization in understanding loss potential, in terms of lost income as well as in an increase in extraordinary expense. The typical quantitative impact categories might include revenue or sales loss, lost trade discounts, interest paid on borrowed money, interest lost on float, penalties for late payment to vendors or lost discounts, contractual fines or penalties, unavailability of funds, canceled orders due to late delivery, etc. Extraordinary expense categories might include acqui-

sition of outside services, temporary employees, emergency purchases, rental/lease equipment, wages paid to idle staff, and temporary relocation of employees.

- *Qualitative questions.* Although the economic impacts can be stated in terms of dollar loss, the qualitative questions ask the participants to estimate potential loss impact in terms of their emotional understanding or feelings. It is surprising how often the qualitative measurements are used to put forth a convincing argument for a shorter recovery window. The typical qualitative impact categories might include loss of customer services capability, loss of confidence, etc.
- *Specialized questions.* Make sure that the questionnaire is customized to the organization. It is especially important to make sure that both the economic and operational impact categories (lost sales, interest paid on borrowed funds, business interruption, customer inconvenience, etc.) are stated in such a way that each interviewee will understand the intent of the measurement. Simple is better here.

Using an Automated Tool

If an automated tool is being used to collect and correlate the BIA interview information, make sure that the questions in the database and questions of the questionnaire are synchronized to avoid duplication of effort or going back to interviewees with questions that might have been handled initially.

A word of warning here, however. This author has seen people pick up a BIA questionnaire off the Internet or from book or periodical (like this one) and use it without regard to the culture and practices of their own organization. Never, ever, use a noncustomized BIA questionnaire. The qualitative and quantitative questions must be structured to the environment and style of the organization. There is a real opportunity for failure should this point be dismissed.

BIA Interview Logistics and Coordination

This portion of the report will address the logistics and coordination while performing the BIA interviews themselves. Having scoped the BIA process, the next step is to determine who and how many people one is going to interview. To do this, here are some techniques that one might use.

Methods for Identifying Appropriate BIA Interviewees

One certainly is not going to interview everyone in the organization. One must select a sample of those management and staff personnel who will provide the best information in the shortest period. To do that, one must have a precise feel for the scope of the project (i.e., technological platform recovery, business unit recovery, communications recovery, crisis management plans, etc.) and with that understanding one can use:

- *Organizational process models.* Identification of organizational mega and major business processes is the first place to start. Enterprises that are organized along process lines lend themselves to development of recovery planning strategies that will eventually result in the most efficient recovery infrastructure. Use of or development of models that reflect organizational processes will go a long way toward assisting BIA team members in identifying those personnel crucial to determining time-critical process requirements. Exhibit 140.4 attempts to demonstrate that while the enterprisewide recovery planning/E-continuity infrastructure includes consideration of crisis management, technology disaster recovery, business unit resumption, and E-commerce E-availability components, all aspects of the resulting infrastructure flow from proper identification of time-critical business processes.
- *Organizational chart reviews.* The use of formal, or sometimes even informal organization charts is the first place to start. This method includes examining the organizational chart of the enterprise to understand those functional positions that should be included. Review the organizational chart to determine which organizational structures will be directly involved in the overall effort as well as those that will be the recipients of the benefits of the finished recovery plan.
- *Overlaying systems technology.* Overlay systems technology (applications, networks, etc.) configuration information over the organization chart to understand the components of the organization that may be affected by an outage of the systems. Mapping applications, systems, and networks to the organizations business functions will help tremendously when attempting to identify the appropriate names and numbers of people to interview.

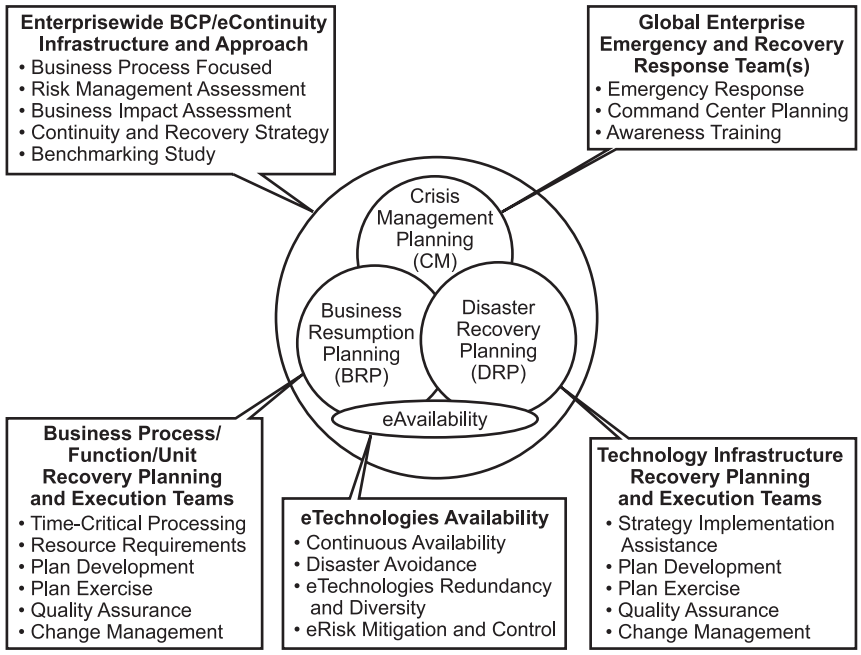


EXHIBIT 140.4 Enterprisewide BCP/E-continuity infrastructure.

- *Executive management interviews.* This method includes conducting introductory interviews with selected senior management representatives in order to identify critical personnel to be included in the BIA interview process, as well as to receive high-level guidance and to raise overall executive-level management awareness and support.

Coordinate with the IT Group

If the scope of the BIA process is recovery of technological platforms or communications systems, then conducting interviews with a number of IT personnel could help shorten the data-gathering effort. Although IT users will certainly need to be spoken to, IT personnel can often provide much valuable information, but should not be solely relied on as the primary source of business impact outage information (i.e., revenue loss, extra expense, etc.).

Send Questionnaire Out in Advance

It is a useful technique to distribute the questionnaire to the interviewees in advance. Whether in hardcopy or electronic media format, the person being interviewed should have a chance to review the questions, and be able to invite others into the interview or redirect the interview to others, and begin to develop the responses. One should emphasize to the people who receive the questionnaire in advance to not fill it out, but to simply review it and be prepared to address the questions.

Scheduling of Interviews

Ideally, the BIA interview should last between 45 minutes and 1 hour and 15 minutes. It sometimes can be an advantage to go longer than this; but if one sees many of the interviews lasting longer than the 1 hour, 15 minute window, there may be a BIA scoping issue that should be addressed, necessitating the need to schedule and conduct a larger number of additional interviews.

Limit Number of Interviewees

It is important to limit the number of interviewees in the session to one, two, or three, but no more. Given the amount and quality of information one is hoping to elicit from this group, more than three people can deliver a tremendous amount of good information that can be missed when too many people are delivering the message at the same time.

Try to Schedule Two Interviewers

When setting up the BIA interview schedule, try to ensure that at least two interviewers can attend and take notes. This will help eliminate the possibility that good information may be missed. Every additional trip back to an interviewee for confirmation of details will add overhead to the process.

Validate Financial Impact Thresholds

An often-overlooked component of the process includes discussing with executive management the thresholds of pain that could be associated with a disaster. Asking the question as to whether a $5 million loss or a $50 million loss impact has enough significance to the long-term bottom line of the organization can lead to interesting results. A lack of understanding on the BIA team's part as to what financial impacts are acceptable, or conversely unacceptable, is crucial to framing BIA financial loss questions and the final findings and recommendations that the BIA report will reflect.

Conducting the BIA

When actually explaining the intent of the BIA to those being interviewed, the following concepts should be observed and perhaps discussed with the participants.

Intelligent Questions Asked of Knowledgeable People

Based loosely on the concept that if one asks enough reasonably intelligent people a consistent set of measurable questions, one will eventually reach a conclusion that is more or less correct. The BIA questions serve to elicit qualitative results from a number of knowledgeable people. The precise number of people interviewed obviously depends on the scope of the BCP activity and the size of the organization. However, when consistently directing a well-developed number of questions to an informed audience, the results will reflect a high degree of reliability. This is the point when conducting qualitatively oriented BIA: ask the right people good questions and one will come up with the right results.

Ask to Be Directed to the Correct People

As the interview unfolds, it may become evident that the interviewee is the wrong person to be answering the questions. One should ask who else within this area would be better suited to address these issues. They might be invited into the room at that point, or one may want to schedule a meeting with them at another time.

Assure Them that Their Contribution Is Valuable

A very important way to build the esteem of the interviewee is to mention that their input to this process is considered valuable, as it will be used to formulate strategies necessary to recover the organization following a disruption or disaster. Explaining to them that one is there to help by getting their business unit's relevant information for input to planning a recovery strategy can sometimes change the tone of the interview in a positive manner.

Explain that the Plan Is Not Strictly an IT Plan

Even if the purpose of the BIA is for IT recovery and, when interviewing business unit management for the process of preparing a technological platform recovery plan, it is sometimes useful to couch the discussion in

terms of … "a good IT recovery plan, while helping IT recover, is really a business unit plan … Why? … Because the IT plan will recover the business functionality of the interviewees business unit as well, and that is why one is there."

Focus on Who Will Really Be Exercising the Plan

Another technique is to mention that the recovery plan that will eventually be developed can be used by the interviewees, but is not necessarily developed for them. Why? Because the people being interviewed probably already understand what to do following a disaster, without having to refer to extensive written recovery procedures. But the fact of the matter is that following the disruption, these people may not be available. It may well be the responsibility of the next generation of management to recover, and it will be the issues identified by this interviewee that will serve as the recovery roadmap.

Focus on Time-Critical Business Processes and Support Resources

As the BIA interview progresses, it is important to fall back from time to time and reinforce the concept of being interested in the identification of time-critical functions and processes.

Assume Worst-Case Disaster

When faced with the question as to when the disruption will occur, the answer should be: "It will occur at the worst possible time for your business unit. If you close your books on 12/31, and you need the computer system the most on 12/30 and 12/31, the disaster will occur on 12/29." Only when measuring the impacts of a disruption at the worst time can the interviewer get an idea as to the full impact of the disaster, and so that the impact information can be meaningfully compared from one business unit to the next.

Assume No Recovery Capability Exists

To reach results that are comparable, it is essential to insist that the interviewee assume that no recovery capability will exist as they answer the impact questions. The reason for this is that when they attempt to quantify or qualify the impact potential, they may confuse a preexisting recovery plan or capability with no impact, and that is incorrect. No matter the existing recovery capability, the impact of a loss of services must be measured in raw terms so that as one compares the results of the interviews from business unit to business unit, the results are comparable (apples to apples, so to speak). Exhibit 140.5 provides an example. In this example, if one allows Interviewees #2 and #4 to assume that they can go somewhere else and use an alternate resource to support their process, the true impact of the potential disruption is reduced by one-half ($40K vs. $80K). By not allowing them to assume that an appropriate recovery alternative exists, one will recognize the true impact of a disruption, that of $80,000 per-day. The $80,000-per day impact is what one is trying to understand, whether or not a recovery alternative already exists.

EXHIBIT 140.5 Comparing the Results of the Interviews

Interviewee	Total Loss Impact if Disaster?	Preconceived Recovery Alternative?	Resulting Estimated Loss Potential	No Allowance for Preconceived Recovery Alternative
#1	$20K per day	No	$20,000	$20,000
#2	$20K per day	Yes	0	20,000
#3	$20K per day	No	20,000	20,000
#4	$20K per day	Yes	0	20,000
Totals	—	—	$40,000[a]	$80,000[b]

[a] Incorrect estimate, as one should not allow the interviewee to assume a recovery alternative exists (although one may very well exist).

[b] Correct estimate, based on raw loss potential regardless of preexisting recovery alternatives (which may or may not be valid should a disruption or disaster occur).

Order-of-Magnitude Numbers and Estimates

The financial impact information is needed in orders-of-magnitude estimates only. Do not get bogged down in minutia, as it is easy to get lost in the detail. The BIA process is not a quantitative risk assessment. It is not meant to be. It is qualitative in nature and, as such, orders-of-magnitude impacts are completely appropriate and even desirable. Why? Because preciseness in estimation of loss impact almost always results in arguments about the numbers. When this occurs, the true goal of the BIA is lost, because it turns the discussion into a numbers game, not a balanced discussion concerning financial and operational impact potentials. Because of the unlimited and unknown numbers of varieties of disasters that could possibly befall an organization, the true numbers can never ever be precisely known, at least until after the disaster. The financial impact numbers are merely estimates intended to illustrate degrees of impacts. So skip the numbers exercise and get to the point.

Stay Focused on the BCP Scope

Whether the BIA process is for development of technological platforms, end user, facilities recovery, voice network, etc., it is very important that one not allow scope creep in the minds of the interviewees. The discussion can become very unwieldy if one does not hold the focus of the loss impact discussions on the precise scope of the BCP project.

There Are No Wrong Answers

Because all the results will be compared with one another before the BIA report is forwarded, one can emphasize that the interviewee should not worry about wrong numbers. As the BIA process evolves, each business unit's financial and operational impacts will be compared with the others, and those impact estimates that are out of line with the rest will be challenged and adjusted accordingly.

Do Not Insist on Getting the Financial Information on the Spot

Sometimes, the compilation of financial loss impact information requires a little time to accomplish. The author often tells the interviewee that he will return within a few days to collect the information, so that additional care can be taken in preparation, making sure that he does actually return and picks up the information later.

The Value of Pushback

Do not underestimate the value of pushback when conducting BIA interviews. Business unit personnel will, most times, tend to view their activities as extremely time-critical, with little or no downtime acceptable. In reality, their operations will be arranged in some priority order with the other business processes of the organization for recovery priority. Realistic MTDs must be reached, and sometimes the interviewer must push back and challenge what may be considered unrealistic recovery requirements. Be realistic in challenging, and request that the interviewee be realistic in estimating their business unit's MTDs. Common ground will eventually be found that will be more meaningful to those who will read the *BIA Findings and Recommendations* — the senior management group.

Interpreting and Documenting the Results

As the BIA interview information is gathered, there is a considerable tabular and written information that begins to quickly accumulate. This information must be correlated and analyzed. Many issues will arise here that may result in some follow-up interviews or information-gathering requirements. The focus at this point in the BIA process should be as follows.

Begin Documentation of the Results Immediately

Even as the initial BIA interviews are being scheduled and completed, it is a good idea to begin preparation of the *BIA Findings and Recommendations* and actually start entering preliminary information. The reason is

twofold. The first is that if one waits to the end of the process to start formally documenting the results, it is going to be more difficult to recall details that should be included. Second, as the report begins to evolve, there will be issues that arise where one will want to perform additional investigation, while one still has time to ensure the investigation can be thoroughly performed.

Develop Individual Business Unit BIA Summary Sheets

Another practical technique is to document each and every BIA interview with its own *BIA Summary Sheet*. This information can eventually be used directly by importing it into the *BIA Findings and Recommendations,* and can also be distributed back out to each particular interviewee to authenticate the results of the interview. The *BIA Summary Sheet* contains a summation of all the verbal information that was documented during the interview. This information will be of great value later as the BIA process evolves.

Send Early Results Back to Interviewees for Confirmation

By returning the *BIA Summary Sheet* for each of the interviews back to the interviewee, one can continue to build consensus for the BCP project and begin to ensure that any future misunderstandings regarding the results can be avoided. Sometimes, one may want to get a formal sign-off, and other times the process is simply informal.

We Are Not Trying to Surprise Anyone

The purpose for diligently pursuing the formalization of the BIA interviews and returning to confirm the understandings from the interview process is to make very sure that there are no surprises later. This is especially important in large BCP projects where the BIA process takes a substantial amount of time. There is always a possibility that someone might forget what was said.

Definition of Time-Critical Business Functions/Processes

As has been emphasized, all issues should focus back to the true time-critical business processes of the organization. Allowing the attention to be shifted to specific recovery scenarios too early in the BIA phase will result in confusion and lack of attention toward what is really important.

Tabulation of Financial Impact Information

There can be a tremendous amount of tabular information generated through the BIA process. It should be boiled down to its essence and presented in such a way as to support the eventual conclusions of the BIA project team. It is easy to overdo it with numbers. Just ensure that the numbers do not overwhelm the reader and that they fairly represent the impacts.

Understanding the Implications of the Operational Impact Information

Often times, the weight of evidence and the basis for the recovery alternative decision are based on operational rather than the financial information. Why? Usually, the financial impacts are more difficult to accurately quantify because the precise disaster situation and the recovery circumstances are difficult to visualize. One knows that there will be a customer service impact because of a fire, for example. But one would have a difficult time telling someone, with any degree of confidence, what the revenue loss impact would be for a fire that affects one particular location of the organization. Because the BIA process should provide a qualitative estimate (orders of magnitude), the basis for making the difficult decisions regarding acquisition of recovery resources are, in many cases, based on the operational impact estimates rather than hard financial impact information.

Preparing the Management Presentation

Presentation of the results of the BIA to concerned management should result in no surprises for them. If one is careful to ensure that the BIA findings are communicated and adjusted as the process has unfolded, then

EXHIBIT 140.6 BIA Report Table of Contents

1. Executive Summary
2. Background
3. Current State Assessment
4. Threats and Vulnerabilities
5. Time-Critical Business Functions
6. Business Impacts (Operational)
7. Business Impacts (Financial)
8. Recovery Approach
9. Next Steps/Recommendations
10. Conclusion
11. Appendices (as needed)

the management review process should really become more of a formality in most cases. The final presentation meeting with the senior management group is not the time to surface new issues and make public startling results for the first time.

To achieve the best results in the management presentation, the following suggestions are offered.

Draft Report for Review Internally First

Begin drafting the report following the initial interviews. By doing this, one captures fresh information. This information will be used to build the tables, graphs, and other visual demonstrations of the results, and it will be used to record the interpretations of the results in the verbiage of the final *BIA Findings and Recommendations Report*. One method for accomplishing a well-constructed *BIA Findings and Recommendations* from the very beginning is to, at the completion of each interview, record the tabular information into the BIA database or manual filing system in use to record this information. Second, the verbal information should be transcribed into a *BIA Summary Sheet* for each interview. This *BIA Summary Sheet* should be completed for each interviewee and contain the highlights of the interview in summarized form. As the BIA process continues, the BIA tabular information and the transcribed verbal information can be combined into the draft *BIA Findings and Recommendations*. The table of contents for a BIA Report might look like the one depicted in Exhibit 140.6.

Schedule Individual Senior Management Meetings as Necessary

Near the time for final BIA presentation, it is sometimes a good idea to conduct a series of one-on-one meetings with selected senior management representatives in order to brief them on the results and gather their feedback for inclusion in the final deliverables. In addition, this is a good time to begin building grassroots support for the final recommendations that will come out of the BIA process and at the same time provide an opportunity to practice making one's points and discussing the pros and cons of the recommendations.

Prepare Senior Management Presentation (Bullet Point)

The author's experience reveals that senior management-level presentations, most often, are better prepared in a brief and focused manner. It will undoubtedly become necessary to present much of the background information used to make the decisions and recommendations, but the formal presentation should be in bullet-point format, crisp, and to the point. Of course, every organization has its own culture, so be sure to understand and comply with the traditional means of making presentations within that environment. Copies of the report, which have been thoroughly reviewed, corrected, bound, and bundled for delivery, can be distributed at the beginning or end of the presentation, depending on circumstances. In addition, copies of the bullet-point handouts can also be supplied so attendees can make notes and for reference at a later time. Remember, the BIA process should end with a formalized agreement as to management's intentions with regard to MTDs, so that business unit and support services managers can be guided accordingly. It is here that that formalized agreement should be discussed and the mechanism for acquiring and communicating it determined.

Distribute Report

Once the management team has had an opportunity to review the contents of the BIA Report and made appropriate decisions or given other input, the final report should be distributed within the organization to the appropriate numbers of interested individuals.

Past Y2K and Current E-availability Considerations

The author's experience with development of Y2K-related recovery plans was that time was of the essence. Because of the constricted timeframe for development of Y2K plans, it was necessary to truncate the BIA process as much as possible to meet timelines. Modification of the process to shorten the critical path was necessary — resulting in several group meetings focusing on a very selective set of BIA criteria.

Limit Interviews and Focus on Upper-Level Management

To become a little creative in obtaining BIA information in this Y2K example, it was necessary to severely limit the number of interviews and to interview higher-level executives to receive overall guidance, and then move to recovery alternative selection and implementation rapidly.

Truncated BIAs for E-availability Application

Additionally, when considering gathering BIA information during an E-availability application, it is important to remember that delivery of E-commerce-related services through the Internet means that supply-chain downtime tolerances — including E-commerce technologies and channels — are usually extremely short (minutes or even seconds), and that it may not be necessary to perform an exhaustive BIA to determine the MTD/RTO only. What is necessary for a BIA under these circumstances, however, is that it helps to determine which business processes truly rely on E-commerce technologies and channels so that they (business unit personnel) can be prepared to react in a timely manner should E-commerce technologies be impacted by a disruption or disaster.

Next Steps

The BIA is truly completed when formalized senior management decisions have been made regarding: (1) MTDs/RTOs, (2) priorities for business process and support services recovery, and (3) recovery/E-availability resource funding sources.

The next step is the selection of the most effective recovery alternative. The work gets a little easier here. One knows what the recovery windows are, and one understands what the recovery priorities are. One must now investigate and select recovery alternative solutions that fit the recovery window and recovery priority expectations of the organization. Once the alternatives have been agreed upon, the actual recovery plans can be developed and tested, with organization personnel organized and trained to execute the recovery plans when needed.

Summary

The process of business continuity planning has matured substantially since the 1980s. BCP is no longer viewed as just a technological question. A practical and cost-effective approach toward planning for disruptions or disasters begins with the business impact assessment. In addition, the rapidly evolving dependence on E-commerce-related supply-chain technologies has caused a refocus of the traditional BCP professional on not only recovery, but also continuous operations or E-availability imperatives.

The goal of the BIA is to assist the management group in identifying time-critical processes, and determining their degree of reliance on support services. Then, map these processes to supporting IT, voice and data networks, facilities, human resources, E-commerce initiatives, etc. Time-critical business processes are prior-

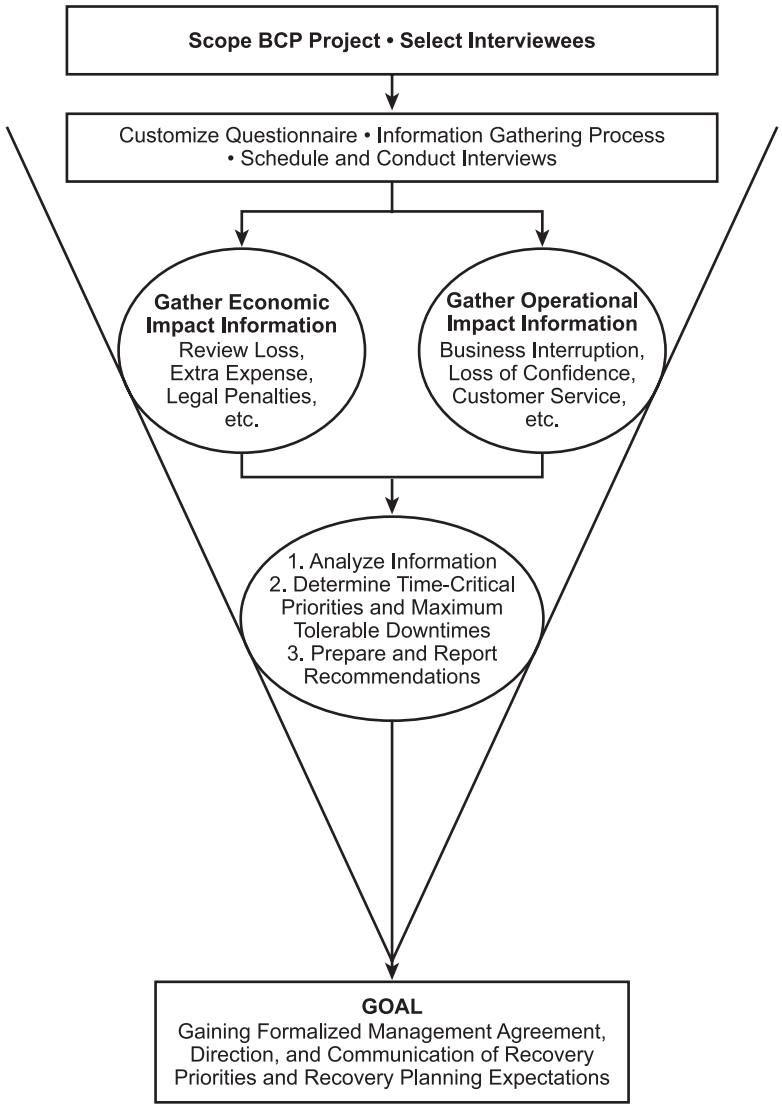


EXHIBIT 140.7 Business continuity planning route map.

itized in terms of their MTDs/RTOs, so that executive management can make reasonable decisions as to the recovery costs and timeframes that it is willing to fund and support.

This chapter has focused on how organizations can facilitate the BIA process. See the BCP Route Map in Exhibit 140.7 for a pictorial representation of the BIA process. Understanding and applying the various methods and techniques for gathering the BIA information is the key to success.

Only when executive management formalizes its decisions regarding recovery timeframes and priorities can each business unit and support service manager formulate acceptable and efficient plans for recovery of operations in the event of disruption or disaster. It is for this reason that the BIA process is so important when developing efficient and cost-effective business continuity plans and E-availability strategies.

Domain 9
Law, Investigations, and Ethics

The Law, Investigations, and Ethics Domain addresses computer crime laws and regulations. It reviews investigative measures and techniques used to determine if a crime has been committed and methods to gather evidence. It also reviews the ethical constraints that provide a code of conduct for the security professional.

In this domain we discuss methods for determining if a computer crime has been committed and the laws that are applicable for the crime. We examine laws prohibiting specific types of computer crime and methods to gather and preserve evidence of a computer crime. We review investigative methods and techniques. Finally, we study ways in which RFC 1087 and the $(ISC)^2$ Code of Ethics can be applied to resolve ethical dilemmas.

Contents

Section 9.4 Incident Handling

Section 9.5 Ethics

141

Jurisdictional Issues in Global Transmissions

Ralph Spencer Poore, CISSP, CISA, CFE

In the information age where teleconferences replace in-person meetings, where telecommuting replaces going to the office, and where international networks facilitate global transmissions with the apparent ease of calling your neighbor, valuable assets change ownership at the speed of light. Louis Jionet, secretary-general of the French Commission on Data Processing and Liberties stated: "Information is power and economic information is economic power." Customs officials and border patrols cannot control the movement of these assets. But does this mean companies may transmit the data which either represents or is *the* valuable asset without regard to the legal jurisdictions through which they pass? To adequately address this question we will discuss both the legal issues and the practical issues involved in transnational data flows.

Legal Issues

All legally incorporated enterprises have official books of record. Whether these are in manual or automated form, they are the records governmental authorities turn to when determining the status of an enterprise. The ability to enforce a subpoena or court order for these records reflects the effective sovereignty of the nation in which the enterprise operates. Most countries require enterprises incorporated, created, or registered in their jurisdiction to maintain official books of record physically within their borders. For example, a company relying on a service bureau in another country for data processing services may cause the official records to exist only in that other country. This could occur if the printouts reflected only a historical position of the company, perhaps month-end conditions, where the current position of the company — the position on which management relies — exists only through online access to the company's executive information system. From a nation's perspective, two issues of sovereignty arise:

1. That other country might exercise its rights and take custody of the company's records — possibly forcing it out of business — for actions alleged against the company that the company's "home" nation considers legal.
2. The company's "home" nation may be unable to enforce its access rights.

Another, usually overriding factor is a nation's ability to enforce its tax laws. Many nations have value-added taxes (VATs) or taxes on "publications," "computer software," and "services." Your organization's data may qualify as a "publication" or as "computer software" or even as "services" in some jurisdictions. Thus, many nations have an interest in the data that flows across their borders because it may qualify for taxation. In some cases, the tax is a tariff intended to discourage the importation of "computer software" or "publications" in order to protect the nation's own emerging businesses. More so than when the tax is solely for revenue generation, protective tariffs may carry heavy fines and be more difficult to negotiate around. With the advent of Internet businesses, determining a business' nexus for tax purposes has become even more complex. Such

businesses may have income, franchise, and inventory or property tax issues in addition to sales tax, excise tax, and import or export duties. Business taxes, registration or license fees, and even reporting requirements depend on the applicability of a given jurisdiction.

National security interests may include controlling the import and export of information. State secrecy laws exist for almost all nations. The United States, for example, restricts government-classified data (e.g., Confidential, Secret, Top Secret), but also restricts some information even if it is not classified (e.g., technical data about nuclear munitions, some biological research, some advanced computer technology, and, to varying degrees, cryptography).

Among those nations concerned with an individual's privacy rights, the laws vary greatly. Laws like the United States' Privacy Act of 1974 (5 USC 552a) have limited applicability (generally applying only to government agencies and their contractors). The United Kingdom's Data Protection Act of 1984 (1984 c 35 [*Halsbury's Statutes 4th Edition*, Butterworths, London, 1992, vol. 6, pp. 899–949]), however, applies to the commercial sector as does the 1981 Council of Europe's Convention for the Protection of Individuals with Regard to Automatic Processing of Personal Data. (An excellent discussion of this can be found in Anne W. Brandscomb's *Toward a Law of Global Communications Networks*, The Science and Technology section of the American Bar Association, Longman, New York, 1986.) Privacy laws generally have at least the following three characteristics:

1. They provide notice to the subject of the existence of a database containing the subject's personal data (usually by requiring registration of the database).
2. They provide a process for the subject to inspect and to correct the personal data.
3. They provide a requirement for maintaining an audit trail of accessors to the private data.

The granularity of privacy law requirements also varies greatly. Some laws, e.g., the U.S. Fair Credit Reporting Act of 1970 (see 15 USC 1681 *et seq.*), require only the name of the company that requested the information. Other laws require accountability to a specific office or individual. Because the granularity of accountability may differ from jurisdiction to jurisdiction, organizations may need to develop their applications to meet the most stringent requirements, i.e., individual accountability. In my experience, few electronic data interchange (EDI) systems support this level of accountability. (*UNCID Uniform Rules of Conduct for Interchange of Trade Data by Teletransmission*, ICC Publishing Corporation, New York, 1988. All protective measures and audit measures are described as options with granularity left to the discretion of the parties.)

To further complicate data transfer issues, patent, copyright, and trade secrets laws are not uniform. Although international conventions exist, e.g., General Agreement on Tariffs and Trade (GATT), not all nations subscribe to these conventions, and the conventions often allow for substantial differences among signatories. Rights you may have and can enforce in one jurisdiction may not exist (or may not be enforceable) in another. In some cases, the rights you have in one jurisdiction constitute an infringement in another jurisdiction. For example, you may hold a U.S. registered trademark on a product. A trademark is a design (often a stylized name or monogram) showing the origin or ownership of merchandise and reserved to the owner's exclusive use. The Trade-Mark Act of 1946 (see 15 USC 1124) provides that no article shall be imported that copies or simulates a trademark registered under U.S. laws. A similar law protecting, for example, trademarks registered in India might prevent your using the trademark in India if a similar or identical trademark is already registered there.

Disclosure of information not in accordance with the laws of the jurisdictions involved may subject the parties to criminal penalties. For example, the U.K.'s Official Secrets Act of 1989 clearly defines areas wherein disclosure of the government's secrets is a criminal offense. Most nations have similar laws (of varying specificity) making the disclosure of state secrets a crime. However, technical information considered public in one jurisdiction may be considered a state secret in another. Similarly, biographical information on a national leader may be mere background information for a news story in one country, but be viewed as espionage by another. These areas are particularly difficult because most governments will not advise you in advance what constitutes a state secret (as this might compromise the secret). Unless your organization has a presence in each jurisdiction sensitive to these political and legal issues to whom you can turn for guidance, you should seek competent legal advice before transmitting text or textual database materials containing information about individuals or organizations.

From a business perspective, civil law rather than criminal law may take center stage. Although the United States probably has the dubious distinction as the nation in which it is easiest to initiate litigation, law suits are possible in most jurisdictions worldwide. No company wants to become entangled in litigation, especially in foreign jurisdictions. However, when information is transmitted from one nation to another, the rules may

change significantly. For example, what are the implied warranties in the receiving jurisdiction? What constitutes profanity, defamation, libel, or similar actionable content? What contract terms are unenforceable (e.g., can you enforce a nondisclosure agreement of 10 years' duration)?

In some jurisdictions ecclesiastical courts may have jurisdiction for offenses against a state-supported religion. Circumstances viewed in one jurisdiction as standard business practices (e.g., "gifts") may be viewed in another as unethical or illegal. Even whether an organization has standing (i.e., may be represented in court) varies among nations. An organization's rights to defend itself, for example, vary from excellent to nil in jurisdictions ranging from Canada to Iran.

Fortunately, companies may generally choose the jurisdictions in which they will hold assets. Most countries enforce their laws (and the actions of their courts) against corporations by threat of asset seizure. A company with no seizable assets (and no desire to conduct future business) in a country is effectively judgment-proof. The reverse can also be true, i.e., a company may be unable to enforce a contract (or legal judgment) because the other party has no assets within a jurisdiction willing to enforce the contract or judgment. When you contract with a company to develop software, for example, and that company exists solely in a foreign country, your organization should research the enforceability of any contract and, if you have any doubt, require a bond be posted in your jurisdiction to ensure at least bond forfeiture as recourse.

Specific and General Jurisdiction

In September 1997, in *Bensusan Restaurant Corp. v. King* (1997 U.S. App. Lexis 23742 (2d Cir., Sept. 10, 1997)), the second U.S. Circuit Court of Appeals held that a Missouri resident's Web site, accessed in New York, did not give rise to jurisdiction under New York's long-arm statute. The court ruled there was no jurisdiction because the defendant was not physically in New York when he created the offending Web page. However, a similar case in California with a similar ruling was reversed on appeal (*Hall v. LaRonde*, 1997 Cal. App. Lexis 633 (Aug. 7, 1997)). Citing the changing "role that electronic communications plays in business transactions," the court decided that jurisdiction should not be determined by whether the defendant's communications were made physically within the state, instead concluding: "[t]here is no reason why the requisite minimum contacts cannot be electronic."

To comply with due process, the exercise of specific jurisdiction generally requires that the defendant took advantage of the benefits of the jurisdiction intentionally, and so could have expected to be hauled into court there. The nature of electronic communications and their growing role in commerce have contributed to findings that defendants' Internet communications constitute "purposeful availment" (legalese for intentionally taking advantage of the benefits) and establish jurisdiction. For example, in *California Software Inc. v. Reliability Research Inc.* (631 F. Supp. 1356 (C.D. Cal. 1986)) the court held that a nonresident's defamatory e-mail to a resident was sufficient to establish specific jurisdiction. The court noted that, as modern technology makes nationwide commercial transactions more feasible, it broadens the scope of jurisdiction.

Courts have also pointed out the distinguishing features of the Internet when holding that a Web site gives rise to specific jurisdiction for infringement claims arising out of the site's content. In *Maritz Inc. v. Cybergold Inc.*, (947 F. Supp. 1328, 1332, 1334 (E.D. Mo. 1996)) the court suggested that Web site advertising more likely amounts to purposeful availment than advertising by direct mail or an "800" telephone number, noting the "different nature" of electronic communications.

Conceivably, a Web site could reflect contacts with a state's residents that were sufficiently continuous and systematic to establish general jurisdiction over the site owner. Courts have held, however, that the mere creation of a Web site does not create general jurisdiction. See, for example, *McDonough v. Fallon McElligott, Inc.*, (1996 U.S. Dist. Lexis 15139 (S.D. Cal., Aug. 6, 1996)). Further, courts have held in more traditional contexts that merely placing advertisements in nationally distributed periodicals or communicating through a national computer-based information system does not subject a nonresident to jurisdiction. See, for example, *Federal Rural Elec. Ins. Corp. v. Kootenai Elec. Corp.* (17 F.3d 1302, 1305 (10th Cir. 1994)).

This area of law is evolving rapidly, with many jurisdictions asserting what amounts to extraterritorial jurisdiction on the basis of electronic transactions into, through, or out of their territory. The Council of Europe's Convention for the Protection of Individuals with Regard to Automatic Processing of Personal Data is but one of many examples. The entire area of cryptography, for example, is another. In January 1999, the French dramatically eased their long-standing restriction on the use of cryptography within its jurisdiction. This announcement came only six weeks after France joined with 32 other countries to sign an update of a document known as the Wassenaar Agreement. Signatories to this agreement promised to tighten restrictions

on the import or export of cryptography. The so-called "long arm" provisions of many laws and the lack of consensus among nations on important issues including privacy, intellectual property rights, communications security, and taxes will challenge (or plague) us for the foreseeable future.

Technical Issues

Any nation wishing to enforce its laws with regard to data transmitted within or across its borders must have (1) the ability to monitor/intercept the data and (2) the ability to interpret/understand the data. Almost all nations can intercept wire (i.e., telephone/telegraph) communications. Most can intercept radio, microwave, and satellite transmissions. Unless your organization uses exotic technologies (e.g., point-to-point laser, extremely low frequency (ELF), super high frequency, spread spectrum), interception will remain likely.

The second requirement, however, is another matter. Even simple messages encoded in accordance with international standards may have meaning only in a specific context or template not inherent in the message itself. For example: "142667456043052" could be a phone number (e.g., 1-426-674-5604 x3052), or it could be a Social Security number and birthday (e.g., 142-66-7456 04/30/52), or it could be dollar amounts ($14,266.74 $560,430.52), or inventory counts by part number (PN) (e.g., PN 142667 Quantity 45, PN 604305 Quantity 2), or zip codes (e.g., 41266, 74560, 43052). Almost limitless possibilities exist even without using codes or ciphers. And this example used human-readable digits. Many transmissions may be graphic images, object code, or compressed text files completely unintelligible to a human "reading" the data on a datascope.

From the preceding, you might conclude that interception and interpretation by even a technologically advanced nation was too great a challenge. This is, however, far from true. Every "kind" of data has a signature or set of attributes which, when known, permits its detection and identification. This includes encrypted data where the fact of encryption is determinable. Where transmitting or receiving encrypted messages is a crime, a company using encryption risks detection. Once the "kind" of data is determined, applying the correct application is often a trivial exercise. Some examples of such strong typing of data include:

- Rich text format (RTF) documents and most word processing documents
- SQL transactions
- Spreadsheets (e.g., Lotus 1-2-3, Microsoft Excel)
- DOS, Windows, UNIX, and other operating system executables
- Standardized EDI messages
- ASCII vs. EBCDIC

If this were not the case, sending data from one computer to another would require extensive advanced planning at the receiving computer — severely impacting data portability and interoperability, two attributes widely sought in business transactions.

Countries with sufficient technology to intercept and interpret your organization's data may pose an additional problem beyond their law enforcement: government-sponsored industrial espionage. Many countries have engaged in espionage with the specific objective of obtaining technical or financial information of benefit to the countries' businesses. A search of news accounts of industrial espionage resulted in a list including the following countries: Argentina, Cuba, France, Germany, Greece, India, Iran, Iraq, Israel, Japan, North Korea, People's Republic of China, Russia, South Korea, and Turkey. Most of these countries have public policies against such espionage, and countries like the United States find it awkward to accuse allies of such activities (both because the technical means of catching them at it may be a state secret and because what one nation views as counter-espionage another nation might view as espionage!).

Protective Technologies

For most businesses, the integrity of transmitted data is more important than its privacy. Cryptographic techniques a business might otherwise be unable to use because of import or export restrictions associated with the cryptographic process or the use of a privacy-protected message may be used in some applications for data integrity. For example, the Data Encryption Standard (DES), when used for message authentication in accordance with the American National Standard X9.9 for the protection of electronic funds transfers between financial institutions, may be approved by the U.S. Department of the Treasury without having to meet the requirements of the International Trade in Arms Regulations (ITAR). (Note that technological

advances may also impact this. For example, the key space exhaustion attack in January 1999 of a DES Challenge was successful in 22.25 hours. Both the U.S. and French governments made policy changes that permit stronger cryptography for export and import than had previously been permitted.)

Integrity measures generally address one or both of the following problems:

- Unauthorized (including accidental) modification or substitution of the message
- Falsification of identity or repudiation of the message

The techniques used to address the first problem are generally called message authentication techniques. Those addressing the second class of problems are generally called digital signature techniques.

Message authentication works by applying a cryptographic algorithm to a message in such a way as to produce a resulting message authentication code (MAC), which has a very high probability of being affected by a change to any bit or bits in the message. The receiving party recalculates the MAC and compares it to the transmitted MAC. If they match, the message is considered authentic (i.e, received as sent); otherwise, the message is rejected.

Because international standards include standards for message authentication (e.g., ISO 9797), an enterprise wanting to protect the integrity of its messages can find suitable algorithms that should be (and historically have been) acceptable to most jurisdictions worldwide. With some exceptions, even the Data Encryption Algorithm (DEA), also known as the Data Encryption Standard (DES), may be used in hardware implementations of message authentication. For digital signature this may also be true, although several excellent implementations (both public key and secret key) rely on algorithms with import/export restrictions. The data protected by digital signature or message authentication, however, is not the problem, as both message authentication and digital signature leave the message in plaintext. Objections to their use center primarily on access to the cryptographic security hardware or software needed to support these services. If the cryptographic hardware or software can be obtained legally within a given jurisdiction without violating export restrictions, then using these services rarely poses any problems.

Digital signature techniques exist for both public key and secret key algorithm systems (also known respectively as asymmetric- and symmetric-key systems). The purpose of digital signature is to authenticate the sender's identity and to prevent repudiation (where an alleged sender claims not to have sent the message). The digital signature implementation may or may not also authenticate the contents of the signed message.

Privacy measures address the concern for unauthorized disclosure of a message in transit. Cipher systems, e.g., DEA, transform data into what appear to be random streams of bits. Some ciphers, e.g., a Vernam cipher with a keystream equal to or longer than the message stream, provide almost unbreakable privacy. As such, the better cipher systems almost always run afoul of export or import restrictions. In May 2002, NIST announced that the Rijndael algorithm had been selected as the AES standard, FIPS 197, to replace DES. One of the policy isssues with AES will be its exportability, as it will allow 128- and 256-bit encryption keys.

In some cases, the use of codes is practical and less likely to run into restrictions. As long as the "codebook" containing the interpretations of the codes is kept secret, an organization could send very sensitive messages without risk of disclosure if intercepted in route. For example, an oil company preparing its bid for an offshore property might arrange a set of codes as shown in Exhibit 141.1.

The message "RED SUN NOVEMBER MAY MAY" would make little sense to an eavesdropper but would tell your representative the maximum authorized bid is 900 (the units would be prearranged, so this could mean $900,000).

Other privacy techniques that do not rely on secret codes or ciphers include:

1. Continuous stream messages (the good message is hidden in a continuous stream of otherwise meaningless text). For example: "THVSTOPREAXZTRECEEBNKLLWSYAINNTHELAUNCHGBMEAZY" contains the message "STOP THE LAUNCH." When short messages are sent as part of a continuous, binary stream, this technique (one of a class known as steganography) can be effective. This technique is often combined with cipher techniques where very high levels of message security are needed.
2. Split knowledge routing (a bit pattern is sent along a route independent of another route on which a second bit pattern is sent; the two bit streams are exclusive-ORed together by the receiving party to form the original message). For example, if the bit pattern of the message you wished to send was 0011 1001 1101 0110, a random pattern of equal length would be exclusive-ORed with the message, e.g., 1001 1110 0101 0010, to make a new message 1010 0111 1000 0100. The random pattern would be sent along one telecommunication path, and the new message would be sent along another, independent telecommunications path. The recipient would exclusively OR the two messages back together, resulting

EXHIBIT 141.1 Sample Codebook.

Code	Meaning
Red Sun	Highest authorized bid is
Blue Moon	Stall, we aren't ready
White Flower	Kill the deal; we aren't interested
June	1
April	2
July	3
December	4
August	5
January	6
March	7
September	8
November	9
May	0

in the original message. Because no cryptographic key management is required and because the exclusive-OR operation is very fast, this is an attractive technique where the requirement of independent routing can be met.

3. The use of templates (which must remain secret) that permit the receiver to retrieve the important values and ignore others in the same message. For example, our string used above: "THVSTOPRE-AXZTRECEEBNKLLWSYAINNTHELAUNCHGBMEAZY" used with the following template reveals a different message: "XXXXXXXNNXXXNNNXXXXXXXXXXXXNXXXNXXXXXXXXXXXXXXX" where only the letters at the places marked with "N" are used: RETREAT.

The first technique may also be effective against traffic analysis. The second technique requires the ability to ensure independent telecommunications routes (often infeasible). The third technique has roughly the same distribution problems that codebook systems have, i.e., the templates must be delivered to the receiver in advance of the transmission and in a secure manner. These techniques do, however, avoid the import and export problems associated with cryptographic systems.

In addition to cryptographic systems, most industrialized nations restrict the export of specific technologies, including those with a direct military use (or police use) and those advanced technologies easily misused by other nations to suppress human rights, improve intelligence gathering, or counter security measures. Thus, an efficient relational database product might be restricted from export because oppressive third-world nations might use it to maintain data on their citizens (e.g., "subversive activities lists"). Restrictions on software export can sometimes be averted by finding a nation in which the desired product is sold legally without the export restriction. (*Note*: Check with your legal counsel in your enterprise's official jurisdiction as this work-around may be illegal — some countries claim extraterritorial jurisdiction, or claim that their laws take precedence for legal entities residing within their borders.) For example, the Foreign Corrupt Practices Act (see 15 USC 78) of the United States prohibits giving gifts (i.e., paying graft or bribes) by U.S. corporations even if such practice is legal and traditional in a country within which you are doing business. Similarly, if the People's Republic of China produces clones of hardware and software that violate intellectual property laws of other countries but which are not viewed by China as a punishable offense, using such a product to permit processing between the United States and China would doubtlessly be viewed by U.S. authorities as unacceptable.

The Long View

New technologies (e.g., Software Defined Digital Network (SDDN) and Frame Relay) will make our networks increasingly intelligent, capable of enforcing complex compliance rules and allowing each enterprise to carefully craft the jurisdictions from which, through which, and into which its data will flow. North America, the European Community, Japan, and similar information age countries will see these technologies soon. But many nations will not have these capabilities for decades.

Most jurisdictions will acquire the ability to detect cryptographic messages and to process cleartext messages even before they acquire the networking technologies that would honor an enterprise's routing requests. The result may be a long period of risk for those organizations determined to send and to receive whatever data they deem necessary through whatever jurisdictions happen to provide the most expeditious routing.

The use of public key infrastructures (PKI) and the reliance on certificate authorities (CA) for electronic commerce will force many changes in international law. The jurisdictional location of a registration authority (RA), for example, may dictate whose personal data may be captured for registration. In a ruling by the EC Privacy Council early in 1999 with regard to IP addresses, it was determined that a static IP address constituted privacy-protected data, just as a name and mailing address would. The existence of a CA in a jurisdiction may constitute a nexus for an assertion of general jurisdiction or for taxation if the certificates signed by this CA are used for commercial purposes. Although this technology promises solutions to many problems — including restricting access to data on a selective basis that could bound jurisdictions — it also introduces rapid change and complexity with which societies (and legal systems) are already struggling.

Summary

Data daily flows from jurisdiction to jurisdiction with most organizations unaware of the obligations they may incur. As nations become more sophisticated in detecting data traffic transiting their borders, organizations will face more effective enforcement of laws, treaties, and regulations ranging from privacy to state secrets, and from tax law to intellectual property rights. The risk of state-sponsored industrial espionage will also increase. Because organizations value the information transferred electronically, more and more will turn to cryptography to protect their information. Cryptography, however, has import and export implications in many jurisdictions worldwide. The technology required to intelligently control the routing of communications is increasingly available, but will not solve the problems in the short term. Rather, the advancing technology will complicate matters further in two ways:

1. Where the controls become available, it will make their nonuse indefensible.
2. Where the controls are used, it will make the jurisdictions intentional, thereby strengthening the state's case that it has jurisdiction.

With more legal entities asserting jurisdiction, conflict of laws cases will increase. Implicit contracts will become extremely hazardous (e.g., an e-mail message may be sufficient to constitute a contract, but what are its default terms?). Ultimately, the need for effective commerce will prevail and jurisdictional issues will be resolved. But for the near term, jurisdictional issues in global transmissions remains a growth industry for legal professionals, politicians, lobbyists, tax accountants, and electronic commerce consultants.

Companies will need to exercise care when they place their data on open networks, the routings of which they cannot control. They will need to understand the jurisdictions in which and through which their global information infrastructure operates. The information security professional will want to have competent legal assistance on the team and to stay well informed. The effectiveness of the enterprise's information security program is now irreversibly intertwined with the jurisdictional issues of global electronic commerce.

142

Liability for Lax Computer Security in DDoS Attacks

Dorsey Morrow, JD, CISSP

In the middle of February 2000, Internet security changed dramatically when Amazon.com, CNN, Yahoo! E*Trade, ZDNet, and others fell victim to what has come to be known as a distributed denial-of-service attack or, more commonly, DDoS. Although denial-of-service attacks can be found as far back as 1998, it was not until these sites were brought down through the use of distributed computing that the media spotlight focused on such attacks. No longer were the attackers few in number and relatively easy to trace. A DDoS attack occurs when a targeted system is flooded with traffic by hundreds or even thousands of coordinated computer systems simultaneously. These attacking computer systems are surreptitiously *commandeered* by a single source well in advance of the actual attack. Through the use of a well-placed Trojan program that awaits further commands from the originating computer, the attacking computer is turned into what is commonly referred to as a *zombie*. These zombie computers are then coordinated in an assault against single or multiple targets. Zombie computers are typically targeted and utilized because of their lax security. Although a DDoS attack has two victims — the attacking zombie computer and the ultimate target — it is the latter of these two that suffers the most damage. Not only has the security and performance of the victim's computer system been compromised, but economic damage can run into the millions of dollars for some companies. Thus, the question arises: does the attack by a zombie computer system, because of lax security, create liability on the part of the zombie system to the target? To address this issue, this chapter provides a jurisdictional-independent analysis of the tort of negligence and the duty that attaches upon connection to the Internet.

There is a universal caveat in tort law stating that, whenever you are out of a familiar element, a reasonable and prudent person becomes even more cautious. The Internet fits the profile of an unfamiliar element in every sense of the word, be it transactional, jurisdictional, or legal. There is no clear, concise, ecumenical standard for the Internet as it applies to business transactions, political borders, or legal jurisdictions and standards. Thus, every computer user, service provider, and business entity on the Internet should exercise extra caution in travels across the Internet. But, beyond such a general duty to be extra cautious, is there more expected of those who join the broad Internet community and become *Netizens*? Specifically, is there a duty to others online?

Computer security is a dynamic field; and in today's business and legal environments, the demands for confidentiality, integrity, and availability of computer data are increasing at fantastic rates. But at what level is computer security sufficient? For years we have looked to a 1932 case in the 2nd Circuit (see *In re T.J. Hooper*, 60 F.2d 737) that involved a tugboat caught up in a tremendous storm and was subsequently involved in an accident that resulted in the loss of property. Naturally, a lawsuit resulted; and the captain was found guilty of negligence for failing to use a device that was not industry-standard at the time, but was available nonetheless — a two-way radio. The court succinctly stated, "There are precautions so imperative that even their universal disregard will not excuse their omission." In essence, the court stated that, despite what the industry might be doing, or more precisely, failing to do, there are certain precautions we must implement to avoid disaster and

liability. What the courts look to is what the reasonable and prudent person (or member of industry) might do in such unfamiliar territory.

Because computer security is so dynamic, instead of trying to define a universal standard of what to do, the more practical method would be to attempt to define what rises to the standard of negligence. Negligence has developed into a common law standard of three elements. First, there must be some duty owed between the plaintiff and the defendant; second, there must be a breach of that duty by the defendant; third, the breach of duty is a proximate cause of damages that result. (See *City of Mobile v. Havard*, 289 Ala. 532, 268 So.2d 805, [1972]. See also *United States Liab. Ins. Co. v. Haidinger-Hayes, Inc.* [1970] 1 Cal.3d 586, 594, 463 P.2d 770.) So it seems we must first address whether there is a duty between the plaintiff (the victim of a DDoS attack) and the defendant zombie computer in such an attack.

Does being tied to the Internet impose a duty of security upon businesses? Do businesses have an implicit requirement to ensure their security is functional and that their systems will not harm others on the wild, wild Internet? It is important to remember that the theory of negligence does not make us insurers of all around us, but rather that we act as a reasonable and prudent person would in the same circumstances. We have already established that the Internet, despite being commercially viable for the past ten years, is still a new frontier. As such, it is challenging historical business and legal concepts. This, of course, creates a new paradigm of caution for the reasonable person or business. The Internet creates an unbridled connection among all who would join. It is undisputed that no one *owns* the Internet or is charged with regulating content, format, or acceptable use. However, there is a duty imposed upon all who connect and become part of the Internet. As in the physical world, we owe a duty to *do no harm* to those around us. Although the ultimate determination of *duty* lies properly within the discretion of the courts as a matter of law, there are a number of *duties* that have been routinely recognized by the courts.

Perhaps the duty from which we can draw the greatest inference is the duty of landowners to maintain their land. This general duty of maintenance, which is owed to tenants and patrons, has been held to include "the duty to take reasonable steps to secure common areas against foreseeable criminal acts of third parties that are likely to occur in the absence of such precautionary measures." (See *Frances T. v. Village Green Owners Assoc.* [1986] 42 Cal.3d 490, 499–501 [229 Cal.Rptr 456, 723 P.2d 573, 59 A.L.R.4th 447].) Similarly, in Illinois, there is no duty imposed to protect others from criminal attacks by a third party, *unless* the criminal attack was reasonably foreseeable and the parties had a "special relationship." (See *Figueroa v. Evangelical Covenant Church*, 879 F.2d 1427 [7th Cir. 1989].) And, in *Comolli v. 81 And 13 Cortland Assoc.*, ___A.D.2d _____ (3d Dept. 2001), the New York Appellate Division, quoting *Rivera v. Goldstein*, 152 A.D.2d 556, 557, stated, "There will ordinarily be no duty imposed on a defendant to prevent a third party from causing harm to another unless the intervening act which caused the plaintiff's injuries was a normal or foreseeable consequence of the situation created by the defendant's negligence." As a shop owner in a high-crime area owes a greater duty of security and safety to those who come to his shop because criminal action is more likely and reasonably foreseeable, thus a computer system tied to the Internet owes a duty of security to others tied to the Internet because of the reasonably foreseeable criminal actions of others. Similarly, if we live in an area where there have been repeated car thefts, and those stolen cars have been used to strike and assault those who walk in the area, it could be reasonably stated we have a duty to the walkers to secure our vehicles. It is reasonably foreseeable that our car would be stolen and used to injure someone if we left it in the open and accessible. The extent to which we left it accessible would determine whether we breached that duty and, pursuant to law, left to the decision of a jury. Whether it was parked in the street, unlocked, and the keys in it, or locked with an active alarm system would be factors the jury would consider in determining if we had been negligent in securing the automobile. Granted, this is a rather extreme and unlikely scenario; but it nonetheless illustrates our duty to others in the digital community.

Statistics that bolster the claim that computer crime is a reasonably foreseeable event include a study by the Computer Security Institute and the San Francisco Federal Bureau of Investigation Computer Intrusion Squad of various organizations on the issue of computer security compiled in March 2001. In their study, 85 percent of respondents detected computer security breaches within the previous 12 months; 38 percent detected DoS attacks in 2001 compared to 27 percent for 2000; and 95 percent of those surveyed detected computer viruses. These numbers clearly show a need for computer security and how reasonably foreseeable computer crime is when connected to the Internet.

When viewed in the light of increasing numbers of viruses, Trojan horses, and security breaches, and the extensive media attention given them, computer crime on the Internet almost passes beyond "reasonably foreseeable" to "expected." A case in Texas, *Dickinson Arms-Reo v. Campbell*, 4 S.W.3d 333 (Tex.App. [1st Dist.]

1999) held that the element of "'foreseeability" would require only that the general danger, not the exact sequence of events that produced the harm, be foreseeable. The court went further to identify specific factors in considering "foreseeability" to include: (1) the proximity of other crimes; (2) the recency and frequency of other crimes; (3) the similarity of other crimes; and (4) the publicity of other crimes. Although this is not a ubiquitous checklist to be used as a universal standard, it does give a good reference point with which to measure whether a computer crime could be reasonably expected and foreseeable. Of course, in cyberspace, there is no physical land, tenants, or licensees. However, there is still a duty to secure systems against unauthorized use, whether mandated by statute (Health Insurance Portability and Accountability Act, Graham-Leach-Bliley Act), by regulation, or by common sense. Because of the public nature of the recent DDoS attacks, we now have a better understanding of the synergistic and interconnected nature of the Internet and the ramifications of poor security.

Perhaps the most striking argument for the duty of precaution comes from a 1933 Mississippi case in which the court stated:

> Precaution is a duty only so far as there is reason for apprehension. Ordinary care of a reasonably prudent man does not demand that a person should prevision or anticipate an unusual, improbable, or extraordinary occurrence, though such happening is within the range of possibilities. Care or foresight as to the probable effect of an act is not to be weighed on jewelers' scales, nor calculated by the expert mind of the philosopher, from cause to effect, in all situations. Probability arises in the law of negligence when viewed from the standpoint of the judgment of a reasonably prudent man, as a reasonable thing to be expected. Remote possibilities do not constitute negligence from the judicial standpoint.
>
> ***— Illinois Central RR Co. v. Bloodworth***
> **166 Miss. 602, 145 So. 333 (1933)**

A 1962 Mississippi case (*Dr. Pepper Bottling Co. v. Bruner*, 245 Miss. 276, 148 So.2d 199) went further in stating that:

> As a general rule, it is the natural inherent duty owed by one person to his fellowmen, in his intercourse with them, to protect life and limb against peril, when it is in his power to reasonably do so. The law imposes upon every person who undertakes the performance of an act which, it is apparent, if not done carefully, will be dangerous to other persons, or the property of other persons — the duty to exercise his senses and intelligence to avoid injury, and he may be held accountable at law for an injury to person or property which is directly attributable to a breach of such duty…. Stated broadly, one who undertakes to do an act or discharge a duty by which conduct of others may be properly regulated and governed is under a duty to shape his conduct in such matter that those rightfully led to act on the faith of his performance shall not suffer loss or injury through his negligence.

We have established the requirement of a duty; but in the context of computer security, what rises to the level of a breach of such a duty? Assuming that a duty is found, a plaintiff must establish that a defendant's acts or omissions violated the applicable standard of care. We must then ask, "What is the standard of care?" According to a 1971 case from the Fifth Circuit, evidence of the custom and practice in a particular business or industry is usually admissible as to the standard of care in negligence actions. (See *Ward v. Hobart Mfg. Co.*, 460 F.2d 1176, 1185.) When a practice becomes so well defined within an industry that a reasonable person is charged with knowing that is the way it is done, a standard has been established. Although computer security is an industry unto itself, its standards vary due to environmental constraints of the industry or business within which it is used. Although both a chicken processing plant and a nuclear processing plant use computer security, the risks are of two extremes. To further skew our ability to arrive at a common standard, the courts have held that evidence of accepted customs and practices of a trade or industry does not *conclusively* establish the legal standard of care. (See *Anderson v. Malloy*, 700 F.2d 1208, 1212 [1983].) In fact, the cost justification of the custom may be considered a relevant factor by some courts, including the determination of whether the expected accident cost associated with the practice exceeded the cost of abandoning the practice. (See *United States Fidelity & Guar. Co. v. Plovidba*, 683 F.2d 1022, 1026 [7th Cir. 1982].) So if we are unable to arrive at a uniform standard of care for computer security in general, what do we look to? Clearly there must be a minimum standard for computer security with which we benchmark our duty to others on the Internet. To

arrive at that standard we must use a balancing test of utility versus risk. Such a test helps to determine whether a certain computer security measure ought to be done by weighing the risk of not doing it versus the social utility or benefit of doing it, notwithstanding the cost. In June 2001, in *Moody v. Blanchard Place*, 34,587 (La.App. 2nd Cir. 6/20/01); ___ So.2d ___, the Court of Appeals for Louisiana held that, in determining the risk and utility of doing something, there are several factors to consider: (1) a determination of whether a thing presents an unreasonable risk of harm should be made "in light of all relevant moral, economic, and social considerations" (quoting *Celestine v. Union Oil Co. of California*, 94-1868 [La. 4/10/95], 652 So.2d 1299; quoting *Entrevia v. Hood*, 427 So.2d 1146 [La. 1983]); and (2) in applying the risk–utility balancing test, the fact finder must weigh factors such as gravity and risk of harm, individual and societal rights and obligations, and the social utility involved. (Quoting *Boyle v. Board of Supervisors, Louisiana State University*, 96-1158 [La. 1/14/97], 685 So.2d 1080.) So whether to implement a security measure may be considered in light of economical and social considerations weighed against the gravity and risk of harm. This in turn works to establish the standard of care. If the defendant failed to meet this standard of care, then the duty to the plaintiff has been breached.

Finally, we must consider whether the breach of duty by the defendant to the plaintiff was the proximate cause of damages the plaintiff experienced. To arrive at such a claim, we must have damages. Over the years the courts have generally required physical harm or damages. In fact, economic loss, absent some correlating physical loss, has traditionally been unrecoverable. (See *Pennsylvania v. General Public Utilities Corp.* [1983, CA3 Pa] 710 F.2d 117.) Over the past two decades, however, the courts have been allowing for the recovery of purely economic losses. (See *People Express Airlines v. Consol. Rail Corp.*, 194 N.J. Super. 349 [1984], 476 A.2d 1256.) Thus, although the computer and Internet are not physically dangerous machines (unless attached to some other equipment that is dangerous) and thus incapable of creating a physical loss or causing physical damage, they can produce far-reaching economic damage. This is especially true as more and more of our infrastructure and financial systems are controlled by computer and attached to the Internet. Hence, we arrive at the ability to have damages as the result of action by a computer.

The final question is whether the action or inaction by the defendant to secure his computer systems is a proximate cause of the damages suffered by the plaintiff as the result of a DDoS attack by a third party. And, of course, this question is left to the jury as a matter of fact. Each case carrying its own unique set of circumstances and timelines creates issues that must be resolved by the trier of fact — the jury. However, in order to be a proximate cause, the defendant's conduct must be a cause-in-fact. In other words, if the DDoS attack would not have occurred without the defendant's conduct, it is not a cause-in-fact. Of course, in any DDoS there are a multitude of other parties who also contributed to the attack by their failure to adequately secure their systems from becoming zombies. But this does nothing to suppress the liability of the single defendant. It merely makes others suitable parties to the suit as alternatively liable. If the defendant's action was a material element and a substantial factor in bringing about the event, regardless of the liability of any other party, their conduct was still a cause-in-fact and thus a proximate cause. In 1995, an Ohio court addressed the issue of having multiple defendants for a single proximate cause, even if some of the potential defendants were not named in the suit. In *Jackson v. Glidden*, 98 Ohio App.2d 100 (1995), 647 N.E.2d 879, the court, quoting an earlier case, stated:

> In *Minnich v. Ashland Oil Co.* (1984), 15 Ohio St.3d 396, 15 OBR 511, 473 N.E.2d 1199, the Ohio Supreme Court recognized the theory of alternative liability. The court held in its syllabus:
>
> "Where the conduct of two or more actors is tortious, and it is proved that harm has been caused to the plaintiff by only one of them, but there is uncertainty as to which one has caused it, the burden is upon each such actor to prove that he has not caused the harm. (2 Restatement of the Law 2d, Torts, Section 433[B][3], adopted.)"
>
> The court stated that the shifting of the burden of proof avoids the injustice of permitting proved wrongdoers, who among them have inflicted an injury upon an innocent plaintiff, to escape liability merely because the nature of their conduct and the resulting harm have made it difficult or impossible to prove which of them have caused the harm.
>
> The court specifically held that the plaintiff must still prove (1) that two or more defendants committed tortious acts, and (2) that plaintiff was injured as a proximate result of the wrongdoing of one of the defendants. The burden then shifts to the defendants to prove that they were not the cause

of the plaintiff's injuries. The court noted that there were multiple defendants but a single proximate cause.

This case does not create a loophole for a defendant in a DDoS attack to escape liability by denying his computer security created the basis for the attack; rather, it allows the plaintiff to list all possible defendants and then require them to prove they did not contribute to the injury. If a computer system was part of the zombie attack, it is a potential party and must prove otherwise that its computer security measures met the standard of care and due diligence required to avoid such a breach.

In conclusion, we must look to the totality of circumstances in any attack to determine liability. Naturally, the ultimate responsibility lies at the feet of the instigator of the attack. It is imperative that the Internet community prosecute these nefarious and illegitimate users of computer resources to the fullest and reduce such assaults through every legitimate and legal means available. However, this does not reduce the economic damages suffered by the victim. For that, we look to "deep pockets" and their roles in the attacks. Typically, the deep pockets will be the zombies. But the true determination of their liability is in their security. We must look to the standard of care in the computer security field, in the zombie's particular industry, and the utility and risk of implementing certain security procedures that could have prevented the attack. Could this attack have been prevented or mitigated by the implementation of certain security measures, policies, or procedures? Was there a technological "silver bullet" that was available, inexpensive, and that the defendant knew or should have known about? Would a firewall or intrusion detection system have made a difference? Did the attack exploit a well-known and documented weakness that the defendant zombie should have corrected? Each of these questions will be raised and considered by a jury to arrive at the answer of liability. Each of these questions should be asked and answered by every company before such an attack even transpires.

It is highly probable that those who allow their computer systems, because of weak security, to become jumping-off points for attacks on other systems will be liable to those that are the victims of such attacks. It is incumbent upon all who wish to become part of the community that is the Internet to exercise reasonable care in such an uncertain environment. Ensuring the security of one's own computer systems inherently increases the security of all other systems on the Internet.

143

The Final HIPAA Security Rule Is Here! Now What?

Todd Fitzgerald, CISSP, CISA

We are privileged to live in a society that values freedom and the individual rights of its citizens to have the opportunity to make choices that affect their own well-being. These freedoms are exercised on a daily basis without conscious thought and are many times taken for granted. For example, people make choices about where they will eat lunch, where they will have cars repaired, who will provide care for their children, where they will spend their money, what leisure activities they will participate in, and how they will use their time. One of the most important choices individuals make is the selection of healthcare. The choice of healthcare provider, be it a doctor, a hospital, or an integrated clinical system with a network of doctors and treatment facilities, is a personal choice based on many factors such as professional competence, practice location, specialty of the medicine, and trust in the ability of the medical professional. Selection of someone to provide medical attention is no small matter to be taken lightly; being able to trust the medical professional is arguably of the utmost importance.

In a generation where access to information is literally only seconds away, this trust is not blind. The Internet is used extensively by patients or concerned family members for researching medical ailments and then suggesting treatments or questioning the physician's recommended course of action. Even though a high level of trust may be invested with the physician, individuals still feel a need to find other sources of information that corroborate the recommended treatment. Due to this phenomenon, the patient is much more informed about treatment choices, medications, and potential outcomes. The Internet has accelerated this shift, which started as "Baby Boomers," also known as the "sandwich generation," needed to care simultaneously for their children and elderly parents, in addition to being concerned with the medical effects of their own aging.

Just as patients must be able to trust their medical professionals for their treatment, patients trust that they are using the medical health information, their personal medical health information, solely for the purposes of treatment, payment, or operations. They also trust that this information is kept private and that appropriate measures are taken to ensure that the information is not inadvertently disclosed, destroyed, or changed in a way that could adversely affect their treatment or create personal embarrassment. However, analogous to the trust that is placed in the medical professional, much more information is available today about privacy issues; thus people are also much more informed. The media has communicated countless examples such as hackers disclosing personal medical information by posting on the Internet, company e-mails inadvertently revealing patients using a particular medication, being solicited through someone having knowledge of personal medical history, or disclosure within an organization of psychological notes of other employees. People expect that their confidentiality will be maintained and the trust relationship between patient and provider is not compromised. Privacy issues address the rights of the individual with respect to this trust relationship, whereas security is the mechanism that ensures that this privacy is reasonably maintained throughout the system. True privacy of information cannot be achieved without adequate security controls. The Health Insurance Portability and Accountability Act (HIPAA) has several objectives, one of which is to ensure the appropriate security safeguards are in place to protect the privacy of health information.

HIPAA Arrives on the Scene

The Health Insurance Portability and Accountability Act (HIPAA) of 1996 was enacted by Congress (Public Law 104-191) with two purposes in mind: (1) to reform health insurance to protect insurance coverage for workers and their families when they changed or lost their jobs, and (2) to simplify the administrative processes by adopting standards to improve the efficiency and effectiveness of the nation's healthcare system. Title I of HIPAA contains provisions to address health insurance reform. Title II addresses national standards for electronic transactions, unique health identifiers, privacy, and security. Title II is known as Administrative Simplification and is intended to reduce the costs of healthcare through the widespread use of electronic data interchange. Administrative Simplification was added to Title XI of the Social Security Act through subtitle F of Title II of the enacted HIPAA law.

Although the initial intent of Administrative Simplification was to reduce the administrative costs associated with processing healthcare transactions, Congress recognized that standardizing and electronically aggregating healthcare information would increase the risk of disclosure of confidential information, and the patient's privacy rights needed to be protected. Security provisions were needed not only to protect the confidentiality of information, but also to ensure that information retained the appropriate integrity. Consider the situation where the diagnosis or vital sign information is changed on a medical record, and subsequent treatment decisions are based on this information. The impact of not being able to rely on the information stored within the healthcare environment could have life-threatening consequences. Thus, privacy issues are primarily centered on the confidentiality of information to ensure that only the appropriate individuals have access to the information, whereas the security standards take on a larger scope to address issues of integrity and availability of information.

The Rule-Making Process

Each provision of Administrative Simplification must follow a rule-making process that is designed to achieve consensus within the Department of Health and Human Services (HHS) and other federal departments. When the rule is approved within the government, the public has the opportunity to comment on the proposal, and then these comments are evaluated in the determination of the final rule. Once the rules have gone through this process, they have the force of federal law. The Department of Health and Human Services implementation teams draft Notices of Proposed Rule Making (NPRMs), which are subsequently published in the Federal Register after being reviewed within the federal government, according to the process shown in Exhibit 143.1. Once the NPRMs are published, they are available for a 60-day public comment period, which provides for input and for interested parties to influence the outcome of the final regulation. After the publication of the final rule, most large health plans, clearinghouses, and providers have 24 months to be in compliance, and smaller parties have 36 months.

The proposed security and electronic signature standards were originally published in the Federal Register on August 12, 1998. The Security Rule has been delayed on several occasions, as resources were committed to and focused on the proposed transaction and code set and Privacy Rules, both of which generated a large number of public comments. The number of public comments can be large, and each one must be reviewed. Over 17,000 public comments were received on the Transaction and Code Sets NPRM and several thousand on the Privacy Rule and on the proposed Security Rule. The transaction and code set compliance date was also delayed by one year, to October 16, 2003, as long as the covered entity filed an extension request by October

EXHIBIT 143.1 Administrative Simplification Rule-Making Process

1. HHS implementation team drafts Notice of Proposed Rule-Making (NPRM) for review
2. HHS Data Council Committee on Health Data Standards reviews
3. Advisors to HHS Secretary (division agency heads) agree
4. Office of Management and Budget (OMB) reviews
5. Proposed NPRM published in Federal Register
6. Public comments are solicited for 60-day period
7. Comments open for public view
8. Comments are analyzed and content summarized by implementation team
9. Final rule is published, standards become effective 24 months after adoption, 36 months for small health plans

15, 2002. Additionally, the Security Rule was initiated during the Clinton administration and was carried over into the Bush administration, which created political challenges for expedient passage of the rule. As a result, the language was rewritten during 2002 to coincide with the Privacy Rule, which needed to go through the HHS clearance process prior to final rule publication. During 2002, the Centers for Medicare and Medicaid Services several times provided their best estimates of publication of the final rule, which passed through the clearance process and was submitted to the Office of Management and Budget (OMB) in early 2003 and was published in the Federal Register as 45 CFR Parts 160, 162, and 164 on February 20, 2003. The regulations became effective on April 21, 2003, and covered entities must comply with the requirements by April 21, 2005. Small health plans have until April 21, 2006, to comply with the rule.

The Security Objectives of the Final Rule Did Not Change Substantially

Many organizations had been "waiting" for the final rule to be published before seriously embarking on security issues. Some started HIPAA security gap analysis efforts, but many were reluctant to invest large sums of money when there was the potential that the rules might change. The reality is that the rule embodies security practices that should be performed during the normal course of business to protect the information assets and should be initiated regardless of the rule. Waiting only shortens the time available to dedicate to reasonable security and can also have the negative effect of driving up costs at a later date. For example, if a new Web-based application is in the process of being designed and adequate attention to security is not taking place during early phases of the system development cycle, the costs of retrofitting security after implementation will be 10 to 20 times the cost. Reanalysis, rewriting of the applications, integrating technical security mechanisms, and retesting and implementing the system a second time all drive up the cost. There is also the business opportunity cost of deploying scarce information technology and business resources toward retrofitting the application vs. building new functionality.

Many of the security constructs remained in the rule, as these constructs are generally industry security practices necessary to secure information that have been applied successfully in other arenas requiring higher levels of security, such as the Department of Defense, financial institutions, and companies heavily engaged in E-commerce. The final HIPAA Security Rule recognizes the need to protect electronic health information with the appropriate administrative, physical, and technical safeguards that have been applied to other industries.

The final Security Rule was reoriented to support the final Privacy Rule, which was issued on December 28, 2000, and was last modified on August 14, 2002. The Privacy Rule compliance date for most covered entities was April 14, 2003. The proposed Security Rule focused on information maintained or transmitted by a covered entity in electronic form. The scope of the information now covered by the final Security Rule has been narrowed to health information addressed by the Privacy Rule. The Privacy Rule addresses individually identifiable health information known as protected health information (PHI) in all forms, including electronic and paper. The final Security Rule focuses only on the PHI that is in electronic form (e-PHI), in transit or in storage (data at rest); otherwise, the scope is the same as the Privacy Rule. This eliminates some of the confusion surrounding what information needed to be addressed by the Security Rule, which seemed to be in conflict with the Privacy Rule in the Security Rule NPRM.

In addition to the reorientation with the Privacy Rule, the final Security Rule changed the nomenclature of the "requirements" and "implementation features" and replaced these with "standards" and "implementation specifications," respectively. The implementation specifications were also categorized as "required" or "addressable." This was done to provide consistency with the Privacy Rule and the Transactions Rule and provide common terminology. The new approach is much cleaner, manageable, and easier to interpret. In making this change, the original 69 implementation features were reduced to 14 required implementation specifications to support the requirements, now referred to as the Security Standards.

There also appeared to be a change from a proscriptive approach to one that requires a covered entity to look at the risks and vulnerabilities to the protected health information that it transmits or maintains in electronic form and determine the reasonable and appropriate security measures to provide adequate protection of this information. The Administrative Simplification revisions to the Social Security Act required that that Secretary of HHS adopt standards that consider:

1. The technical capabilities of the record systems used to maintain the information
2. Costs of the security measures

3. Training needs for those who have access to health information
4. The value of audit trails in computerized record systems
5. The needs and capabilities of small health and rural health providers

Whereas these requirements apply to the broader topic of "health information," the final Security Rule has taken this approach with respect to electronic protected health information. Therefore, each organization must make the judgments as to what is "reasonable and appropriate" based on its size, complexity of systems, capabilities, cost of security measures, and probability and criticality of potential risks to e-PHI. Larger organizations are expected to provide more resources and have the financial ability to introduce more complex solutions.

Approximately 2350 comments were received on the initial Security Rule. These comments were assessed and taken into account with keeping the underlying goals of information protection in mind. Some of the proposed implementation specification changes were seen as resulting in standards that would be too difficult to understand or apply. Some comments proposed the expansion of applicability to all entities involved in healthcare, others sought clarification of their particular entity's requirements. Some comments demonstrated the confusion with understanding the requirements, or felt that the requirements were too granular or restrictive, or that the definitions needed further explanation. These comments were reviewed and considered in the final rule, with HHS providing changes to the rule based on industry practices, government regulations, and its mandate to produce a set of security standards.

Privacy Rule Requirements for Security

Even in the absence of the final Security Rule being available for most of the period that organizations were addressing Privacy Rule issues, the references in the Privacy Rule, which was originally published for public comment on November 11, 1999, and subsequently issued with a compliance date of April 14, 2003, as shown in Exhibit 143.2, clearly indicated the need for a reasonable level of security practices to be in place. The safeguard standard contained within §164.530 of the Privacy Rule states:

> A covered entity must have in place appropriate administrative, technical, and physical safeguards to protect the privacy of protected health information.

This appears to suggest a linkage to the Security Rule requirements, which have a compliance date much further out (at least two years) from the compliance date of the Privacy Rule!

The implementation specification for safeguards in the final Privacy Rule continues this thought, by stating:

> A covered entity must reasonably safeguard protected health information from any intentional or unintentional use or disclosure that is in violation of the standards, implementation specifications or other requirements of this subpart…. A covered entity must reasonably safeguard protected health

EXHIBIT 143.2 Notice of Proposed Rule-Making (NPRM) Dates

Proposed Rule	NPRM Date	Final Date	Compliance Date
Transaction and code sets	5/07/1998	8/17/2000[a]	10/16/2003[a]
Privacy	11/11/1999	March 2001[b]	4/14/2003[c]
Security	8/12/1998	2/20/2003	4/21/2005[d]
Employer ID	6/16/1998	3/31/2002	7/30/2004[e]

[a] Compliance date for Transaction and Code Sets was extended through legislation enacted on December 27, 2001, titled the Administrative Simplification Compliance Act, as long as providers submitted a request for extension by October 15, 2002. Modifications were made February 20, 2003, and corrected on March 10, 2003.

[b] Privacy rule changes were proposed March 27, 2002, and the final rule published August 14, 2002; however, the compliance date was not changed from the original date. Guidance was previously issued on July 6, 2001.

[c] Small health plans must be compliant by April 14, 2004.

[d] Small health plans must be compliant by April 21, 2006.

[e] Small health plans must be compliant by August 1, 2005.

> information to limit incidental uses or disclosures made pursuant to an otherwise permitted or required use or disclosure.

It is clear from these excerpts that "reasonable" security is expected to be implemented for the Privacy Rule to protect the privacy of health information. Moreover, the proposed Security Rule only applies to electronic information, whereas the Privacy Rule applies to all forms of protected health information. This creates a situation where the Privacy Rule assumes broader application in the form of protected information being addressed than the proposed Security Rule.

The Final HIPAA Security Rule

The Administrative Simplification (Part C of Title XI of the Social Security Act) provisions state that covered entities that maintain or transmit health information are required to:

> …maintain reasonable and appropriate administrative, physical, and technical safeguards to ensure the integrity and confidentiality of the information and to protect against any reasonable anticipated threats or hazards to the security or integrity of the information and unauthorized use or disclosure of the information.

Because the final Security Rule was written to be consistent with the Privacy Rule, the focus of security standards applied to "health information" in support of the Administrative Simplification requirements were shifted to PHI and specifically to e-PHI. The applicability statement of the final Security Rule states:

> A covered entity must comply with the applicable standards, implementation specifications, and requirements of this subpart with respect to electronic protected health information.

Covered entities are defined as (1) a health plan, (2) a healthcare clearinghouse, and (3) a healthcare provider who transmits any health information in electronic form in connection with a transaction covered by Part 162 of Title 45 of the Code of Federal Regulations (CFR).

This is where the security standards become important. According to the Security Rule, these standards were written to "define the administrative, physical, and technical safeguards to protect the confidentiality, integrity, and availability of electronic protected health information." Therefore, by applying the security standards on electronic PHI as the scope, the objectives of Administrative Simplification will be satisfied. All of the security standards must be satisfied, some through required implementation specifications and some through addressable implementation specifications.

As shown in Exhibit 143.3, protecting the confidentiality, integrity, and availability of electronic protected health information is at the core of the security requirements, while reasonably anticipated threats (security), uses, and disclosures (privacy) must also be protected and compliance of the workforce with the security standards ensured.

Let's Just Be Reasonable

The definition of "reasonable" can vary from person to person. The final assessment appears to be headed for the courts and will be determined by case law as a result of lawsuits. Consider the case where an employer has installed a proximity card reader for 500 employees at a data center containing protected health information. Assume the facility has a guard during the daytime; however, during the evening hours the computer operators watch the surveillance cameras for suspicious activity. One evening, while the night operator went to the restroom, someone using an unreturned visitor's badge obtained during the day entered the building and removed three laptops. Were reasonable steps taken to prevent the theft? Was the fact that the operator left his station unattended unreasonable? Was it unreasonable that the unreturned visitor's badge still worked? Or, would a jury view this situation as one that could be reasonably expected to occur? Consider another example where patient information is discovered after a Web server is hacked. If correct firewall configurations were set 99 percent of the time, except for one instance where the network engineer was upgrading the server and inadvertently opened some ports after a long, tiring weekend, was the information not reasonably protected? Is "most of the time" reasonable?

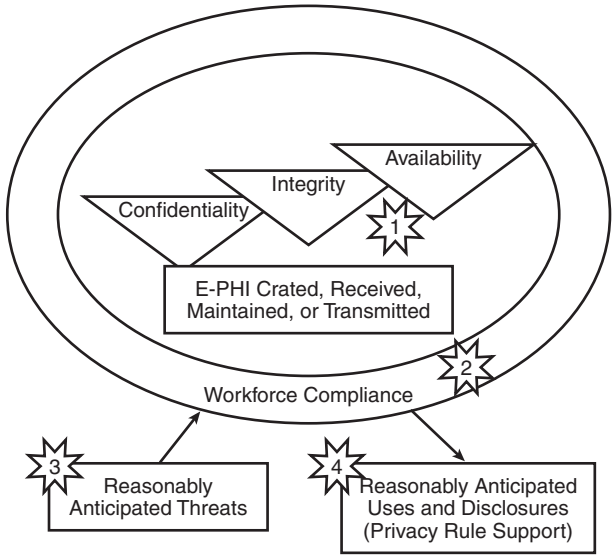


EXHIBIT 143.3 Security Rule general requirements.

Different organizations make different security decisions based on the risk that they are willing to assume. Organizations take into account the costs, technical abilities, and the risk that they are willing to assume based on their business objectives. It is critical that companies assess the threats, vulnerabilities, and risks to electronic information and develop reasonable steps to address the risk. Each of these decisions and their rationale should be documented so that it can be understood at a later point in time why the decision was made. Documenting these decisions also forces the organization to really look at the decisions that are being made and whether or not they make sense. It is not uncommon to go through this process, only to find out that management team members were making different assumptions as to the level of risk and were accepting an unreasonable level of risk without being aware that they were.

The Security Standards

The 1998 proposed Security Rule defined standards for the security of individual health information under the control of the covered entities (health plans, clearinghouses, and healthcare providers). The three safeguard categories of Administrative, Physical, and Technical contain a total of 18 security standards (vs. 24 requirements in the proposed rule) that must be addressed, as shown in Exhibit 143.4. The standards are intended to be technology neutral so that advances in technology can be used to the best advantage as they evolve.

In support of the security standards, there are 14 required implementation specifications that address seven of the eighteen security standards, as some security standards are comprised of multiple required implementation specifications. For example, the Security Management Process security standard contains four required implementation specifications, including Risk Analysis, Risk Management, Sanction Policy, and Information System Activity Review.

The covered entity must decide, through executing the risk analysis, risk mitigation strategy, cost of implementation, and evaluating the security measures that are already in place, whether or not the "addressable implementation specification" is reasonable and appropriate and should be implemented. If the specification is viewed as not reasonable and appropriate, but for the standard to be met another security safeguard is necessary to be implemented, the entity may implement the safeguard using an alternative control as long as it accomplishes the same result as the addressable implementation specification. In other words, an organization could select other controls as long as the security standard is met. In this case, the organization must document the decision not to implement the addressable implementation specification, the rationale behind it, and the alternative control that was implemented in its place. There are 22 addressable implementation specifications,

which address nine of the Security Standards; four of the Security Standards also contain required implementation specifications as well.

The six remaining Security Standards contain neither an addressable implementation specification nor a required implementation specification. In these cases it was felt that the definition of the standard itself was sufficient to understand the implementation required. For example, the Assigned Security Responsibility Standard is "identify the security official who is responsible for the development and implementation of the policies and procedures required by this subpart [Security Rule] for the entity." Additional explanation is really not necessary to understand the standard; someone needs to be designated to fulfill this role to satisfy the standard.

To "meet the Security Standards," 36 required or addressable implementation specifications must be reviewed and complied with, either through the required implementation specification, the prescribed (addressable) implementation specification, or an alternative control; combined with six Security Standards without any implementation specification noted totals 42 areas that are required to be acted upon in some manner. Although some of these tasks can be completed quickly depending on the current security profile of the organization, this still represents a significant undertaking, requiring about two of these areas to be evaluated each and every month from now until the compliance date! If someone is not "charged with the security responsibility," this would be a great time to satisfy the Assigned Security Responsibility Security Standard and draft someone. In many organizations, the need was recognized and the positions filled during the attention to the Privacy Rule due to the requirements to "have in place appropriate administrative, technical, and physical safeguards to protect the privacy of protected health information."

Changes to the Proposed Standards in the Final Rule

The following is a brief summary of the intent of each Security Standard, along with the changes from the proposed Security Rule. Each of the security standards descriptions contains references to addressable or required implementation specifications. The reader is referred to Exhibit 143.4 for the specific implementation specification designation (required or addressable).

Administrative Safeguards

Administrative safeguards consist of the formal organizational practices that manage the selection and execution of security measures to protect data and the conduct of personnel in the protection of the data. It is important that these practices are documented in the form of policies, procedures, standards, or guidelines that are followed by the organization. Although there may be accepted practices that are followed within the organization, without proper documentation it is difficult to demonstrate that all employees are working with the same assumptions. Additionally, without the documented procedures new employees may not be adequately informed as to their security responsibilities.

Much of the detail of this section was removed and the requirements were generalized to be less proscriptive. The order of the previous requirements (alphabetical) was rearranged to be more logical, with the establishment of the security management process occurring first, as everything else within the security program should be built on this. The previous requirements for system configurations and for a formal mechanism for processing records were dropped from the final rule as they were seen as redundant, unnecessary, or ambiguous with other requirements for documentation and processes.

Security Management Process

Conduct risk analysis to assess vulnerabilities and risks to the confidentiality, integrity, and availability of e-PHI, risk management of the implemented security measures, apply appropriate sanctions to workforce members who fail to comply with the security policies and procedures, and implement procedures to regularly review records of information system activity (i.e., audit logs, access reports, security incident tracking reports). The specification of "Internal Audit" was changed from the proposed rule, as it was not intended to have a rigid, costly review process over the system activity related to security, but rather to ensure that appropriate attention to security continues to take place over time. Sanction policies were seen as necessary to meet the requirement of "ensuring" compliance of the officers and employees and that the introduction of negative consequences for noncompliance increases the chances that compliance will be achieved. It is a typical result

EXHIBIT 143.4 HIPAA Security Standards and Implementation Specifications

Security Standard	Required Implementation Specification	Addressable Implementation Specification
Administrative Safeguards		
Security management process	Risk analysis Risk management Sanction policy Information system activity review	
Assigned security responsibility	Required (no implementation specification)	
Workforce security		Authorization and supervision Workforce clearance procedure Termination procedures
Information access management	Isolating healthcare clearinghouse function	Access authorization Access establishment and modification
Security awareness and training		Security reminders Protection from malicious software Log-in monitoring Password management
Security incident procedures	Response and reporting	
Contingency plan	Data backup plan Disaster recovery plan Emergency mode operation plan	Testing and revision procedure Applications and data criticality analysis
Evaluation	Required (no implementation specification)	
Business associate contracts and other arrangements	Written contract or other arrangement	
Physical Safeguards		
Facility access controls		Contingency operations Facility security plan Access control and validation procedures Maintenance records
Workstation use	Required (no implementation specification)	
Workstation security	Required (no implementation specification)	
Device and media controls	Disposal Media reuse	Accountability Data backup and storage
Technical Safeguards		
Access control	Unique user identification Emergency access procedure	Automatic log-off Encryption and decryption
Audit controls	Required (no implementation specification)	
Integrity		Mechanism to authenticate electronic protected health information
Person or entity authentication	Required (no implementation specification)	
Transmission security		Integrity controls Encryption

that if employees know that something is being monitored and followed, they are less likely to be in noncompliance with the expectations.

Security management is an ongoing function with a continuous cycle of risk analysis, risk management, and issuance of security policies and their sanctions. Over time, attention to security within organizations tends to dissipate, which (unknowingly) increases the risk profile.

Assigned Security Responsibility

An individual must be identified who is responsible for the development and implementation of security policies and procedures. Many individuals may be involved in security for the organization, but there must be one individual named with the responsibility of protecting e-PHI. The proposed rule indicated an individual or organization could be named; however, this is no longer the case; it must be a single official. Multiple people are typically involved in the security function in larger organizations; however, someone must be named with accountability for the function to ensure that policies and procedures are developed and implemented as required by the rule. The individual and supporting organization utilize the security management processes to carry out the mission of the information security program.

Workforce Security

Implement policies and procedures to ensure that every member of the workforce has appropriate access to e-PHI and prevent those who should not have access through authorization/supervision of workforce members, clearance procedures, and termination procedures. The specifications are all addressable because it will vary by organization as to whether or not they need to be formalized. Background checks are not required for all employees through the clearance specification; however, some form of screening needs to take place prior to permitting access to e-PHI. The detailed requirements of the termination procedures have been removed, again to be less proscriptive and allow flexibility for the specific environments. The intent is to ensure that when individuals with access to e-PHI are no longer associated with the entity, the exposure for potential damage is mitigated by removing their access. Small offices would most likely not require the formalized procedures that large organizations would require to meet the standard.

Information Access Management

Implement policies and procedures for establishing, authorizing, reviewing, documenting, and modifying a user's right to access a workstation, transaction, program, process, or other means of accessing e-PHI. This forms the basis of acceptable information security access management practices through (1) authorizing appropriate access, and then (2) establishing the access. This standard supports the minimum necessary requirements of the Privacy Rule, and as such, specific references to "role-based," "user-based," "context-based," discretionary/mandatory access control, and the distinctions between authorization and access control were omitted from the final rule. An added required implementation specification to isolate the clearinghouse functions from the larger organization through their own policies and procedures was added to this requirement.

Security Awareness and Training

Implement a security awareness and training program for all members of the workforce, including management, training on protection from malicious software (viruses, Trojan horses, worms, scripting, etc.), log-in monitoring, password management, and periodic security reminders. The end users are the key to successful security, and each member of the workforce must receive ongoing training. Flexibility is left up to the organization as to how this can be implemented through techniques such as face-to-face, pamphlets, new employee orientation, Web-based, etc. Many security practitioners feel that security awareness and training are the most effective areas to invest in security. These individuals represent the "security front line" and education here causes individuals to support the security program through awareness and preventing larger security issues. It does little good to implement a complex technical solution, such as implementing dynamic passwords utilizing RADIUS or TACACS+ authentication and token cards, if the user tapes a PIN to the back of the token card. Similarly, having policies that deal with the handling of confidential information would be ineffective if the users were not aware of the types of information considered confidential and needed extra measures to provide adequate protection. Training is a continual process that should focus on different aspects of information security.

Security Incident Procedures

Incident response and reporting procedures are required to mitigate the potential harmful effects of the incident and provide documentation of the incident and outcome. An incident is defined as the attempted or successful unauthorized access, use, disclosure, modification, or destruction of information or interference with system operations. Each organization must define what event would be considered a security incident and the internal/external reporting processes necessary to support the incident.

Formal, current, accurate, and documented procedures for the reporting and response to security incidents are necessary to ensure that violations are reported and handled promptly. Seemingly small incidents may be symptomatic of a larger problem and should be thoroughly investigated. Lack of attention to the small incidents also creates a culture that is desensitized to information security and creates a greater risk that a larger risk may occur. For example, if attention is not paid to the occasional laptop that is missing every few months because the information stored on the laptop was not seen as valuable, then the larger problem of laptop security awareness and the need for locking devices may be missed. Subsequently, a nurse's laptop containing health information or an executive's laptop containing confidential business strategic information may be compromised when it could have been prevented.

Contingency Plan

In the aftermath of September 11, 2001, many organizations have increased their focus toward disaster recovery. The contingency plan provides the organizational readiness to respond to systems emergencies so that critical operations can be continued during an emergency. To meet the requirement, applications and data criticality analysis, data backup plans, disaster recovery plans, emergency-mode operation plans, and testing and revision procedures are included as required or addressable implementation specifications to support the Contingency Plan standard. Most large organizations have disaster recovery plans covering the mainframe environments as a result of the Y2K contingency planning that had previously taken place. However, infrastructure and staffing are constantly changing, and as a result, many of these plans need to be updated. Although most organizations tend to back up network environments on a regular schedule, these environments rarely have adequate disaster recovery plans or are tested on a regular basis. With the continuing shift to the network/server environment for mission-critical applications, increased attention will need to be paid to the contingency planning of these facilities.

Policies and procedures are to be implemented for responding to emergencies or other occurrences (i.e., fire, vandalism, system failure, natural disaster) that damage systems that contain e-PHI. This is accomplished through data backups, disaster recovery plans, emergency mode operation plans (ability to continue business during the crisis), testing and revision procedures, and applications and data criticality analysis. This standard was proposed and remained in the final rule as data becomes most vulnerable during crisis events because security controls are typically bypassed to bring the systems back into operation. e-PHI lost during these events impacts the availability and integrity of the information, exposing the data to confidentiality issues of improper use and disclosure.

Evaluation

Perform a periodic technical and nontechnical evaluation based on the standards initially and also after environmental and operational changes affecting e-PHI. This evaluation can be performed internally or externally and replaces the certification requirement of the earlier rule. It can be expected that independent certification guides, secure software listings, and compliance guidelines will emerge from private enterprise. To form a meaningful evaluation, the risk-level acceptance of the organization should be understood prior to the evaluation, as the security measures chosen should be a result of the risk assessment decisions.

Evaluation processes, whether performed internally or externally, have the positive impact of documenting the security actions taken and obtaining management sign-off, which tends to create greater accountability beyond the security department for the implementation. It also tends to ensure that the agreed-upon security parameters in the design process are carried through to implementation.

Business Associates and Other Arrangements

The final rule eliminated the chain-of-trust agreement and replaced it with the requirement for a covered entity to ensure that appropriate safeguards are assured by the business associate through inclusion of security requirements in written contracts. The scope is limited to e-PHI, as is the rest of the Security Rule. The business associate definitions are those that are utilized within the Privacy Rule. In the event that a covered entity is aware of a pattern or practice that the business associate is engaged in that is considered a violation of the business associate's obligation, the covered entity would be in noncompliance if it failed to take reasonable steps to end the violation. Other arrangements specify situations such as how the rules apply to government entities, other laws, terminations of contracts, etc.

Physical Safeguards

Protecting the covered entity's electronic information systems and the buildings that contain these systems from fire, natural and environmental hazards, and unauthorized intrusion are the focus of the Physical Safeguards. These controls support many of the administrative and technical controls defined in the other safeguard sections. Consider the situation where very tight logical access controls (Technical Safeguard Access Control Security Standard) are defined to support the Administrative Safeguard Standards For Information Access. Assume that a computer containing these controls is located in an area where other building tenants have unrestricted access. Even with two-factor authentication, encryption of files, and properly implemented access control facilities, if the physical server can be accessed, an alternate operating system could be loaded, or worse yet, the server could be stolen, thus providing the intruder with ample opportunity to decipher encrypted files. Unauthorized employees having physical access to the server creates unnecessary additional risk.

Two requirements of the proposed rule, assigned security responsibility and security awareness training, made much more sense in the Administrative Safeguards section, and they were moved to that section in the final Security Rule. Following is a discussion of the Physical Safeguard Standards and related implementation specifications.

Facility Access Controls

Focus on the facilities that provide physical access to the electronic information systems, the standard limits physical access through contingency operations (facility access in the event of an emergency), facility security plan (safeguard facility from unauthorized physical access, tampering, and theft), access control and validation procedures (validate access to facilities based on role, control access to programs for testing and revision), and maintenance records (document repairs to facility security components). The standards appear to be straightforward and permit the organization to review the risks and implement the appropriate controls, unlike the proposed rule, which appeared to require all of the implementation specifications without regard to the risk analysis. It is still the covered entity's responsibility to ensure the facilities where e-PHI is located and transmitted are secured properly, whether or not the facility is owned by the covered entity.

Workstation Use

For workstations that are allowed access to PHI, implement policies and procedures specifying proper functions and the manner in which those functions are to be performed (i.e., locking workstations, logging off, invoking screensavers) and the physical attributes of the space surrounding the workstations. The workstation terminology is used to replace "terminal" and applies to the broad range of computing equipment with access to e-PHI (laptops, desktops, personal digital assistants, etc.) and is not limited to the desktop PC.

Workstation Security

Implement physical safeguards for all workstations that access e-PHI, restricting access to authorized users, consistent with proposed rule. Contents displayed on the workstation, especially those in open areas such as nurses' stations, must be secured so that private information is not viewable by unauthorized persons. Workstations should also be secured so that only authorized personnel would have access to the workstation. In practice, some workstations need to be in open areas and approaches such as turning the monitor away from public viewing, logging off the workstation when unattended, utilizing screensavers, and ensuring that the workstation is protected from theft would appear to be reasonable.

Device and Media Controls

Implement policies and procedures governing the receipt and removal of hardware and software in and out of a facility and movement within a facility through disposal procedures, media reuse procedures, accountability (record of movements), and data backup and storage (in this case, this is related to the backup of e-PHI prior to the moving of equipment). Media reuse procedures were added to the rule to address reuse and recycling. There have been news stories of hard drives purchased on E-Bay that contained sensitive information, which subsequently was retrieved because it was not properly disposed of after final disposition.

Technical Safeguards

The technical security services (processes that protect, control, and monitor information access), and the technical security mechanisms (processes that prevent unauthorized access over a communications network)

have been combined into the Technical Safeguards category. This is very logical, as many organizations viewed these as technical requirements. Data authentication was renamed to the standard security terminology of integrity. Following is a discussion of the Technical Safeguard Standards and related implementation specifications.

Access Control

Implement technical policies and procedures for electronic information systems containing e-PHI to allow access only to those persons who have been granted access through the Information Access Management processes in the Administrative Safeguards. Unique user IDs are necessary to identify and track the individual's activity, emergency access procedures are required to support operations in an emergency situation, and implementation specifications of automatic log-off and the use of encryption can support the access security standards. There was much confusion around the requirements for role-based, context-based, mandatory access control, discretionary access control, etc., in the proposed rule. The new specification is much cleaner and affords the organization the ability to implement the appropriate rules based on the risk. For example, an organization may decide to encrypt highly sensitive e-PHI data-at-rest if there is an assessment that this information could be compromised, such as in the case of fraud investigation involving health information stored on a CD.

A procedure for emergency access during a crisis must be implemented. Consider the situation where a specialist is called in to perform an emergency procedure, but does not have access to needed health information from the local information system. The specialist needs a method to gain emergency access without "waiting for forms to be processed" by the security department.

Audit Controls

Recording system activity is important so that the organization can identify suspect data access activities, assess the effectiveness of the security program, and respond to potential weaknesses. Implementation of hardware, software, and procedural mechanisms that record and examine activity in information systems containing e-PHI is necessary. Some organizations have assumed that the audit trails specified under this requirement would support the Privacy Rule. Typically, the types of information dealt with are different. Whereas the Privacy Rule is concerned with tracking of uses and disclosures, the Security Rule is concerned with tracking system activity, such as log-in attempts, access, and modification to records. Although similar, audit trails within the system context are typically not geared toward tracking the business-level information surrounding the use and disclosures, even though some records may provide supporting information. System audit trails are also not typically turned on to monitor read access to information due to the volume of information.

Integrity (Formerly Data Authentication)

Implement policies and procedures to protect e-PHI from improper alteration or destruction through mechanisms that corroborate that the information has not been destroyed in an unauthorized manner. Techniques such as digital signatures, checksums, and error-correcting memory are all methods of ensuring data integrity. Again, the ability to assess risk, provide technology neutrality, and not be proscriptive enables the covered entity to determine the appropriate methods to ensure the integrity of the data.

Person or Entity Authentication (Combined Authentication Requirements)

Implement procedures to verify that the person or entity seeking access to e-PHI is really the person or entity. The proposed rule was very proscriptive in suggesting biometrics, passwords, telephone callbacks, token systems, PINS, etc., where the rule now allows the implementation to be determined by the entity based on the risk assessment. The requirements for "irrefutable" entity authentication were removed in the final rule.

Transmission Security

Implement technical security guarding against unauthorized access to e-PHI transmitted over an electronic communications network (vs. open network in the proposed rule). Integrity controls and encryption may be applied according to the risk level of the information. In cases such as dial-up lines or over a private network, encryption may not be necessary to achieve the standard's objectives; however, over the Internet the appropriate encryption levels to thwart brute-force cracking may be necessary. This is an area where technology is constantly changing, there are interoperability issues, and the feasibility of solutions may make this prohibitive for small providers.

Documentation and Other Related Standards

To comply with the standards, implementation specifications, and any other requirements of the Security Rule, the covered entity must implement reasonable and appropriate policies and procedures in addition to the security standards specified in the Administrative, Technical, and Physical Safeguards. These must be documented and can be changed at any time. The covered entity can take into consideration the size, complexity, and capabilities of the covered entity; the technical infrastructure, hardware and software security capabilities, the costs of the security measures, and the probability and criticality of potential risks to the e-PHI.

Documentation of policies and procedures may seem to be such a logical practice that it may appear unnecessary to state it. However, many organizations operate without defined policies and procedures, and the work gets done. The difficulty is that many times it is done several different ways, depending on the individual performing the activity. This increases the likelihood that inconsistencies will occur, increasing the potential for security incidents. Although the Security Rule does not specify a requirement to adopt ISO 9000-type standard processes, implementing procedures that follow this approach would further support that a "reasonable" approach was taken. This also permits the opportunity to review and discuss the processes across organizations and work toward improving the processes, thus increasing service delivery capabilities and reducing waste.

Consider as an example the practice of security configuration management changes. Security measures, practices, and procedures need to be documented and integrated with the other system configuration practices to ensure that routine changes to system hardware and software do not contribute to compromising the overall security. Security design efforts placed into new systems could easily be compromised and resources wasted without the appropriate level of security review for what appears to be a simple change. Security management is a continuous process. For example, a systems engineer who was upgrading a server unintentionally opened a security hole on the mail server that provided the capability to perform mail relays. This happened at 1:30 a.m., and the systems engineer discovered his error and closed the hole by 2:00 a.m.. Unfortunately, within that timeframe a hacker discovered the open relay and used the mail server to send "get rich quick" e-mails to more than 2000 individuals. Each of the e-mail addresses included the mail header information, which showed that it was coming from the system engineer's organization. This demonstrates that clear, documented configurations and procedures for changing these configurations are necessary.

Many times, documentation is an afterthought. The more organizations get into the practice of seeing this as an important deliverable of the development process, the more efficient and effective the organization can become because the opportunity for future improvement becomes more visible.

Pragmatic Approach

At first glance, the 18 standards and their related implementation specifications can certainly seem daunting, presenting a case for the senior leadership, the Information Technology team, and the Information Security Officer to head for the emergency room!

The Security Rule was meant to be scalable such that small providers would not be burdened with excessive costs of implementation, and the large providers, health plans, and clearinghouses could take steps appropriate to their business environments. For example, backing up the data of a small provider may be a simple process of rotating the information to an offsite location on a weekly basis from one server, whereas a large operation may contract with a disaster recovery company or may employ electronic vaulting of the information. Decisions have to be made to reasonably protect the information, and document how the decisions were determined. Earlier it was recognized that security is always a risk-based decision, and it is sometimes difficult to determine what is "reasonable" under the circumstances.

A security plan for improvement is the most pragmatic way to move toward HIPAA compliance. Stepwise improvements in the security infrastructure, beginning with an understanding of what risks are being casually accepted within the environment, followed by targeting solutions to mitigate the critical risks, seems reasonable. Early in the process, someone needs to be assigned security responsibility to champion the security efforts. Management support should be obtained through articulation of the risks to the assets, not because "HIPAA requires that we become compliant." This approach only causes management to take a "wait and see" attitude until it is understood what other organizations are doing with the "HIPAA issue." Squarely explaining the risks and incrementally building support through successful delivery is the formula that will provide for longer-term benefits for maintaining the security program. Selling the protection of assets as an ongoing activity provides the view that security is not "done" at the end of the HIPAA project. The idea should be generated

that information is an asset that must be managed on an ongoing basis, just like the financial, human resources, and fixed assets of the organization. Although the temptation may be even stronger to use the "HIPAA Hammer" to pound the message into the organization now that the Final Security Rule has been published, the value of protecting the information assets, providing reduction in long-term "hidden" costs, and the opportunities enabled though secure systems, should be surfaced and promoted. A HIPAA project will have a beginning and an end, but the security program to continue protecting the information assets must survive as a fundamental business operation.

The first task in the plan should be to establish security responsibility, followed by formation of security policy and review committees, development of high-level policies, network assessments, and successive implementation of policies, procedures, and technical implementations to satisfy the various aspects of the HIPAA rule. The key is to get started, somewhere, and begin making progress. Individuals within the organization may already be working on efforts related to one of the security standards — use the opportunity to expand the scope and ensure that the security practices are formalized and documented and will meet the HIPAA security requirements.

Risk, Risk, Risk!

It should be very apparent at this point that much of the "proscriptive" nature of the Security Rule has been changed to an approach that places the emphasis on assessing the risk and determining the implementation choices that are "reasonable and appropriate." True, different organizations may look at the same risk information pertaining to e-PHI and evaluate it differently. This is to be expected, as management teams have different value systems, experiences, and views of criticality. As time goes on, industry best practices for various sizes of organizations, case law, civil suits, cost effective technology innovations, standards development, an increased focus on security and the efforts of local and national associations focused on healthcare and HIPAA will all contribute to the emergence of "*de facto* healthcare security practices." Some of these practices/standards currently exist and others will emerge prior to the compliance date, but this will be an evolutionary refinement process over time as organizations within this industry determine what security approaches support the business of healthcare. Security practices borrowed from other industries are excellent starting points for investigation. Risk assessment and risk management activities should proactively take into account the capabilities to ensure adequate protection of e-PHI.

The change away from the proscriptive nature of the Security Rule makes the rule much easier to read and understand. It also better supports the technology neutrality and scalability principles desired. Some may view the heavy reliance on the risk assessment as the lack of ability to make a "tough" standard. The more appropriate view is that each covered entity must meet the security standard, and the level to which they meet that standard must be consistent with the risk assessment results. In the case of large organizations, with the size, capabilities, and financial ability to implement the addressable specifications, they will most likely be expected to commit the resources. Taking the view that the standard does not need to be taken seriously because it is "addressable" is erroneous.

It is all about risk assessment, documenting the risks, and making good judgments as to the security measures that are reasonable and appropriate for the covered entity's individual situation.

Conclusion

HIPAA should be viewed as an opportunity to address some areas that may not have received attention in the past due to other funding priorities. Protection of health information should be viewed as an opportunity — an opportunity to place some controls around health information such that new processes can be enabled. Technologies continue to emerge with exciting new possibilities, such as wireless access, personal digital assistants, digital photography advances, cell phone proliferation, and instant messaging, to name a few. These new technologies deliver new security challenges as well as new opportunities for collaboration. Creating the proper security foundation will enable these new uses to be exploited, increasing the availability and quality of healthcare, such as Internet health information lookup, while reducing some overhead costs, such as reducing staffing requirements (or providing more funds for increased quality) for customer service.

In the short term, the struggle will continue to move toward compliance. By starting now, HIPAA decisions can be made with more planning and less reaction to the immediate security concern.

How can a covered entity possibly achieve compliance in less than two years (three years for small health plans)? Disney World has the answer. Anyone that has been there knows that it is a magical place where fun things happen and the rest of life is temporarily forgotten. They also know that Disney World has an equally magical way of hiding the length of the lines to the amusements by snaking around one corner, and then the next, showing only a "manageable" line of people directly in front of you. This illusion makes the line seem shorter, as you can see only a little at a time.

Implementing the Security Standards is like Disney World in many ways. It is a very long line, with many dependencies. If we look at the whole line, we might just give up in frustration and decide to try again another day. If we view each security standard as a small line along the way to meeting our goal of protecting e-PHI, the effort does not seem quite so bad.

We are now at Disney World, we have been waiting to stand in line for the past several years, and now is our chance. There is the thrill of anticipation of getting to our destination, coupled with the fear of not getting there on time. But, we are in line now, and we need to celebrate our accomplishments…one turn at a time, and maybe, just maybe, have a little fun along the way.

References

Health Insurance Reform: Security Standards; final rule, February 20, 2003, Federal Register 45 CFR Parts 160, 162 and 164, Department of Health and Human Services.

Security and Electronic Signature Standards — Proposed Rule, August 12, 1998, Federal Register 45 CFR Part 142, Department of Health and Human Services.

Health Insurance Portability and Accountability Act of 1996, August 21, 1996, Public Law 104–191.

The Health Insurance Portability and Accountability Act of 1996 (HIPAA), Centers for Medicare and Medicaid Services, http://cms.hhs.gov/hipaa.

HIPAA Administrative Simplification, Centers for Medicare and Medicaid Services, http://cms.hhs.gov/hipaa/hipaa2.

Standards for Privacy of Individually Identifiable Health Information; final rule, August 14, 2002, Federal Register 45 CFR Parts 160 and 164, Department of Health and Human Services.

144

HIPAA 201: A Framework Approach to HIPAA Security Readiness

David MacLeod, Ph.D., CISSP, Brian Geffert, CISSP, CISA, and David Deckter, CISSP

The Health Insurance Portability and Accountability Act (HIPAA) has presented numerous challenges for most healthcare organizations, but through using a framework approach we have been able to effectively identify gaps and develop plans to address those gaps in a timely and organized manner.

— Wayne Haddad

Chief Information Officer for The Regence Group

HIPAA Security Readiness Framework

Within the U.S. healthcare industry, increased attention is focusing on Health Insurance Portability and Accountability Act (HIPAA) readiness. For the past five years, healthcare organizations (HCOs) across the country have moved to prepare their environments for compliance with the proposed HIPAA security regulations. The past five years have also proved that HIPAA security readiness will not be a point-in-time activity for HCOs. Rather, organizations will need to ensure that HIPAA security readiness becomes a part of their operational processes that need to be maintained on a go-forward basis.

To incorporate HIPAA security readiness into your organization's operational processes, you must be able to functionally decompose your organization to ensure that you have effectively addressed all the areas within your organization. You must also be able to interpret the proposed HIPAA security regulations[1] as they relate to your organization, identify any gaps, develop plans to address any gaps within your current organization, and monitor your progress to ensure you are addressing the identified gaps. For most HCOs, the path to HIPAA security readiness will mean the development of a framework that will allow you to complete the tasks outlined in Exhibit 144.1.

This chapter guides you through the framework that will assist you in identifying and addressing your organization's HIPAA security readiness issues. In doing so, we assume that your organization has already established a HIPAA security team and developed a plan to apply the framework (e.g., Phase 0 activities). Finally, we do not address HIPAA's transactions, code sets, and identifiers (TCI) or privacy requirements, but you will need to consider both sets of requirements as you move through the phases of the framework.

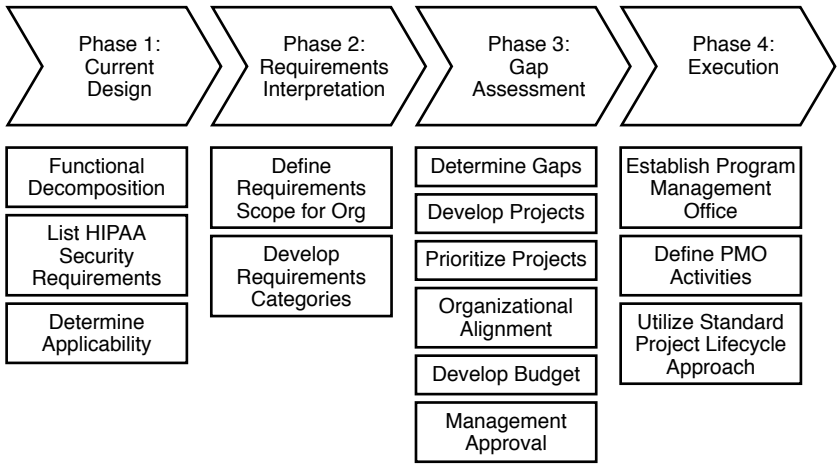


EXHIBIT 144.1 HIPAA security readiness framework.

Phase 1: Current Design[2]

The framework begins with the construction of a matrix that documents your organization's current design. The matrix captures the nuances of the environment (both physical and logical), its business processes, and the initiatives that make your HCO unique. It also lists the HIPAA security requirements and determines the applicability of the requirements to your organization's environment.

Functional Decomposition of the Organization

Organizations have typically approached HIPAA security readiness by starting with the HIPAA security requirements and applying those requirements to their information technology (IT) departments. By relying solely on this approach, organizations have failed to recognize that security is cross-organizational, including business units and individual users alike. Today's Internet era is requiring ever more information sharing, further blurring the boundaries of internal access and external access. How then do you break down your organization to ensure you have adequately addressed all the areas of your organization concerning HIPAA security readiness?

Organizations can functionally decompose themselves in a number of ways, including IT environment, strategic initiatives, key business processes, or locations. To illustrate the idea of functionally decomposing your organization, we provide some examples of processes, applications, IT environment elements, strategic initiatives, and locations for a typical payer and provider in Exhibit 144.2.

List HIPAA Security Requirements

The next step in building the matrix is to list the requirements for the five categories of the HIPAA security regulations as shown in Exhibit 144.3. These include:

- Administrative procedures
- Physical safeguards
- Technical security services
- Technical security mechanisms
- Electronic signatures

Once you have completed the functional decomposition and listed the HIPAA security requirements, you will have created your organization's current design matrix.

EXHIBIT 144.2 HCO Functional Decomposition

Provider (Hospital and Physician)	Payer
Processes	
Administration	Membership and enrollment
Financial	Claims administration
Scheduling	Contract management
Registration	Medical management
Admission, discharge, and transfer	Underwriting and actuarial
Billing and A/R	Provider network management
Insurance verification	Financial management
Practice management	Customer service
Applications	
AMR (EMR, CPR)	Enrollment
Laboratory	Billing and A/R
Radiology	Provider management
Pharmacy	Sales management
Order entry	Medical management
Nurse management	Claims
Financial	Financial
IT Environment	
Wireless	Wireless
WAN	WAN
LAN	LAN
Dial-up	Dial-up
Web	Web
Servers	Servers
Workstations	Workstations
Facilities	Facilities
Databases	Databases
Strategic Initiatives	
Integrating the healthcare enterprise (IHE)	Customer relationship management (CRM)
Electronic medical records	E-business
Web-enabling clinical applications	Electronic data interchange (EDI)
Electronic data interchange (EDI)	
Location	
Hospital	Headquarters
Outpatient clinic	Remote sales office
Off-site storage	Data center

Determine Applicability

The final step in the current design phase will be to determine the areas from the functional decomposition where the security requirements apply. The outcome of this exercise will be an initial list of areas on which to focus for developing the scope of the requirements. Exhibit 144.4 illustrates a partial current design matrix for a typical payer organization.

Phase 2: Requirements Interpretation

The HIPAA security requirements were designed to be used as guidelines, which means that each organization needs to interpret how it will implement them. In this section, we provide some context for defining the scope of each requirement as it applies to your organization, categorizing the practices for the security requirements,

EXHIBIT 144.3 HIPAA Security Requirements List

Administrative Procedures	.308(a)(1)	Certification	
	.308(a)(2)	Chain of Trust Partner Agreement	
	.308(a)(3)	Contingency Plan	Applications and data criticality analysis Data backup plan Disaster recovery plan Emergency mode operation plan Testing and revision
	.308(a)(4)	Formal Mechanism for Processing Records	
	.308(a)(1)	Information Access Control	Access authorization Access establishment Access modification

and developing the approach for meeting the security requirements based on the practices. In addition, we develop one of the security requirements as an example to support each of the steps in the process.

Define the Scope of the Security Requirements

The first step to define the scope of the security requirements is to understand the generally accepted practices and principles and where they apply for each of the requirements. To determine these generally accepted practices and their applications, you can use a number of different sources that are recognized as standards bodies for information security. The standards bodies typically fall into two categories: general practices and industry-specific practices. This is an important distinction because some industry-specific practices may be different from what is generally accepted across all industries (i.e., healthcare industry versus automotive industry). Utilizing industry standards may be necessary when addressing a very specific area of risk for the organization. Exhibit 144.5 provides a short list of standards bodies, although additional standards bodies can be located in the source listing of the HIPAA security regulations.

The next step is to evaluate the generally accepted practices against the description of each security requirement in the HIPAA security regulations, and then apply them to your environment to develop the scope of the requirements for your organization.

For our example, we use the certification requirement. Generally accepted practices for certification include the review of a system or application during its design to ensure it meets certain security criteria. Once implemented, periodic reviews are conducted to ensure the system or application continues to meet those specified criteria. The certification requirement has been defined by the HIPAA security regulations as follows:

> The technical evaluation performed is part of, and in support of, the accreditation process that establishes the extent to which a particular computer system or network design and implementation meet a prespecified set of security requirements. This evaluation may be performed internally or by an external accrediting agency.

To define the scope based on this definition, we focus on two key sets of wording: *computer systems* and *network*. The term *computer system* is generally accepted to include operating systems, applications, databases, and middleware. The term *network* is generally accepted to include the architecture, design, and implementation of the components of the wide area network (WAN), extranet, dial-in, wireless, and the local area network (LAN); and it typically addresses such items as networking equipment (e.g., routers, switches, cabling, etc.). To summarize the scope of our example, we apply the certification requirement to the following areas:

- Network
- Operating systems
- Applications

EXHIBIT 144.4 Partial Current Design Matrix

				Processes				Locations			Applications			IT Environment		
	HIPAA Security Requirements		Claims/Encounters	Customer Service	Membership	Claims	Data Center	Headquarters	Remote Sales Office	Claims	Sales Management	Enrollment	Internet	WAN	LAN
Administrative Procedures	.308(a)(1)	Certification									X	X	X	X	X	X
	.308(a)(2)	Chain of Trust Partner Agreement				X										
	.308(a)(3)	Contingency Plan	Applications and data criticality analysis					X	X	X	X	X	X	X	X	X
			Data backup plan	X	X	X	X				X	X	X			X
			Disaster recovery plan					X	X	X	X					
			Emergency mode operation plan	X	X	X	X	X	X	X						
			Testing and revision	X	X	X	X	X	X	X	X	X	X			X
	.308(a)(4)	Formal Mechanism for Processing Records		X	X	X	X				X	X	X			
	.308(a)(5)	Information Access Control	Access authorization								X	X	X	X	X	X
			Access establishment								X	X	X	X	X	X
			Access modification								X	X	X	X	X	X

EXHIBIT 144.5 Generally Accepted Information Security Standards Bodies

Standards Bodies	Category
United States Department of Commerce — National Institute of Standards and Technology (NIST)	General
System Administration, Networking, and Security (SANS) Institute	General
Critical Infrastructure Assurance Office (CIAO)	General
International Organization for Standardization (ISO) 17799	General
Health Care Financing Administration (HCFA)	Industry-specific: healthcare

EXHIBIT 144.6 Certification Scope and Assumptions

Scope	Network, operating systems, applications, databases, and middleware
Assumptions	None identified
Categories	Policy/standards
	Procedures
	Tools/infrastructure
	Operational

- Databases
- Middleware

In addition, we document any assumptions made during the scoping process, because they will be important inputs to the solution design and as part of the final compliance assessment to understand why some areas were addressed and others were not. Finally, we store this information in each cell containing an X in our current design matrix from the applicability task in the current design phase as shown in Exhibit 144.6.

Develop Requirements Categories

Developing categories for each of the security requirements assists organizations in understanding what needs to be implemented to meet the requirements. Most organizations develop security controls in a technology vacuum, meaning that they see and understand how the technology fits into their organizations, but do not understand the relationship of that technology to the policies, standards, procedures, or operations of their organizations and business. Using the technology-vacuum approach typically develops security solutions that will deteriorate over time because the solution does not have the supporting operational processes to appropriately maintain itself. We define operations as those areas that support and maintain the technology within the organization, such as assigning owners who are responsible and accountable for the technology and its supporting processes. By taking a more holistic approach that includes policies/standards, procedures, technology, and operations, you will develop security solutions to address your gaps that can be more rapidly implemented and maintained over time. Based on this approach, we typically use the following four categories for grouping the practices identified through defining the scope of requirements in the section above:

1. *Policies or standards.* Policies include senior management's directives to create a computer security function, establish goals for the function, and assign responsibilities for the function. Standards include specific security rules for particular information systems and practices.
2. *Procedures.* Procedures include the activities and tasks that dictate how the policies or supporting standards will be implemented in the organization's environment.
3. *Tools or infrastructure.* Tools or infrastructure includes the elements that are necessary to support implementation of the requirements within the organization such as process, organizational structure, network and system-related controls, and logging and monitoring devices.

EXHIBIT 144.7 Practice Categories — Certification

Administrative Procedures — Certification	
Categories	**Practices**
Policies or standards	Written policy that identifies certification requirements
	Policy identifies individuals responsible for implementing that policy and defines what their duties are
	Policy identifies consequences of noncompliance
	Security standards for the configuration of networks, security services and mechanism, systems, applications, databases, and middleware
Procedures	Identifying certification need review
	Precertification review
	Certification readiness
	Periodic recertification review
Tools or infrastructure	Precertification readiness tool
	Certification criteria tool (standards)
	Certification compliance issue resolution tool
Operational	Operational when the following criteria are established:
	Owner
	Budget
	Charter
	Certification plan

4. *Operational.* Operational includes all the activities and supporting processes associated with maintaining the solution or system and ensuring it is running as intended. Typically, an owner is assigned to manage the execution of the activities and supporting processes. Examples of activities and supporting processes include maintenance, configuration management, technical documentation, backups, software support, and user support.

In addition, the categories will be used to monitor your progress with implementing the practices related to each requirement. To continue with our certification requirement example, we have identified some practices related to certification and placed them into categories as illustrated in Exhibit 144.7.

Finally, we store this information in the current design matrix as illustrated in Exhibit 144.8.

By completing your organization's current design matrix, you have developed your organization's to-be state, which includes a minimum set of practices for each area of your organization based on your interpretation of the HIPAA security requirements. You can now use this to-be state to conduct your gap assessment.

Phase 3: Gap Assessment

With interpretation of the HIPAA security requirements complete, you are ready to conduct your HIPAA security readiness or gap assessment. The time it will take to conduct the assessment will vary greatly, depending on a number of factors that include, at a minimum, the size of the organization, the number of locations, the number of systems/applications, and current level of maturity of the security function within the organization. An example of a mature security organization is an organization with a defined security policy, an established enterprise security architecture (ESA), documented standards, procedures with defined roles and responsibilities that are followed, established metrics that measure the effectiveness of the security controls, and regular reporting to management.

The outcome of the assessment provides you with gaps based on your previously defined scope and practices for each of the security requirements. Because the identified gaps will pose certain risks to your organization, an important point to keep in mind, as your organization reviews the assessment gaps, is that your organization will not be able to address all the gaps due to limited time and resources. Typically, the gaps that you can translate into business risks need to be addressed, particularly the ones that will affect your organization's HIPAA TCI and privacy initiatives. One way of determining if a particular gap poses a business risk to the organization is to answer the question, "So what?" (by which we mean that, if we do not address this risk, how will it adversely impact our business?). For example, application security access controls are lacking on extranet-accessible applications, allow-

EXHIBIT 144.8 Certification Categories

Scope	Network, operating systems, applications, databases, and middleware
Assumptions	None identified
Categories	Policy/standards: 1. Written policy that identifies certification requirements 2. Policy identifies individuals responsible for implementing that policy and what their duties are 3. Policy identifies consequences of noncompliance 4. Security standards for the configuration of networks, security services, and mechanism, systems, applications, databases, and middleware Procedures: 1. Identifying certification need review 2. Precertification review 3. Certification readiness 4. Periodic recertification review Tools/infrastructure: 1. Precertification readiness tool 2. Certification criteria tool (standards) 3. Certification compliance issue resolution tool Operational: 1. Operational when the following criteria are established: A. Owner, budget, charter, and certification plan

ing for the compromise of sensitive health information and clearly having an adverse impact on your bottom line. If the gap does not adversely affect your business at this point in time, document the gap because it may become a business risk in the future. For example, consider an operating system that supports a nonsensitive application that has not been certified. The application, however, will be replaced in 30 days with a newer version that requires another operating system altogether. Therefore, there is no adverse impact on your bottom line. However, if the organization has resources available, then consider taking actions to mitigate the risk posed by the gap.

Once you have completed your assessment and identified your gaps, you need to define a set of projects to remediate the issues. After you have defined these projects, you need to determine the resources and level of effort required to complete the projects, prioritize them, and develop a budget. In addition, you need to obtain organizational alignment around the projects. Finally, you need to get management approval for the projects.

Defining Projects

Gaps are identified based on analysis of prior requirements and then reevaluated against strategic initiatives to determine a project assignment. That is, some gaps are dealt with as stand-alone HIPAA security projects, and others are bundled or packaged within projects that more directly support strategic goals. A typical set of projects developed from an assessment includes the following:

- *High-risk mitigation.* Address high-risk vulnerabilities and exposures to your bottom line that were discovered as part of your assessment.
- *Security management.* Address the development of the core security plans and processes required to manage the day-to-day business operations at an acceptable level of risk, such as reporting and ownership, resources and skills, roles and responsibilities, risk management, data classification, operations, and maintenance for security management systems.

- *Policy development and implementation.* Address the development of security policies and standards with a supporting policy structure, a policy change management process, and a policy compliance function.
- *Education and awareness.* Address areas such as new employee orientation to meet legal and HR requirements, ongoing user and management awareness programs, and ongoing user training and education programs.
- *Security baseline.* Address development of an inventory of information assets, networking equipment, and entity connections to baseline your current environment.
- *Technical control architecture.* Address the development of a standards-based security strategy and architecture that is aligned with the organization's IT and business strategies and is applied across the organization.
- *Identity management solution.* Address the consistent use of authorization, authentication, and access controls for employees, customers, suppliers, and partners.
- *Physical safeguards.* Address physical access controls and safeguards.
- *Business continuity planning/disaster recovery planning.* Address an overall BCP/DRP program (backup and recovery plan, emergency mode operation plan, recovery plan, and restoration plan) to support the critical business functions.
- *Logging and monitoring.* Address monitoring, logging, and reporting requirements, as well as developing and implementing the monitoring architecture, policies, and standards
- *Policy compliance function.* Address the development of a policy compliance auditing and measurement process, which will also identify the process for coordinating with other compliance activities such as internal audit, regulatory, etc.
- *HIPAA security readiness support.* Address the management of the overall SRAP and supporting compliance assessment activities.

Once you have defined the projects, you have to estimate the resources and level of effort required to complete each of the projects. In addition, following management approval, further refinement of the estimate will be necessary during the scoping and planning phase of the project lifecycle.

Prioritizing Projects

For the identified projects, you need to prioritize them based on preselected criteria such as:

- *HIPAA interdependencies.* Does the project support HIPAA readiness for security, privacy, or TCI? For example, a project that includes the development of a data classification scheme can support both privacy and security.
- *Strategic initiatives.* Does the project support strategic initiatives for the organization? For example, a project that includes the development of a service to e-mail members' explanation of benefits (EOB) supports the strategic initiative to reduce paper-based transactions while facilitating HIPAA readiness for security and privacy.
- *Cost reduction.* Does the project help the organization reduce costs? For example, a project that includes the development of a VPN solution can support HIPAA security implementation requirements as well as support cost-reduction efforts related to migrating providers from extranet-based or dial-up access over the WAN to the Internet.
- *Improve customer service/experience.* Will the project improve customer service/experience? For example, implementing user provisioning and Web access control solutions supports HIPAA security implementation requirements, as well as improves the customer experience by allowing for single sign-on (SSO) and the ability for end users to reset their own passwords with a challenge–response.
- *Foundation building.* Does the project facilitate the execution of future projects, or is it in the critical path of other necessary projects? For example, an organization will need to execute the project to develop and implement policies before executing a project to facilitate compliance.

Based on the prioritization, you can then arrange the projects into an initial order of completion or plan to present them for review by the organization.

Develop Budget

Once you have the proposed plan developed, you need to develop an initial budget, which should include:

- Resources to be used to complete the project
- The duration of time needed to complete the project
- Hardware or software required to support the project's completion
- Training for new processes, and hardware or software additions
- Capitalization and accounting guidelines

Organizational Alignment and Management Approval

The plan you present to the organization will consist of the projects you have defined based on the gaps in your assessment, the resources and time needed to complete the projects, and the order of the projects' completion based on prioritization criteria. Based on input from the organization, you can modify your plan accordingly. The outcome of this activity will be to gain organizational buy-in and approval of your plan, which is especially critical when you require resources from outside of your organizational area to complete the projects.

Phase 4: Execution

Execution deals with both the management of projects and the reporting of completion status to the organization.

Program Management Office

Due to the sheer number of projects, the amount of work required to complete those projects, and the need to manage the issues arising from the projects, a formal program management office (PMO) and supporting structure will be required for the successful completion of your projects on time and within budget. You do not necessarily have to create your own security PMO, but instead you may wish to leverage an existing overall HIPAA or enterprise PMO to assist you with your project execution.

Define PMO Activities

Typically, a PMO performs the following activities:

- *Provides oversight for multiple projects.* Prioritize projects, manage project interdependencies and corresponding critical path items.
- *Manages the allocation of resources.* De-conflict resource constraints and shortages resulting from multiple project demands.
- *Manages budget.* Manage the budget for all related projects.
- *Resolves issues.* Facilitate resolution of issues both within projects and between cross-organizational departments.
- *Reports status.* Provide status reports on a periodic basis to oversight committees and management to report on the progress, issues, and challenges of the overall program.

Utilize a Standard Project Lifecycle Approach

Organizations should utilize a project lifecycle approach with a standard set of project documentation. Using a standard project lifecycle approach will streamline the design and implementation activities and support consistent, high-quality standards among different project teams and, potentially, different locations.

Summary

Addressing HIPAA security readiness may seem like an unmanageable task for most organizations. As outlined in this chapter, by applying a framework approach to break down the task into manageable pieces, you should

be able document your organization's current design, effectively identify your organization's gaps, develop an action plan to address those gaps, and execute that plan in an organized and systematic manner.

Notes

Department of Health and Human Services (HHS) 45 CFR, Part 142 — Security and Electronic Standards; Proposed Rule published in the Federal Register (August 12, 1998). Any reference to the HIPAA security regulations in this chapter refer to the proposed HIPAA security regulations.
The framework can be used for any organization to address information security readiness by simply modifying, adding or changing the criteria (HIPAA security regulations, FDA regulations, ISO 17799, NIST, SANS, etc.).

References

1. Guttman, Barbara and Roback, Edward, A., An introduction to computer security, *The NIST Handbook*; NIST Special Publication 800-12; U.S. Department of Commerce, Technology Administration, National Institute of Standards and Technology.
2. *Federal Register,* Part III, Department of Health and Human Services, 45 CFR Part 142 — Security and Electronic Signature Standards; Proposed Rule, August 12, 1998.
3. Scholtz, Tom, *Global Networking Strategies —The Security Center of Excellence*; META Group; April 19, 2001.
4. *Practices for Securing Critical Information Assets*; Critical Infrastructure Assurance Office, January 2000.
5. Rishel, W. and Frey, N., Strategic Analysis Report R-14-2030, *Integration Architecture for HIPAA Compliance: From 'Getting It Done' to 'Doing It Right,'* Gartner, August 23, 2001.
6. Guttman, Barbara and Swanson, Marriane, *Generally Accepted Principles and Practices for Security Information Technology Systems*; NIST Special Publication 800-14; U.S. Department of Commerce, Technology Administration, National Institute of Standards and Technology.

Internet Gripe Sites: *Bally v. Faber*

Edward H. Freeman

EVERY LARGE ORGANIZATION HAS UNHAPPY CUSTOMERS AND DISGRUNTLED EMPLOYEES. Until recently, a dissatisfied person had a limited number of ways of expressing his complaints, reaching only a small group of friends and sympathizers. Organizations would often simply ignore the situation, realizing that public denials would only draw more attention to the complaint.

The phenomenal growth of the Internet has made it easier for unhappy customers to criticize organizations and to have their complaints heard by a large audience. *Gripe sites* have become common on the Internet. Almost every large organization and many smaller ones have been the subject of gripe sites. Such sites not only display the operator's dissatisfaction but also allow unhappy customers, employees, competitors, and vendors to post their complaints against the organization. Sensitive internal documents have found their way onto these Web pages. Potential customers and job seekers often visit these sites before deciding whether to do business or to accept a job offer. Such sites may receive thousands of hits monthly.

Gripe sites can be a small but genuine source of embarrassment, even for large, seemingly untouchable corporations. Due to the open nature of the Internet, anyone with a computer and a complaint can purchase a Web site with a derogatory name for under $100. Complaints posted on the sites are often untraceable so there is no way for potential customers to know whether what they read there is true.

This chapter discusses *Bally v. Faber,* a 1998 federal court decision dealing with gripe sites. Bally Total Fitness, a nationwide chain of exercise clubs, attempted to shut down an Internet gripe site that used its registered trademark negatively. The column deals with trademark in-fringement and dilution and offers practical advice for concerned corporations. Actual court cases are cited as examples throughout the column.

The Legal Issues of Disaster Recovery Planning

Tari Schreider

Payoff

The legal issues involved in corporate contingency planning are some of the most misunderstood and confusing aspects of the entire process of creating a disaster recovery plan. Data center managers often must assume the role of disaster recovery planners, and whereas they are not expected to be as knowledgeable as lawyers in this role, they are encumbered with the responsibility of understanding the minutiae of existing regulatory guidelines and the legal consequences of their companies' failure to implement an effective disaster recovery plan. No specific laws state categorically that an organization must have a disaster recovery plan, but there is a body of legal precedents that can be used to hold companies responsible to those affected by a company's inability to cope with or recover from a disaster. This article outlines those precedents and suggests precautions.

Introduction

Despite the widespread reporting in the media of disasters and their effects, many companies, corporate directors, and officers remain apathetic toward implementing a disaster recovery plan. Companies are generally unwilling to commit the finances and resources to implement a plan unless they are forced to do so. However, implementing a proper disaster recovery plan is a strategic, moral, and legal obligation to one's company.

If the billions of dollars spent annually on technology to maintain a competitive edge is an indication of how reliant society is on technology, then failing to implement a disaster recovery plan is an indication of corporate negligence. Standards of care and due diligence are required of all corporations, public or private. Not having a disaster recovery plan violates that fiduciary standard of care.

The legal issues involved in corporate contingency planning are some of the most misunderstood and confusing aspects of the entire process of creating a disaster recovery plan. Disaster recovery planners are not expected to be lawyers; however, they are encumbered with the responsibility of understanding the minutiae and vagueness of existing regulatory guidelines and the legal consequences of their companies' failure to implement an effective disaster recovery plan. Although no specific laws state categorically that an organization must have a disaster recovery plan, there is a body of legal precedents that can be used to hold companies and individuals responsible to those affected by a company's inability to cope with or recover from a disaster.

The entire basis of law relating to the development of disaster recovery plans is found in civil statutes and an interpretation of applicability to disaster recovery planning. These legal precedents form the basis of this article.

One of the precedents that can be used against companies that fail to plan for a disaster is drawn from the case of FJS Electronics v. Fidelity Bank. In this 1981 case, FJS Electronics sued Fidelity Bank over a failure to stop payment on a check. Although the failure to stop payment of the check was more procedural in nature, the court ruled that Fidelity Bank assumed, and therefore was responsible for, the risk that the system would fail to stop a check. FJS was able to prove that safeguards should have been in place and therefore was awarded damages.

This case shows that the use of a computer system in business does not change or lessen an organization's duty of reasonable care in its daily operations. The court ruled that the bank's failure to install a more flexible, error-tolerant system inevitably led to problems. As a result, information technology professionals will be held to a standard of reasonable care. They can breach that duty to maintain reasonable care by not diligently pursuing the development of a disaster recovery plan.

Categories of Applicable Statutes

To help make the data center manager aware of the areas in which disaster recovery planning and the law intersect, Contingency Planning Research, Inc., a White Plains NY-based management consulting firm, has categorized the applicable statutes and illustrated each with an example. Each area is described; however, this discussion is not intended to present a comprehensive list.

Categories of statutes include but are not limited to the following:

- **Contingency Planning Statutes.** These apply to the development of plans to ensure the recoverability of critical systems. An example is the Federal Financial Institutions Examination Council (FFIEC) guidelines, which replace previously issued Banking Circulars BC-177 and BC-226.

- **Liability Statutes.** These statutes establish levels of liability under the “Prudent Man Laws” for directors and officers of a corporation. An example is the Foreign Corrupt Practices Act (FCPA).

- **Life/Safety Statutes.** These set out specific ordinances for ensuring the protection of employees in the workplace. Examples include the National Fire Protection Association (NFPA) and the Occupational Safety & Health Administration (OSHA).

- **Risk-Reduction Statutes.** These stipulate areas of risk management required to reduce or mitigate (or both) the effects of a disaster. Examples include Office of the Comptroller of the Currency (OCC) Circular 235 and Thrift Bulletin 30.

- **Security Statutes.** These cover areas of computer fraud, abuse, and misappropriation of computerized assets. An example is the Federal Computer Security Act.

- **Vital Records Management Statutes.** These include specifications for the retention and disposition of corporate electronic and hard-copy (i.e., paper) records. An example is the body of IRS Records Retention requirements.

Statutory Examples

When the time comes for the data center manager to defend his or her company against a civil or criminal lawsuit resulting from damages caused by the company's failure to meet a standard of care, he or she needs more than an “Act of God” defense. When no direct law or statute exists for a specific industry, the courts look instead to other industries for guidelines and legal precedents. The following three statutes represent the areas in which a court most likely seek a legal precedent.

The Foreign Corrupt Practices Act (FCPA)

The Foreign Corrupt Practices Act (FCPA) of 1977 was originally designed to eliminate bribery and to make illegal the destruction of corporate documents to cover up a crime. To accomplish this, the FCPA requires corporations to “make and keep books, records, and accounts, which, in reasonable detail, accurately and fairly reflect the transactions and dispositions of the assets...” The section of this act that keeps it at the forefront of disaster recovery liability is the “standard of care” wording, whereby management can be judged on their mismanagement of corporate assets.

The FCPA is unique in that it holds corporate managers personally liable for protecting corporate assets. Failure to comply with the FCPA exposes individuals as well as companies to the following penalties:

- Personal fines up to $10,000.
- Corporate fines up to $1,000,000.
- Prison terms up to five years.

The Federal Financial Institutions Examinations Council

The comptroller of the currency has issued various circulars dating back to 1983 (e.g., Banking Circular BC-177) regarding the need for financial institutions to implement disaster recovery plans. However, in 1989, a joint-agency circular was issued on behalf of the following agencies:

- The Board of Governors of the Federal Reserve System (FRB).
- FDIC.
- The National Credit Union Administration (NCUA).
- The Office of the Comptroller of the Currency (OCC).
- The Office of Thrift Supervision (OTS).

The circular states, “The loss or extended interruption of business operations, including central computing processing, end-user computing, local-area networking, and nationwide telecommunications, poses substantial risk of financial loss and could lead to failure of an institution. As a result, contingency planning now requires an institution-wide emphasis...”

The Federal Financial Institutions Examinations Council guidelines relating to contingency planning are actually contained within 10 technology-related Supervisory Policy Statements. These policies are revised every two years and can be acquired through any of the five agencies listed earlier in this section.

The Consumer Credit Protection Act

On November 10, 1992, the 95th Congress, 2nd Session, amended section 2001of the Consumer Credit Protection Act (15 U.S.C. 1601 et seq.) “TITLE IX-Electronic Funds Transfers.” The purpose of this amendment was to remove any ambiguity the previous

statute had in identifying the rights and liabilities and consumers, financial institutions, and intermediaries in “Electronic Funds Transfers.” This Act covers a wide variety of industries, specifically those involved in electronic transactions originating from point-of-sale transfers, automated teller machines, direct deposits or withdrawals of funds, and fund transfers initiated by telephone. The Act further states that any company that facilitates electronic payment requests that ultimately result in a debit or credit to a consumer account must comply with the provisions of the Act.

Failure to comply with the provisions of this Act exposes a company and its employees to the following liabilities:

- Any actual damage sustained by the consumer.
- Amounts of not less than $100 and not greater than $1,000 for each act.
- Amounts of $500,000 or greater in class action suits.
- All costs of the court action and reasonable attorneys' fees.

Companies covered under this Act are subject to all the liabilities and all the resulting damages approximately caused by the failure to make an electronic funds transfer. The Act states that a company may not be liable under the Act if that company can demonstrate a certain set of circumstances. The company must show by a “preponderance of evidence” that its actions or failure to act were caused by “...an Act of God or other circumstances beyond its control, that it expressed reasonable care to prevent such an occurrence, and that it expressed such diligence as the circumstances required...”

Standard of Care.

Each of these three statutes mentioned in this section is based on the precept of standard of care, which is described by the legal publication entitled Corpus Juris Secundum, Volume 19, Section 491. The definition is that “... directors and officers owe a duty to the corporation to be vigilant and to exercise ordinary or reasonable care and diligence and the utmost good faith and fidelity to conserve the corporate property; and, if a loss or depletion of assets results from their willful or negligent failure to perform their duties, or to a willful or fraudulent abuse of their trust, they are liable, provided such losses were the natural and necessary consequences of omission on their part...”

Determining Liability

Courts determine liability by weighing the probability of the loss occurring compared to the magnitude of harm, balanced against the cost of protection. This baseline compels companies to implement a reasonable approach to disaster recovery in which the cost of implementation is in direct correlation to the expected loss. In other words, if a company stands to lose millions of dollars as a result of an interruption to its computerized processing, the courts would take a dim view of a recovery plan which lacked the capability to restore the computer systems in a timely manner.

Another precedent-setting case, referred to as the Hooper Doctrine, can be cited when courts are looking to determine a company's liability. This doctrine establishes that even though many companies do not have a disaster recovery plan, there are “precautions so imperative that even their universal disregard does not excuse their omission.” Simply put,

a company cannot use, as a defense, the fact that there are no specific requirements to have a disaster recovery plan and that many other companies do not have one.

Liability is not just related to corporations but extends to individuals who develop disaster recovery plans as well. In 1989, in Diversified Graphics v. Ernst & Whinney, the United States Eighth Circuit Court of Appeals handed down a decision finding a computer specialist guilty of professional negligence. In this case, professional negligence was defined as a failure to act reasonably in light of special knowledge, skills and abilities.

If the directors and officers of a corporation can be held accountable for not having a disaster recovery plan, then this case provides the precedent for individuals who are certified disaster recovery planners to be held personally accountable for their company's disaster recovery plan.

Insurance as a Defense

Directors and officers (D&O) of companies have a fiduciary responsibility to ensure that any and all reasonable efforts are made to protect their companies. D&O insurance does exist, but it only protects officers if they used good judgment and their decisions resulted in harm to their company or employees, or both. D&O insurance does not cover, however, a company officer who fails to exercise good judgment (e.g., by not implementing a disaster recovery plan).

Errors and omissions (E&O) insurance covers consequential damages that result from errors, omissions, or negligent acts committed in the course of business, or from all of these together. In a 1984 precedent-setting case heard in the District Court of Ohio, the court ruled, “Negligence is a failure to exercise the degree of care that a reasonably prudent person would exercise under the same circumstance.” With regard to a trade, practice, or profession, the court added that “the degree of care and skill required is that skill and knowledge normally possessed by members of that profession in good standing in similar communities.” Liability insurance does not prevent the organization from being brought to court, but it will pay toward the litigation and penalties incurred as a result.

Disaster recovery practitioners possess a unique expertise and subsequently could be held accountable for their actions and advice in the development of a disaster recovery plan. A word of caution here is that if the data center manager passes himself or herself off as an expert, he or she should expect to be held accountable as an expert.

Conclusion

Courts assess liability by determining the probability of loss, multiplying it by the magnitude of the harm, and balancing them against the cost of prevention. Ostensibly, should the data center manager's company end up in court, the burden of proof would be on the company to prove that all reasonable measures had been taken to mitigate the harm caused by the disaster. There are clearly enough legal precedents for the courts to draw on in determining if a standard of care was taken or if due diligence was exercised in mitigating the effects of the disaster on the company's critical business operations. Every business is governed by laws that dictate how it must conduct itself in the normal course of business. By researching these laws and statutes, the data center manager will eventually find where penalties for non-performance are stipulated. These penalties become the demarcation point for reverse engineering the business operations, thus finding the points of failure that could affect the company's ability to perform under the statutes that specifically govern the company's business.

Author Biographies

Tari Schreider

Tari Schreider is director of research with Contingency Planning Research, Inc. (CPR), an eight-year-old management consulting firm dedicated to disaster recovery, contingency planning, and risk management. CPR specializes in helping organizations prepare for and recover from disasters and their consequential effects. CPR has conducted consulting engagements throughout the United States and its research reports are circulated internationally. CPR is based in White Plains NY and can be contacted at (800) CPR-5511.

State Control of Unsolicited E-mail: State of Washington *v.* Heckel

Edward H. Freeman

ONE OF THE MOST FREQUENT COMPLAINTS FROM INTERNET USERS CONCERNS THE ENDLESS FLOOD OF UNWANTED E-MAIL, ALSO KNOWN AS *SPAM*. These unsolicited messages attempt to sell everything from get-rich pyramid schemes to stop-smoking seminars, from Viagra to chain letters to hair-loss treatments. No Internet user is immune from this constant barrage of unsolicited e-mail. In the world of spamming, no claim is too preposterous and no promise is too fantastic.[1]

Bulk e-mail is very inexpensive for the sender, requiring only a basic personal computer and modem. For about $249, the sender can receive a CD-ROM containing over 11,000,000 e-mail addresses. There are no postage or printing costs and no reason for the sender to support a full-time staff to process orders or deal with customer inquiries.

Spammers transfer the costs associated with bulk e-mailing to the end user and to the Internet service provider (ISP). ISPs must provide additional bandwidth and storage devices to process and forward unsolicited e-mail messages. They must maintain additional storage to save messages for delivery to the intended recipient. These costs are eventually passed on to the e-mail user.

This column deals with attempts by the State of Washington to enforce tough ordinances against spam. It discusses the *Commerce Clause* of the U.S. Constitution and how individual states can and cannot limit commercial activities among residents and corporations of different states. Actual court cases are used throughout to highlight specific points.

THE STATE OF WASHINGTON'S ANTI-SPAM LAW

In March 1998, the Washington Legislature unanimously passed the Unsolicited Electronic Mail Act [The Act]. It stated:

1. No person, corporation, partnership, or association may initiate the transmission of a commercial electronic mail message from a computer located in Washington or to an electronic mail address that the sender knows, or has reason to know, is held by a Washington resident that:
   — uses a third party's Internet domain name without permission of the third party, or otherwise misrepresents any information in identifying the point of origin or the transmission path of a commercial electronic mail message
   — contains false or misleading information in the subject line.
2. For purposes of this section, a person, corporation, partnership, or association knows that the intended recipient of a commercial electronic mail message is a Washington resident if that information is available, upon request, from the registrant of the Internet domain name contained in the recipient's electronic mail address.[2]

Fines for violation of the act ranged from $100 to $1000 per e-mail.

As originally proposed, the Act would have completely prohibited sending unsolicited e-mail messages to Washington residents. The Legislature eliminated the concept of a total ban during preliminary deliberations because of challenges from the ACLU and other free-speech advocates. Opponents felt that the Act contained an "exceedingly broad definition of unsolicited commercial speech."[3] These challenges convinced the Legislature to regulate spam indirectly by prohibiting false or misleading commercial e-mail. The Legislature felt that such restrictions were more consistent with First Amendment concerns.[4]

The Act specifically banned two practices commonly used by spammers:

- It prohibited messages containing misleading or incorrect information about its point of origin or return e-mail address.
- It also prohibited false or misleading information in the subject line. Spammers will frequently use a subject line such as "You have just won $1000" or "Employment opportunities in your field" to encourage curious users to open their messages rather than delete them without reading. If an e-mail had such a subject line and then promoted a pyramid marketing scheme, the e-mail violated the Washington law.

Because the Act was state law, its scope was limited geographically to Washington. A spammer could violate the Act only if his computer was physically located in Washington or if the sender knew that the recipient

was located there. As defined in the Act, the sender was considered to know that the receiving party was located in Washington if "that information is available, upon request, from the registrant of the Internet domain contained in the recipient's electronic mail address."[5]

The Attorney General and the Washington Association of Internet Service Providers (WAISP) co-sponsored a statewide registry of e-mail accounts held by Washington residents. Washington e-mail subscribers could register their accounts by accessing the WAISP Registry Page at hyperlink http://registry.waisp.org. According to the Act, spammers were expected to check potential recipients against the WAISP listing to determine whether the user resided in Washington and to remove the user if the e-mail message violated the terms of the Act.

THE FACTS OF THE CASE

At the time of the litigation, Jason Heckel, in his mid-20s, was the sole proprietor of Natural Instincts in Salem, Oregon. In 1997, Heckel developed and printed a 46-page booklet called "How to Profit from the Internet." The booklet sold for $39.95.

To market the booklet, Heckel used a software package called Extractor Pro. The package finds e-mail addresses on the Internet and automatically sends e-mail to each of those addresses. Heckel sent up to 1,000,000 unsolicited e-mails monthly to promote his booklet. These messages went to Internet users all over the world, including users in Washington. The suit claimed that he sold about 40 booklets each month.

Heckel's methods of marketing his pamphlet were typical for spammers:

- He sent his messages on an indirect, circuitous path all over the Internet, making it impossible to determine the origin of the message.
- He gave recipients no way to reply to his messages. The e-mail address cited in the "sender" field did not exist. If recipients complained about his messages, the complaints were returned to the sender as undeliverable because of an invalid e-mail address. If a user wanted to purchase Heckel's pamphlet, he would have to use regular mail along with a credit card number.
- He used a deceptive subject line, such as "Do I have the right address?" This fooled users into opening the e-mail, thinking that the message was from a long-lost friend or associate.

The Washington Attorney General's office received several complaints about Heckel. They sent a warning letter to Heckel, asking him to discontinue sending his messages to Washington residents. When he refused to comply, they sued Heckel in Washington Superior Court, charging that he had violated the terms of the Act.

At trial, Heckel's attorney asked that the court dismiss the case, claiming that the Act violated the Commerce Clause of the U.S. Constitution.[6] The Commerce Clause limits the rights of individual states to restrict interstate commerce if the burden imposed on interstate commerce is excessive.

On March 10, 2000, Judge Parker Robinson of the King County Superior Court granted Heckel's motion and dismissed the case, ruling that the Act was unconstitutional. According to Judge Robinson, the Act was "unduly restrictive and burdensome." It placed a burden on business that clearly outweighed the benefits to consumers. In cyberspace, it is difficult to determine the state in which each e-mail recipient resides. This would subject "someone like Mr. Heckel to potentially 50 different standards of commerce, which I think is a problem in terms of the commerce clause."[7]

On April 10, 2000, the Attorney General's office filed an appeal of Judge Robinson's ruling to the State Supreme Court. As of July 2000, the higher court had not yet reached a decision.[8]

THE INTERSTATE COMMERCE CLAUSE

At the end of the American Revolution, individual states attempted to regulate interstate and international commerce with only their own interests in mind. The Confederation Congress, which represented the states until the adoption of the U.S. Constitution, had no authority to regulate commerce among the states. With each state guarding its own unique interests, 13 conflicting systems of commercial regulation and tax policies governed trade in the new country. This led to conflicts among the states as states retaliated against each other with different markets, tariffs, and industries.[9]

In January 1786, the Virginia Legislature called for a national convention to consider a uniform system of commerce regulation. At the Constitutional Convention in 1787, Congress was empowered to "regulate commerce with foreign nations, and among the several states, and with the Indian tribes." This congressional power, known as the Commerce Clause, gave Congress the power to regulate economic life in the nation and to promote the free flow of interstate commerce, including action within state borders that interfered with that flow.[10] This reduced the potential for economic warfare among the states.

There is a natural conflict between a state's right to control and regulate its own activities and the federal government's desire to maintain control over interstate commerce. The terms of the Commerce Clause have led to numerous Supreme Court decisions. The Court interprets the Commerce Clause as granting virtually complete power to Congress to regulate the economy and business. A court may invalidate state legislation under the Commerce Clause after balancing several factors:

- the necessity and importance of the state regulation upon interstate commerce
- the burden it imposes upon interstate commerce
- the extent to which it discriminates against interstate commerce in favor of local concerns

The states do have certain powers to make laws governing matters of local concern. The courts use a three-part test to determine whether states can regulate a specific form of interstate commerce:[11]

- Does the law discriminate against another state?
- Does the substance of the law require national or uniform regulation?
- Do the interests of the state outweigh the federal government's right to regulate interstate commerce?

The courts usually analyze these factors on a case-by-case basis. In discussing this analysis, the Supreme Court summarized this method. "Where the statute regulates even-handedly to effectuate a legitimate local public interest, and its effects on interstate commerce are only incidental, it will be upheld unless the burden imposed on such commerce is clearly excessive in relation to the putative local interest."[12]

An example of this analytical method arose in a classic 1949 Supreme Court decision. H.P. Hood was a Massachusetts milk distributor that purchased milk from farmers in New York state. Hood brought the milk to its Massachusetts plants and then sold it in Boston. Hood applied to the New York Commissioner of Agriculture and Markets for permission to open another receiving station. The Commissioner denied Hood's request on the ground that the proposed plant would divert milk from the New York market and thereby cause milk prices to rise in New York.

The Supreme Court ruled that New York could not curtail interstate commerce to keep prices lower for New York purchasers. This action would have set up a barrier to free trade among the states. A state may not use the power to tax or use its police powers to establish an economic barrier to competition with the products of another state. Such actions were a violation of the Commerce Clause and were therefore unconstitutional.[13]

The courts will continue to refine the Commerce Clause in future decisions. It is possible that the Supreme Court will eventually decide *Heckel* or a similar case in another state.

ANALYSIS OF *HECKEL*

As previously noted, Judge Robinson's decision stated that the Act was unconstitutional because it violated the Commerce Clause. The decision has drawn generally negative reviews in the cyberspace community. These criticisms were based on three major factors:

- Some critics felt that spam does not rise to the level of interstate commerce protected by the Commerce Clause. No commercial transaction has occurred between the spammer and the recipient, merely an unsolicited and usually unwanted e-mail.
- Judge Robinson felt that it would be "burdensome" for Heckel to determine which recipients live in Washington. Critics have noted that allowing Heckel to send his spam places a burden on both the ISPs and e-mail recipients. Heckel's "right" to send out his messages means that ISPs must provide additional hard drive space to store messages. Users must spend time deleting such messages. Clearly, Heckel's spam constituted a burden to ISP's and e-mail users, both in productivity and in added hardware costs.
- States long ago enacted consumer-protection measures, such as restricting out-of-state telemarketers and junk faxes. There is no real difference between these unwanted methods of advertising and spam.

A higher court will ultimately decide these issues.

CONCLUSION

Experts agree that spam is here to stay. Most Internet users dislike unsolicited, sometimes offensive messages. Spam has become an inexpensive method of advertising and of sending messages throughout the world. Unfortunately, it will continue to attract unscrupulous, fraudulent operators selling every product imaginable as well as some products that are not imaginable.

Legislators, attorneys, civil libertarians, and cyberspace experts will continue to search for a constitutionally acceptable method of reducing unsolicited e-mail, especially when theft, fraud, or abusive conduct is involved. The courts will decide what level of protection from spam is constitutionally sound under the Commerce Clause.

Notes

1. Patty Wentz, "The War on Spam," *Williamette Week,* November 11, 1998.
2. Wash. Rev. Code §19.190.020 (1998).
3. Peter Lewis, *Spam on Trial,* Seattle Times, June 7, 1998, C1 (quoting ACLU's Jerry Sheehan).
4. Note, *"Washington's 'Spam-Killing' Statute: Does It Slaughter Privacy in the Process,"* 74 5. Wash. L.R. 453 (1999).
5. Wash. Rev. Code §19.190.020(1) (1998).
6. Art. I, 8-3.
7. Peter Lewis, *Anti-spam E-mail Suit Tossed Out,* Seattle Times, March 14, 2000.
8. Peter Lewis, *State Asks Supreme Court to Uphold Anti-Spam Law,* Seattle Times, April 7, 2000.
9. Jethro K. Lieberman, *The Evolving Constitution,* (New York: Random House, 1992) p. 42.
10. *Gibbons v. Ogdens,* 22 U.S. 1 (1824).
11. *Southern Pacific Company v. Arizona,* 325 U.S. 761 (1945).
12. *Pike v. Bruce Church, Inc.,* 397 U.S. 137 (1970).
13. *H.P. Hood and Sons v. DuMond,* 336 U.S. 525 (1949).

Exhibit 41-1. Bally's logo.

FACTS OF *BALLY V. FABER*

Bally Total Fitness (see Exhibit 41-1) is a New York Stock Exchange corporation with its international headquarters in Chicago. Bally is the largest commercial operator of fitness centers in North America, with nearly 4,000,000 members and 360 facilities in 27 states and Canada.[1]

Andrew Faber, a Washington, D.C. photographer and Web designer, had a dispute with Bally. When he could not resolve the dispute to his satisfaction, Faber created and maintained a Web site called "Bally Sucks." The site was devoted to consumer complaints about Bally and contained instructions on how members could cancel their membership.[2] Faber's site encouraged other dissatisfied customers to tell their stories. When a Web surfer visited the site, Bally's distinctive trademark (Exhibit 41-1) appeared with the word "Sucks" superimposed on it. At the bottom of the screen were the words "Bally Total Fitness Complaints! Un-authorized" [sic].

In February 1998, Bally sued Faber in federal court in California. Bally asked that Faber stop using its trademark on his Web site. Bally claimed that Faber's Web site was in violation of laws prohibiting trademark infringement, unfair competition, and trademark dilution. In April, the court denied Bally's motion for a temporary restraining order against Faber.

In November, the court granted Faber's motion for summary judgment against Bally. Summary judgment is a device used by the courts when "there is no genuine issue as to any material fact and … the moving party is entitled to a judgment as a matter of law."[3] By granting the motion for summary judgment, the court held that even if all of Bally's claims were true, they would not prove Bally's case. Bally appealed the lower court's verdict, but the parties agreed to a settlement before the higher court reached a decision. As part of the settlement, Faber removed the Bally Sucks Web site.

AN OVERVIEW OF TRADEMARK LAW

A trademark is a distinctive picture or word that a seller adds to a product to identify its origin and to distinguish the product from other products. Trademark law grants protection to many forms of identification, including:

- Invented words such as Kodak and Exxon
- Distinctive and unique packaging such as the Heinz Ketchup glass bottle
- Unique color combinations (yellow and red for Kodak film)
- Building designs (McDonald's golden arches)
- Unique logos or symbols (the IBM symbol or the red K used by Kellogg's)

In 1946, Congress passed the Lanham Act[4] (the Act) to regulate trademarks. Congress enacted the Act under its constitutional right to regulate interstate commerce.[5] A trademark registered under the Act is given federal protection. Parties may register actual or planned trademarks with the Patent and Trademark Office. If examiners initially approve a trademark, it is published in the Official Gazette of the Trademark Office. This is done to notify other parties pending final approval. A full set of legal options is available to resolve trademark disputes.

The Act also discusses certain marks that may not be legally registered as trademarks. They include:

- Generic or geographic product names. (As an example, "Maine Potatoes" cannot be registered as a trademark by any one person. The phrase does not distinguish one person's product, but describes all potatoes grown in Maine. "Johnson's Maine Potatoes" could be registered as a trademark.)
- The name, portrait, or signature of a living person without his or her consent.
- State or municipal flags.[6]

Although the owner of a trademark is guaranteed "exclusive use" of the trademark, that right has certain limitations. These limitations are known as *fair use* and allow others to use the trademark as a descriptive term. The fair use doctrine allowed the use of Bally's trademark as an exhibit in this chapter. If I wanted to sell my 1985 Chevrolet Celebrity, I could advertise that it is a Chevrolet Celebrity although I did not get permission from General Motors to use the name. A competitor may use another person's registered trademark in a comparison of goods. For example, an ad for Coca-Cola can say that it tastes better than Pepsi Cola, although Pepsi did not authorize the use of its trademark.

People and organizations rely on trademarks to make intelligent decisions about product purchases. According to the Act, infringement has occurred when use of the trademark by another party is "likely to cause confusion, or cause mistakes, or to deceive."[7] The court may issue injunctions, compensate the owner for damages, take away profits from the infringer or award attorney fees.[8] The court may even confiscate and destroy goods with the illegal trademark (a frequent occurrence used against illegal vendors at rock concerts).

The owner of a trademark has the exclusive right to use it on its product and on related products, such as T-shirts and lunchboxes. A recognized and respected trademark can be one of an organization's most valuable assets and often has cash value when the company is sold or liquidated.

Trademark dilution takes place when the unauthorized use of a trademark would reduce its value to the owner. Dilution must be commercial in nature and can occur even when there is no direct business competition between the parties.[9] In one recent case,[10] American Express, the worldwide credit card organization, sued the American Express Limousine Service after the limousine service used the same name. Although there was no competition between the two companies (credit cards and limousine service), the court found that the "defendant's use of the AMERICAN EXPRESS mark would 'whittle away' the distinct quality of plaintiff's mark."

ANALYSIS: TRADEMARK INFRINGEMENT

Bally claimed that Faber's Web site constituted both trademark infringement and trademark dilution. By granting summary judgment, the court held that Faber's actions were not a violation of trademark law, even if all of the charges claimed by Bally were true. According to the Act, the court would have to find that Faber's use of the Bally trademark created a likelihood of confusion.[11] Only then could the court find that trademark infringement had occurred.

A major factor in determining whether there is a likelihood of confusion is the similarity of goods produced by the two parties.[12] The more related

the goods are, the more likely it is that the court will find that trademark infringement actually took place. "Related goods are those goods which, though not identical, are related in the minds of consumers."[13] Courts have considered the following pairs of items to be related goods:

- Shirts and pants[14]
- Beer and whiskey[15]
- Locks and flashlights[16]

Bally and Faber did not market similar goods (health club memberships as opposed to Web page design) so there was little likelihood of confusion through related goods. The court then held that:

> No reasonable consumer comparing Bally's official Web site with Faber's site would assume Faber's site 'to come from the same source or thought to be affiliated with, connected with, or sponsored by, the trademark owner.' Therefore, Bally's claim for trademark infringement fails as a matter of law.[17]

ANALYSIS: TRADEMARK DILUTION

The court also granted Faber's motion for summary judgment against Bally's claim of trademark dilution. To show dilution, the defendant's use of the trademark must lessen the capacity of the plaintiff's trademark to identify and distinguish its goods and services and must be commercial in nature. For a dilution claim, Bally had to show that Faber's use of its famous trademark was commercial in nature. Bally also had to show that Faber's use diluted the value of the trademark by lessening the capacity of the mark to identify and distinguish goods and services.[18]

Faber's use of the Bally trademark was noncommercial. He did not use the trademark for the benefit of his own business, and Bally could not show that Faber's use had tarnished the trademark. Faber's site could not confuse consumers, and the court granted the motion for summary judgment.

RECOMMENDATIONS TO ORGANIZATIONS

For even the most stable organization, gripe sites can be an embarrassing nuisance. Potential customers and employees do look at these sites. www.walmartsucks.com has received over 1,000,000 hits through the past three years. Here are some recommendations that may prevent problems.

Some organizations actually purchase the names of potential gripe sites, thereby making them unavailable for outsiders. For example, Chase Manhattan bought the Web site rights to several Web sites, including IhateChase.com, ChaseStinks.com, ChaseSucks.com, ChaseBlows.com, and several others not appropriate for this journal.[19] That did not stop a disgruntled

customer from setting up his own gripe site at chasebanksucks.com. It may make it more difficult for potential customers to find the gripe site.

Organizations would be wise to read their gripe sites regularly. Because of the freewheeling nature of the Internet, there is no real control over the contents of any Web site. A single unhappy person can spread false rumors that could be detrimental to employee morale or even the corporation's reputation.

Lastly, an organization should keep its perspective on gripe sites. Most gripe sites are simply one unhappy customer or ex-employee letting off steam harmlessly. Most of these sites can be safely ignored, unless they threaten personnel or present confidential documents obtained through an internal security leak.

Dissatisfied customers have the free speech right to criticize organizations publicly on the Internet. *Bally v. Faber* allows individuals to use the trademarked logos on such sites as long as there is no reasonable chance of confusion. As the Internet continues to grow, gripe sites will become more common. Organizations should evaluate these sites and learn from them about their relationship with their customers and employees.

Notes

1. www.ballyfitness.com
2. Andrew Malone, Masters of their domain, the scramble for insulting Web sites, *New York,* June 8, 1998.
3. Fed. R. Civ. P. §56(c).
4. 15 USC §§1051–1127.
5. Article I, Section 8, Clause 3.
6. Mark Warda, *How to Register a United States Trademark,* Sphinx Publishing, Clearwater, Florida, 1988, 10–11.
7. §32[a][1].
8. Steven W. Kopp and Tracey A. Suter, Trademark strategies online: implications for intellectual property protection, *Journal of Public Policy & Marketing,* Spring 2000, 119.
9. J. Thomas McCarthy, *McCarthy on Trademarks and Unfair Competition,* §24:89 at 24-137-38 (1997).
10. *American Express v. American Express Limousine Service,* 772 F. Supp 729 (E.D.N.Y. 1991).
11. 15 USC §1114(1)(a).
12. *Petro Stopping Centers, L.P. v. James River Petroleum, Inc.,* 130 F.3d 88 (4th Cir. 1997).
13. *Levi Strauss & Co. v. Blue Bell, Inc.,* 778 F.2d 1352, 1363 (9th Cir. 1985).
14. Id.
15. *Fleischmann Distilling Corp. v. Maier Brewing Co.,* 314 F.2d 149, 152–53 (9th Cir. 1963).
16. *Yale Electric Co. v. Robertson,* 26 F.2d 972 (2d Cir. 1928).
17. *Bally,* 1163–1164.
18. Note, "Bally Total Fitness Holding Corp. v. Faber," *15 Berkeley Tech. L.J.* 229, 2000.
19. Robert Trigaux, Bank-bashing goes digital at Internet gripe sites, *American Banker,* March 26, 1999, 1.

145

Computer Crime Investigations: Managing a Process without Any Golden Rules

George Wade, CISSP

Security is often viewed as an "after-the-fact" service that sets policy to protect physical and logical assets of the company**.** In the event that a policy is violated, the security organization is charged with making a record of the violation and correcting the circumstances that permitted the violation to occur. Unfortunately, the computer security department (CSD) is usually viewed in the same light and both are considered cost-based services. To change that school of thought, security must become a value-added business partner, providing guidance before and after incidents occur.

The Security Continuum

Each incident can be managed in five phases, with each phase acting as a continuation of the previous phase, and predecessor of the next. Exhibit 145.1 displays the continuum.

Flowing in a clockwise, circular fashion, the security continuum begins with the report of an incident or a request for assistance from the CSD business partner (also known as "the customer"). Strong documentation during this initial report phase is the first building block of an ever-evolving incident response plan. Strong documentation will also be used to determine whether or not an investigation is opened, as not every anomaly requires a full investigation. The report phase flows into the investigative phase where intelligence gathering and monitoring begins and documentation continues. At this point, the CSD investigator (CSDI) should understand and be able to define what has occurred so that a determination can be made to begin a full investigation. The investigative phase will flow into the assessment phase, although there may not be a strong demarcation point. The investigative phase and the assessment phase may run concurrently, depending on the incident. Time spent during the assessment phase is dedicated to determining the current state of the business, identifying additional problem areas, and continued documentation. The assessment phase documentation will provide input into the corrective action phase, with this phase beginning as the investigative phase is completed. Intelligence gained during the investigative and assessment phases of the continuum is used to build the corrective action plan. Execution of the correction action plan can be coordinated with the final steps of the investigative phase, in that system holes are plugged as the suspect is being arrested by law enforcement or interviewed by a CSDI. Following the completion of the four previous phases, the proactive phase can begin. This phase should be used to educate management and the user community about incident particulars.

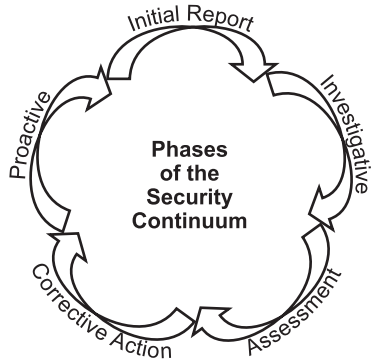


EXHIBIT 145.1 The security continuum.

Education in the form of security awareness presentations will lead to a greater consciousness of the CSD being a value-added business partner that will generate new reports and lead the CSD back into the report phase.

The Initial Report Phase

Before any investigation can begin, the CSD needs to receive a report of an anomaly. One of the best ways to advertise the services of the CSD is through a comprehensive awareness program that includes the methods to report incidents. The CSD should have the ability to receive reports over the phone, via e-mail, and via the World Wide Web (WWW), and each of these methods should permit anonymous reporting. Additionally, the CSD should make the initial report process as painless as possible for the reporter. Because anomalies in computers and networks do not just occur from 9 to 5, convenient 24-hour coverage should be provided. This may be provided by a well-trained guard staff, an internal helpdesk, an external answering service that receives calls after a designated time, or a simple recording that provides a 24-hour reach number such as a pager. It is important that the CSD personnel designated to receive initial reports be well-versed in the structure of the business, have an understanding of common computer terminology, and have excellent customer service skills. They must also understand that all reports must remain confidential because confidentiality is an important aspect of all investigative issues. Without confidentiality, investigative efforts will be hampered, and employees may be wrongfully accused of policy violation or illicit acts.

The CSD Receives an Incident Report

Whether the reports come into the CSD help desk or directly to a CSDI, the same questions need to be asked when receiving the initial report. By asking the "who, what, where, when, why, and how" questions, the CSD trained personnel receiving the initial report should be able to generate a somewhat thorough overview of the anomaly and record this information in a concise, easy-to-read format. An incident is classified as an anomaly at this point because, without initial review, the action or incident may be nothing more than the reporter's misunderstanding of standard business events or practices. The best method to compile and record this information is by using an initial report form (Exhibit 145.2).

Using a form will ensure that each incident is initially handled in the same manner and the same information is recorded. As the types of incidents change, this form can be updated to ensure that the most relevant questions are being asked. It will provide a comprehensive baseline for the CSDI when investigative work begins and can be included as part of the incident case file. Should it be determined during the investigative phase that the anomaly will not be pursued, the form will act as a record of the incident.

An important point to remember is that no question is too trivial to ask. What may seem apparent to CSD personnel may or may not be apparent to the reporter, and vice versa. The person receiving the initial report for the CSD must also be trained to recognize what is and what is not an urgent issue. An urgent issue is a system administrator calling to report watching an unauthorized user peruse and copy files, not a customer calling to report a PC, normally turned off at night, was found on in the morning. Asking key questions and obtaining relevant and pertinent information will accomplish this task.

EXHIBIT 145.2 The Initial Report Form

REPORTER INFORMATION & INITIAL REPORT□

FIRST□ _____ MIDDLE□ _____ LASTR/SR□ _____ ID NUMBER□ _____

FULL ADDRESS/PHONE□ _____

INCIDENT DATE□ _____ INCIDENT TIME□ _____

INCIDENT SUMMARY□ _____

_____

_____

DISCOVERY DATE:□ _____ DISCOVERY TIME:□ _____

STEPS TAKEN:□ _____

_____

_____

_____

_____

SYSTEM NAME:□ _____ IP ADDRESS□ _____

OPERATING SYSTEM:□ _____ VERSION/PATCH No.□ _____

SYSTEM LOCATION□ _____ SA NAME & PHONE□ _____

SUPPORTING DATA:□ _____

CURRENT STATE OF SYSTEM:□ _____

PURPOSE OF MACHINE:□ _____

APPLICATION OWNER:□ _____ PHONE NUMBER:□ _____

APPLICATION USER:□ _____ PHONE NUMBER:□ _____

HOW DID INCIDENT OCCUR:□ _____

WHY DID INCIDENT OCCUR:□ _____

ADDITIONAL INFORMATION:□ _____

ASSIGNED TO:□ _____ ASSIGNMENT NUMBER:□ _____

<Company Name >- Proprietary□

The "who" questions should cover the reporter, witnesses, and victims. The victims are the application owner, user group, and system administrators. Contact information should be obtained for each. The reporter should also be queried as to who has been notified of the incident. This will help the CSDI determine the number of people aware of the issue.

"What" is comprised of two parts: the "anomaly what" and the "environment what." The "anomaly what" should include a description of the conditions that define the anomaly and the reporter's observations. The "environment what" is comprised of questions that identify the operating hardware and software in the impacted environment. Is the system running a UNIX variant such Linux or Solaris, a DOS variant such as Windows 95/98, or Windows NT? The operating system version number should be obtained, as well as the latest release or software patch applied. The reporter should also be queried about the application's value to business. Although all reporters will most likely consider their systems critical, it is important to determine if this is a mission-critical system that will impact revenue stream or shareholder value if the anomaly is confirmed to be a security breech.

The "where" questions cover the location of the incident, the location of the system impacted (these may not be in the same physical location), and the reporter's location. It is very common in logical security incidents that the reporter may be an application user in one location and the system may reside in another location.

"When" should cover the time of discovery and when the reporter suspects the anomaly occurred. This could be the time the system was compromised, a Web page was changed, or data was deleted. If the reporter is utilizing system logs as the basis of the report, the CSD personnel should determine the time zone being used by the system.

"Why" is the reporter's subjective view of the events. By asking why the reporter believes the anomaly occurred, the reporter may provide insight as to ongoing workplace problems, such as layoffs or a disgruntled employee with access to the system. Insight such as this might provide the CDSI with initial investigative direction.

Finally, "how" is the reporter's explanation for how the anomaly occurred. Be sure to ask how the reporter arrived at this conclusion, as this line of questioning will draw out steps the reporter took to parse data. Should the anomaly be confirmed as an incident requiring investigation, these actions would require further understanding and documentation.

When considering logical security incidents, be sure to cover the physical security aspect during the initial report as well. Questions about the physical access to the compromised machine and disaster recovery media (operating system and application data backups) should be covered during the initial report.

The Investigative Phase

Before any monitoring or investigation can take place, the company must set a policy regarding use of business resources. This policy should be broad enough to cover all uses of the resources, yet specific enough so as not to be ambiguous. A policy covering the use of noncompany-owned assets (laptop and desktop computers) should also be considered. This will become important during the evidence-gathering portion of the investigative phase. Once the policies are established, thorough disclosure of the corporate policies must take place. Each employee, contractor, and business partner must be required to read the policies and initial a document indicating that the policy was reviewed, and the document should be kept in the employee's personnel folder. A periodic re-review of the policy should also be required.

In addition to the policy on use of resources, a warning banner should be included in the log-on process and precede access to all systems. The banner should advise the user that activity must adhere to the policy, that activity can be monitored, and any activity deemed illegal can be turned over to law enforcement authorities. The following is an example of a warning message:

> This system is restricted solely to <company name> authorized users for legitimate business purposes only. The actual or attempted unauthorized access, use, or modification of this system is strictly prohibited by <company name>. Unauthorized users are subject to company disciplinary proceedings and/or criminal and civil penalties under state, federal, or other applicable domestic and foreign laws. The use of this system may be monitored and recorded for administrative and security reasons. Anyone accessing this system expressly consents to such monitoring and is advised that if monitoring reveals possible evidence of criminal activity, <company name> might provide the evidence of such activity to law enforcement officials. All users must comply with <company name> Company Instructions regarding the protection of <company name> information and assets.

This warning banner should precede entry into all corporate systems and networks, including stand-alone (nonnetworked) computers and FTP sites. When confronted with the banner, the users should be given the option to exit the log-on process if they do not agree with the policy.

The investigations undertaken by the CSD can be classified into two broad categories: reactive and proactive. Some of the more common reactive reports include unauthorized or suspected unauthorized access to company resources, nonbusiness use of resources, the release of proprietary material, threatening or harassing activity, and activity that creates a hostile work environment. From the reactive cases being generated, the CSD should identify opportunities for prevention of the reactive cases. For example, if the CSD is receiving a large amount of unauthorized access cases, what are the similarities in each? Can a companywide solution be devised and an awareness campaign started to eliminate the vulnerability? Proactive activities can include intelligence-gathering activities such as the monitoring of company access to WWW and newsgroup sites known to hacking tools, offensive, or illegal material. Monitoring of financial message boards may reveal premature proprietary information release or include anticompany postings that are a precursor to workplace violence. Review and monitoring of traffic to free, WWW-based e-mail sites may identify proprietary information being transferred to a competitor. Periodic review of postings to Internet newsgroups could reveal stolen equipment being resold.

Beyond the Initial Report

From the Incident Report Form, the CSDI can begin developing a plan of action. Each investigation will contain two initial steps; anomaly validation and investigation initiation. The first step determines if the anomaly is actually an incident worth investigating. Not every anomaly is the result of a criminal or dishonest act, and not every anomaly warrants a full-scale investigation. An anomaly that presents itself to be unauthorized access to a system with data deletion, may have been an honest mistake caused by the wrong backup tape being loaded. In this instance, a short report of the incident should be recorded. If several similar reports are received in the same area of the company, steps to initiate better data control should be taken. If a Windows 9x system, in an open area, that does not contain sensitive data or support network access is entered, the CSDI must decide if the action justifies full investigation. In this example, it may be prudent to record the incident without further investigative effort and dedicate resources to more mission-critical tasks. Through proactive review of anomaly report records, a decision might be made to conduct an investigation into recurring incidents.

After it is determined that the anomaly requires further investigation, logs supporting the anomaly or logs that may have been altered at the time of the anomaly need to be collected and analyzed. The CSDI must be careful not to view the anomaly with tunnel vision, thereby overlooking important pieces of information. Additionally, more thorough interviews of the reporter and witnesses need to be conducted. These secondary fact-finding interviews will help the CSDI further document what has occurred and what steps the victim or reporter may have taken during the identification of the anomaly. The CSDI should request and obtain from the reporter, and other witnesses, detailed statements of what steps were taken to identify the anomaly. For example, a system administrator (SA) of a UNIX-based system may have examined system logs from the victim system while logged into the victim system using the root ID. In this example, the CSDI should obtain a detailed written statement from the SA, that describes the steps taken and why they were taken. This statement should clearly state why data might have been added or deleted. In addition to the statement, the CSDI should obtain a copy of the shell history file for the ID used, print a copy of the file, and have the SA annotate the listing. The SA's notes should clearly identify which commands were entered during the review and when the commands, to the best of the SA's recollection, were entered. The written statement should be signed by the SA, and placed, along with the annotated version of the shell history, in an investigative case file.

The written statement and data capture (this will be dealt with in more detail later) should be received by the CSDI as soon as possible after the initial report. It is important that witnesses (in this example, the SA) provide written statements while the steps taken are still fresh in their minds. Should it be determined that the anomaly is actually unauthorized activity, the written statements will help to close potential loopholes in any civil or criminal action that may come at the conclusion of the activity.

Intelligence Gathering

It behooves the CSDI to understand as much as possible about the suspect. Understanding the equipment being used, the physical location from which the suspect is initiating the attacks, the time at which the attacks occur, and human factors such as the suspect's persona, all help the CSDI fully understand the tasks at hand.

Initially, the CSDI will want to gather information about the machine being used. By running commands such as nbtstat, ping, trace route, etc., the CSDI can obtain the IP address being used, user ID and machine name being used, and the length of the lease if DHCP is being used.

Following the identification of the machine being used, the CSDI will want to identify where the machine is physically located. If the investigation involves an insider threat, the CSDI could perform physical surveillance on the suspect's office or perform after-business-hours visits to the suspect's office. Before visiting the office, the CSDI should determine the normal business hours at the location, and the ability to gain after-business-hours access. In addition to the physical facility information, the CSDI should determine the type of equipment the suspect utilizes. Once again, if the suspect utilizes a laptop computer to execute the attacks, a late-night visit to the suspect's office may prove fruitless.

If possible, try to gain intelligence about the suspect's work habits in addition to the intelligence gained from the anomalies and initial queries. The suspect may spend the day attacking systems to avoid detection from after-business hours attacks and spend evenings catching up on this work so that management is not aware of his daily activity. In a situation such as this, the CSDI may run into the suspect during a late-night visit. By gathering intelligence, the CSDI can better plan on what equipment will be needed when visiting the suspect's workspace, what actions may need to be taken, and how long the action may take.

No Longer an Anomaly

From the intelligence gathered during the fact-finding interviews and log review, the CSDI should be able to identify the anomaly as an actual incident of unauthorized activity. One of the most important decisions to make while building the action plans is to decide if the activity will be stopped immediately or monitored while additional evidence is gathered. There are several factors to consider before making this decision — most importantly, the impact to the business should the activity continue. The CSDI must be sure that value of identifying the perpetrator outweighs the potential impact to the business. If the CSDI is assured of being able to accurately monitor the activities of the perpetrator, and there is no potential damage such as additional proprietary information being lost or data deleted, the CSDI should proceed with monitoring and build additional evidence. If the perpetrator cannot be controlled or accurately monitored, the activity should be stopped by shutting down the perpetrator's access. In either case, the CSDI must be sure to obtain CSD management approval of the action plan. The selling point to management for continued monitoring is that it buys the CSDI more time to determine what damage may have been done, identify more areas compromised, record new exploits as they occur, and most importantly, identify areas of entry not yet identified.

Active Monitoring

If the activity will be monitored, the first step in the monitoring process is to set up a recording device at the point of entry. If the activity is originating from an office within the CSDI's company, monitoring may consist of a keystroke monitor on the computer being used or a sniffer on the network connection. The traffic captured by the sniffer should be limited to the traffic to and from the machine under electronic surveillance. In addition, video surveillance should be considered — if the environment and law permits. Video surveillance will help confirm the identity of the person sitting at the keyboard. If video surveillance is used, the time on the video recorder should be synchronized to match the time of the system being attacked. Synchronizing the time on the video recorder to that of the system being attacked will confirm that the keystrokes of the person at the keyboard are those reaching the system being attacked. Although this may seem obvious, the attacker could actually be using the machine being monitored as a stepping stone in a series of machines. To do this, the attacker could be in another office and using something as simple as Telnet to access the system in the office being monitored to get to the system being attacked. It is the task of the CSDI to prove that the system attack is originating from the monitored office.

When using video surveillance, the CSDI needs to be aware that the law only permits video — not audio — and that only certain areas can be monitored. Areas that provide a reasonable expectation of privacy, such as a bathroom, cannot be surveyed. Luckily, there are not that many instances of computing environments being set up in bathrooms. Employee offices do not meet that exception and may be surveyed, although the CSDI should only use video surveillance as a means of building evidence during an investigation.

The next step in the monitoring process is to confirm a baseline for the system being attacked. The goal is to identify how the system looked before any changes occur. If the company's disaster recovery plan requires a full system backup once a week, the CSDI, working with the systems administrator (SA), should determine which full backup is most likely not to contain tainted data. This full backup can be used as the baseline. Because the CSDI cannot be expected to understand each system utilized within the company, the CSDI must rely on the SA for assistance. The SA is likely to be the person who knows the system's normal processes and can identify differences between the last-known good backup and the current system. Ideally, the system backup will be loaded on a similar machine so that subtle differences can be noted. However, this is not usually the case. In most instances, the baseline is used for comparison after monitoring has been completed and the attacker repelled.

While monitoring activity, the SA and CSDI should take incremental backups, at a minimum once a day. The incremental backups are then used to confirm changes to the system being attacked on a daily basis.

As the monitoring progresses, the CSDI and the SA should review the captured activity to identify the attacker's targets and methods. As the activity is monitored, the CSDI should begin building spreadsheets and charts to identify the accounts attacked and the methods used to compromise the accounts. The CSDI should also note any accounts of interest that the attacker was unable to compromise. The CSDI must remember that the big picture includes not only what was compromised, but also what was targeted and not compromised.

The Project Plan

In building a picture of the attack, the CSDI should also begin to identify when to begin the assessment, the corrective action phase, when to end monitoring, and when to bring in law enforcement or interview the employee involved. This should be part of the dynamic project plan maintained by the CSDI, and shared with CSD management. As the plan evolves, it is important to get the project plan approved and reapproved as changes are made. Although it is always best to keep those who are knowledgeable or involved in the investigation to a minimum, the CSDI may not be able to make informed decisions about the impact of the unauthorized activity to the victim business unit (BU). With this in mind, the CSDI needs to inform management of the BU impacted by the attack and the company legal team, and keep both apprised of project plan changes. The project plan should include a hierarchy of control for the project, with CSD management at the top of the hierarchy providing support to the CSDI. The CSDI, who controls the investigation will offer options and solutions to the victim BU, and the victim BU will accept or reject the project plan based on its level of comfort.

Legal Considerations

As the investigation progresses, the CSDI should have a good understanding of which laws and company policies may have been violated. Most states now have laws to combat computer crime, but to list them here would take more room than available for this chapter. However, there are several federal laws defined in the United States Code (USC) with which the CSDI should be familiar. Those laws include:

- *18 USC Sec. 1029.* Fraud and related activities in connection with access devices. This covers the production, use, or trafficking in, unauthorized access devices. Examples include passwords gleaned from a targeted computer system. This also provides penalties for violations.
- *18 USC Sec. 1030. The Computer Fraud and Abuse act of 1986.* Fraud and related activity in connection with computers. This covers trespass, computer intrusion, unauthorized access, or exceeding authorized access. It includes and prescribes penalties for violations.
- *The Economic Espionage Act of 1996.* Provides the Department of Justice with sweeping authority to prosecute trade secret theft whether it is in the United States, via the Internet, or outside the United States. This act includes:
  - *18 USC Sec. 1831.* Covers offenses committed while intending or knowing that the offense would benefit a foreign government, foreign instrumentality, or foreign agent.
  - *18 USC Sec. 1832.* Covers copyright and software piracy, specifically those who convert a trade secret to their own benefit or the benefit of others intending or knowing that the offense will injure any owner of the trade secret.
- *The Electronic Communications Privacy Act of 1986.* This act covers the interception or access of wire, oral, and electronic communications. Also included is the unauthorized access of, or intentionally exceeded authorized access, to stored communications. This act includes:
  - *18 USC Sec. 2511.* Interception and disclosure of wire, oral, or electronic communications.
  - *18 USC Sec. 2701.* Unlawful access to stored communications.
- *The No Electronic Theft (NET) Act.* The NET Act amends criminal copyright and trademark provisions in 17 USC and 18 USC. Prior to this act, the government had to prove financial benefit from the activity to prosecute under copyright and trademark laws. This act amended the copyright law so that an individual risks criminal prosecution when there is no direct financial benefit from the reproduction of copyright material. This act is in direct response to *United States v. La Macchia,* 871 F. Supp 535 (D. Mass. 1994), in which an MIT student loaded copyrighted materials onto the Internet and invited others to download this material, free of charge. In *La Macchia,* because the student received no direct financial benefit from his activity, the court held that the criminal provisions of the copyright law did not apply to his infringement.

In addition, those dealing with government computer systems should be familiar with:

- *Public Law 100-235.* The Computer Security Act of 1987. This bill provides for a computer standards program, setting standards for government-wide computer security. It also provides for training of

persons involved in the management, operation, and use of federal computer systems, in security matters.

Evidence collection

Evidence collection must be a very methodical process that is well-documented. Because the CSDI does not know at this point if the incident will result in civil or criminal prosecution, evidence must be collected as if the incident will result in prosecution.

Evidence collection should begin where the anomaly was first noted. If possible, data on the system screen should be captured and a hardcopy and electronic version should be recorded. The hardcopy will provide the starting point in the "series of events" log, a log of activities and events that the CSDI can later use when describing the incident to someone such as a prosecutor, or management making a disciplinary decision. Because CSDIs will be immersed in the investigation from the beginning, they will have a clear picture of the anomaly, the steps taken to verify the anomaly were actually an unauthorized act, the crime committed or policy violated, the actions taken by the suspect, and the damage done. Articulating this event to someone, particularly someone not well-versed in the company's business and who has never used a computer for more than word processing, may be a challenge bigger than the investigation. The series of events log, combined with screen prints, system flows, and charts explaining the accounts and systems compromised and how compromised, will be valuable tools during the education process.

In addition to screen prints, if the system in which the unauthorized access was noted is one of the systems targeted by the suspect or used by the suspect, photographs should be taken. The CSDI should diagram and photograph the room where the equipment was stored to accurately depict the placement of the equipment within the room. Once the room has been photographed, the equipment involved and all of its components should be photographed. The first step is to take close-up photographs of the equipment as it is placed within the room. If possible, photographs of the screen showing the data on the screen should be taken. Be sure to include all peripheral equipment, remembering it may not be physically adjacent to the CPU or monitor. Peripheral equipment may include a printer, scanner, microphone, storage units, and an uninterrupted power supply component. Bear in mind that with the advent of wireless components, not all components may be physically connected to the CPU. The next step is to photograph the wires connected to the CPU. Photographs should include a close-up to allow for clear identification of the ports being used on the machine. The power supply should also be included.

Once the equipment has been photographed, attention should turn to the surrounding area. Assuming one has permission to search the office, one should begin looking for evidence of the activity. It is important to note the location of diskettes and other storage media in relation to the CPU. Careful review of the desktop may reveal a list of compromised IDs or systems attacked, file lists from systems compromised, printouts of data from files compromised, and notes of the activity. Each of these items should be photographed as they are located.

Confiscating and Logging Evidence

After the items have been located, evidence collection should begin. It is important for the CSDI to be familiar with the types of equipment owned and leased by the company. If the CSDI is presented with a machine that is not standard issue, the CSDI must consider the possibility that the machine is privately owned. Without a policy covering privately owned equipment and signed by the employee, the CSDI can not search or confiscate the machine without permission from the owner. Once the CSDI has confirmed company ownership, evidence collection may begin. For each piece of evidence collected, the CSDI needs to identify where and when it was obtained and from whom it was obtained. The best method to accomplish this is a form used to track evidence (Exhibit 145.3).

As the evidence is collected, the CSDI will fill out the form and identify each item using serial numbers and model numbers, if applicable, and list unique features such as scratch marks on a CPU case. Each item should be marked, if possible, with the CSDI's initials, the date the item was collected, and the case number. If it is not possible to mark each item, then each item should be placed in a container and the container sealed with evidence tape. The evidence tape used should not allow for easy removal without breakage. The CSDI should then sign and date the evidence tape. The CSDI should always mark evidence in the same manner because the

EXHIBIT 145.3 The Evidence Form

Evidence/Property Custody Document				
District/Office:			Serial Number:	
Location:			Investigator Assigned To:	
Name and Title of Person from whom received Owner Other			Investigator's Address (include zip code)	
Address from where obtained (including zip code)			Reason Obtained	Date:
Item No.	Quantity	Description of Article(s) *(Include model, serial number, condition and unusual marks or scratches)*		
		CHAIN OF CUSTODY		
Item No.	Date	Released By	Received By	Purpose of Change of Custody
		SIGNATURE NAME,GRADE OR TITLE	SIGNATURE NAME,GRADE OR TITLE	
		SIGNATURE NAME,GRADE OR TITLE	SIGNATURE NAME,GRADE OR TITLE	
		SIGNATURE NAME,GRADE OR TITLE	SIGNATURE NAME,GRADE OR TITLE	
		SIGNATURE NAME,GRADE OR TITLE	SIGNATURE NAME,GRADE OR TITLE	

<Company> - Proprietary

CSDI may be asked to testify that he is the person identified by the marking. By marking items in the same fashion, the CSDI can easily identify where the markings were placed.

Evidence Storage

After the evidence has been collected, it must be transported to and stored in a secure location. During transport, special care must be taken to ensure that custody can be demonstrated from the point of departure until the evidence arrives at, and is logged into, the storage facility. While in custody of the CSD, the evidence must be protected from damage caused by heat, cold, water, fire, magnetic fields, and excessive vibration. Hard drives should be stored in static-free bags and packed in static-free packaging within the storage container. The CSD must take every precaution to ensure the evidence is protected for successful prosecution and eventual return to the owner. Should the confiscated items be damaged during transport, storage, or examination, the owner of the material may hold the CSD liable for the damage.

Evidence Custodian

When the evidence arrives at the storage location, it is preferable that an evidence custodian logs it into the facility. It will be the job of the evidence custodian to ensure safe storage for the material as described above. Using an evidence custodian, as opposed to each CSDI storing evidence from their cases, ensures that the evidence, property owned by others until the case is adjudicated, is managed with a set of checks and balances. The evidence custodian will be responsible for confirming receipt of evidence, release of evidence, and periodic inventory of items in evidence. After the case has been adjudicated, the evidence will need to be removed from evidence storage and returned to the owner. The evidence form should then be stored with the case file.

Business Continuity during Evidence Collection

The CSDI must remember that his responsibility is to the company and shareholders. The CSDI must find the balance between performing an investigation and protecting the business, thereby maintaining shareholder value. If the unauthorized activity required the computer be shut down during the length of an investigation, then an attacker need not gain entry and destroy files if the purpose of the attack is to disrupt business. Simply causing a machine to reboot or drop a connection would, in itself, be enough to disrupt the business.

When an investigation requires the CDSI to obtain evidence from a computer's hard drive or from drives that support a network, the CSDI cannot stop the business for an extended period of time by placing the hard drive into evidence. By performing a forensic backup of the hard drives in question, the CSDI can ensure evidence preservation and allow the business to get up and running in a short amount of time. A forensic image of a hard drive preserves not only the allocated file space, but also the deleted files, swap space, and slack space. The forensic image is the optimal answer to gathering evidence. Once a forensic image has been obtained, a new disk can be placed in the target computer and data loaded from a backup. If data loss is a concern, then the forensic image can be restored to the new disk, allowing the business to proceed as the investigation continues.

Although it is not recommended, the CSDI may not be able to stop the business long enough for a forensic image to be taken. In situations involving a system that cannot be brought down (for example, a production control systems or systems that accept customer orders), the CSDI may be presented with the task of gathering evidence while the system is continuing to process data. In situations such as these, the CSDI may be able to gather some evidence by attaching removable storage media to the machine and copying pertinent files to the removable media. In these situations, the CSDI must remember that the data gathered is not the best evidence to prosecute the case. However, just because the evidence may not be optimal for prosecution, it should not be overlooked. Evidence such as this may be used to support the CSDI's theories and may provide the CSDI with insight to other unauthorized activities not identified thus far.

Gathering Evidence through Forensic Imaging

This section provides a cursory overview of forensic imaging. Forensic imaging of a hard drive is a subject deserving a chapter in itself, so this section only attempts to provide the CSDI with an overview of what steps are taken and what equipment is needed to produce a forensic image.

Once the computer has been accurately photographed, the system can be removed to an area where the forensic image will be made or the CPU box opened so that a forensic image can be taken on site. One problem with performing an on-site image is that without an evidence review machine on hand, in which to load and review the forensic image, the CSDI must trust that the image was successful. Assuming removal of the machine would not compromise the investigation, it is best to remove the machine to an examination area. Once in the examination area, a forensic image can be obtained using a DOS boot diskette, forensic imaging software, and tape backup unit. The suspect machine will be booted using the DOS diskette to ensure that no advanced operating system software tools are loaded. The forensic imaging software (there are many packages on the consumer market) is loaded and run from DOS. Output is then directed to the tape backup unit via the system's SCSI port.

In systems without a SCSI port, the hard drive (called the original drive or suspect drive) will have to be removed and installed as a slave drive in another computer. This exercise should not be taken lightly, as there is much opportunity to damage the suspect's drive and lose or overwrite data. In situations such as this, the equipment used to obtain an image may vary; but in all cases, the target for the image must be as large or

larger than the original disk. Targets for the image may be either magnetic tape or a second hard drive. The first step in creating the image is to physically access the original drive and remove it from the system housing. Next, the original drive must be connected to a secondary machine, preferably as a slave drive. Once this original drive has been connected to the secondary machine, the data can be copied from the slave drive to the backup media.

As electronics get smaller, laptop computers present challenges that are unique in, and of, themselves. When performing a forensic image of a laptop computer hard drive that does not provide a SCSI port or PCMCIA adapter access, special interface cables are needed to ensure power to the original drive and data connectivity from the original to the imaging media. If a PCMCIA socket is available, special adapter cards can be obtained to allow the data transfer through the socket to a SCSI device. In this case, drivers for the PCMCIA card are loaded, in addition to the DOS and imaging software.

Once the forensic image has been obtained, the acquired data needs to be reviewed. There are several commercially available packages on the consumer market that support forensic data review. There are also shareware tools available that claim to perform forensic image review without data alteration. It is best to use a package purchased from a company that has a history of providing expert testimony in court about the integrity of its product. The CSD does not want an investigation challenged in court due to evidence gathering and review methods. Unless a vendor is willing to provide expert testimony as to the technical capabilities of its program, the CSD would be well-advised to steer away from that vendor.

During a review of the acquired hard drive, efforts should be made to recover deleted files, examine slack space, swap space, and temporary files. It is not uncommon for evidence of the unauthorized activity to be found in these areas. Additionally, files with innocuous names should be verified as being unaltered by, or in support of, the unauthorized activity. There are some commercially available products on the market that provide hash values for the more commonly used programs. This will allow the CSDI to automate a search for altered files by identifying those that do not match the hash.

Law Enforcement Now or Later?

Throughout the investigation, the CSDI must continually weigh the options and advantages of involving law enforcement in the investigation. There are several advantages and disadvantages to bringing in law enforcement and there is no golden rule as to when law enforcement should be contacted. Although cases involving outsider threats are a little more apparent, insider threat cases are not as obvious.

When law enforcement is brought into an investigation, the dynamics of that investigation change. Although the CSDI can control information dissemination prior to law enforcement involvement, once law enforcement becomes involved, the CSDI no longer has control due to the Freedom of Information Act. Unless the law enforcement agency can prove the need to seal case information, for reasons such as imminent loss of life due to the information release, they do not have the ability to seal the case once arrests have been made. If law enforcement is being brought into an investigation, the CSDI must notify the company's public relations team as soon as possible. Additionally, any steps taken by the CSDI after law enforcement enters the case could be a violation of the Fourth Amendment to the Constitution of the United States. For example, during an insider threat case, the CSDI would normally search the suspect's office for evidence as part of the normal course of the investigation. Because the CSDI is not a sworn law enforcement officer and an employee of the company, the CSDI is permitted by law to conduct the search and not subject to the rules and laws governing search and seizure. However, this does not hold true when:

- The CSDI performs a search in which law enforcement would have needed a search warrant to conduct
- The CSDI performs that search to assist law enforcement
- Law enforcement is aware of the CSDI's actions and does not object to them

When the above conditions are true, the CSDI is acting as an agent of law enforcement and is in violation of the Fourth Amendment.

As stated above, outsider threat cases will not amount to much unless outside assistance through the courts or law enforcement is sought. The most direct way to receive assistance is to contact law enforcement in the event the anomaly can be proven to be intentional and provide them with evidence of the activity. Law enforcement has the power to subpoena business records from Internet service providers (ISPs), telephone companies, etc. in support of their investigation. A less-used tactic is for the CSDI's company to begin a third-party, "John Doe" lawsuit to assist the company in identifying the suspect. These civil remedies will allow the

CSDI to gather information not normally available. For example, the anomaly detected was confirmed as unauthorized access from a local ISP known to the CSDI as the ISP utilized by an employee under suspicion. By filing the lawsuit, the company and the CSDI will be able to obtain subscriber information not normally available. The CSDI needs to be aware that some ISPs will inform the user when a subpoena from a lawsuit is received.

Regardless of when the CSDI chooses to bring law enforcement into the investigation, it should not be the first meeting between the CSDI and law enforcement agent. It is important for the CSDI to establish ties with local, state, and various branches of federal law enforcement (FBI, Secret Service, Customs, etc.) before incidents occur. One of the best methods to establish the relationship early is by participating in training offered by professional service organizations such as the American Society of Industrial Security (www.asisonline.org) and the High Technology Crime Investigation Association (www.htcia.org). Both international organizations not only provide training, but also provide important networking opportunities before incidents occur.

Assessment Phase

The assessment is the phase where the CSDI knows, or has an idea of what has been done, but needs to determine what other vulnerabilities exist. The assessment phase helps reduce investigative tunnel vision by providing the CSDI with insight as to additional vulnerabilities or changes that may have been made. The assessment phase can run in conjunction with an active investigation and should be run as soon as possible after the unauthorized activity is defined. An exception to this is when active monitoring and recording of the activity is taking place. There are two reasons for this. First, the attacker is already in the company's system so one does not want the attacker to see processes running that would not normally be run. These new processes might give the attacker insight as to other system vulnerabilities or alert the attacker to the investigation. Second, the CSDI needs to be able to distinguish between the vulnerability tests performed by the automated process and the tests performed by the attacker. Once it is determined safe to execute the test, the automated tools should be run and the results removed from the system immediately.

Closing the Investigation

One of the largest management challenges during a computer-related incident is bringing the investigation to a close when a suspect has been identified. The CSDI must orchestrate a plan that might include the participation of law enforcement, systems administrators, BU management, public relations, and legal departments.

By now, the decision has most likely been made to pursue criminal or civil charges, or handle the incident internally. Aiding in this decision will be the amount of damage done and potential business loss, as quantified by high-level management in the victim BU.

The Interview

One of the questions that should be paramount in the CSDI mind is why the suspect engaged in the unauthorized activity. This question can frequently be answered during an interview of the suspect. If involved, law enforcement personnel will usually work with the CSDI to ensure that their questions and the CSDI questions are answered during the interview. If law enforcement is not involved, then it is up to the CSDI to interview the suspect and obtain answers to some very important questions. Other than why the suspect took the actions, the CSDI will want to have the suspect explain the steps taken to perform the unauthorized activity, actions taken before and after the unauthorized activity was noted and reported to the CSD, and what additional unauthorized activity may have occurred. For example, if the activity was unauthorized access to a system, the CSDI should have the suspect explain when the access was first attempted, when access was accomplished, what accounts were accessed, and how the system was accessed. The CSDI should have the suspect identify any changes made to the system (i.e., modified data, deleted data, backdoors planted, etc.), and what gains were achieved as a result of the activity. During the interview, the CSDI should not make any promises as to the outcome of the suspect's employment or potential for criminal or civil prosecution, unless first concurring with CSD management and the company legal team. The CSDI should strive for the suspect to detail the discussion in a written statement and sign and date the statement at the completion of the interview. The CSD should utilize a standard form for written statements that includes a phrase about the company being allowed to use the written statement as the company sees fit and that no promises are made in exchange for the written

statement. This will ensure that the suspect does not later attempt to say that any employment promises were made in exchange for the written statement or that the suspect was promised the statement would not be used in disciplinary, criminal, or civil proceedings.

The Corrective Action Phase

After the assessment has been completed, the corrective action phase can begin. This phase should be coordinated with investigative efforts so as not to interrupt any final investigative details. Optimally, the corrective action phase begins as the suspect is being arrested by law enforcement or interviewed by the CSDI. Once it has been determined that the phases can run concurrently or the investigative efforts have been completed, the target machines should be brought down and a forensic image should be acquired. After a forensic image of the machine is acquired, the operating system should be loaded from original disks and all software patches applied. If possible, all user IDs should be verified in writing. If this is not possible, all user passwords should be changed and all users forced to change their passwords at next log-on. Careful documentation should be kept to identify those IDs not used within a selected timeframe; for example, 30 days from the time the system is reloaded. Any ID not claimed by a user should be documented and removed from the system. This documentation should be kept as a supplement to the investigative case file in the event it is determined that the unclaimed ID was a product of the attacker's work. The CSDI should note any attempted use of any unclaimed IDs. If a suspect has been identified and either arrested or blocked from the system, attempted use of one of the unclaimed IDs may indicate a further problem not previously identified.

The validity of application programs and user data is at best a shot in the dark, unless the CSDI and system administrator can identify the date the system was compromised. To be absolutely sure backdoors placed by the attacker are not reloaded, BU management may have to fall back to a copy of the last application software load to ensure future system security.

Once the system has been restored and before it is brought back online, a full automated assessment should be run once again to identify any existing vulnerability. Any vulnerability identified should be corrected or, if not corrected, identified and signed off on as an acceptable risk by the BU manager. After all vulnerabilities have been corrected or identified as acceptable risks, the victim system can once again be brought back online.

Proactive Phase

After the investigation and corrective phases have been completed, a post-mortem meeting should be conducted to initiate the proactive phase. Problems encountered, root cause determination, and lessons learned from the incident should be documented in this meeting. The meeting should be led by the CSDI and attended by all company personnel involved in the incident. If the CSDI can show cost savings or recovered loss, these facts should be documented and provided to management. An overview of the incident and the lessons learned should be incorporated into the CSD security awareness presentations and presented to employees throughout the company. Timely reporting of incidents to the CSD should be stressed during the presentations. As this incident and others are presented throughout the company, the CSD is advertised as a value-added business partner, thereby generating more business for the CSD.

Summary

Although there are no golden rules to follow when investigating computer crime, following a structured methodology during investigations will provide a means for the CSDI to guarantee thorough investigations. Using the security continuum as a shell for a dynamic project plan, the CSDI will ensure a comprehensive examination of each incident. A strong project plan, coupled with traditional investigative skills and a good understanding of forensics and emerging technology, will provide the CSDI with the tools needed to confront an ever-changing investigative environment.

147

Operational Forensics

Michael J. Corby, CISSP

The increased complexities of computer systems today make it difficult to determine what has happened when a malfunction occurs or a system crashes**.** Sometimes, it is difficult to even make the basic identification of whether the cause was accidental or intentional. If the cause was intentional, legal action may be in order; if the cause was operational, the reason must be identified and corrected. Both require a planned and measured response.

Unfortunately, with today's emphasis on immediate recovery in the networked environment, and with the obligation to get back online as quickly as possible, determining the cause may be impossible. The tendency to restart, or reboot, may remove information that could be valuable in ascertaining cause or providing evidence of criminal wrongdoing.

Operational forensics is a two-phased approach to resolving this problem. The first phase is the proper collection of operational information such as data logs, system monitoring, and evidence-tracking methods. The appropriate attention to this phase makes it much easier to identify the problem in the second phase, the recovery.

At recovery time, the information at hand can be used to decide whether a formal intrusion investigation needs to be initiated and evidence collected needs to be preserved. By responding in prescribed ways, which can include repair/replacment of the equipment, correction of a software weakness, or identification of human-caused error(s) that resulted in the disruption, the system can be returned to operation with a much reduced probability of the same event occurring in the future.

Related Business Requirements

Technology has been more than an efficiency enhancement to the organization. It has become the lifeblood of the successful enterprise and the sole product of the networked application service provider. As such, the maximum availability of this essential resource is critical. When a failure occurs or the system is not operating at expected levels, proper procedures should be used to accurately identify and correct the situation. Failing to do so will result in unpredictable operations, inefficiencies and possibly lost revenue, tarnished image, and failure to thrive. The business case for investing in the time, procedures, and the relatively small cost of computer hardware or software components seems clear.

Why then, do companies not have operational forensics (or the same functions by other names) programs in place? Well, for two reasons: People have started with the assumption that computers are perfectly reliable and therefore will only fail under rare circumstances if programs are well-written. Why waste resources in pointing the finger at something that should never occur? Second, the topic of methodical, procedural investigations is new to other than law enforcement, and only recently has come into the foreground with the advent of computer crimes, cyber terrorism, and the relationship of vengeance and violence linked to some computer "chat rooms," e-mail, and personal private data intrusions.

The good news is that operational forensics is not an expensive option. There is some additional cost needed to properly equip the systems and the process for secure log creation; but unless the need is determined for a full-scale criminal investigation and trial preparation, the process is almost transparent to most operations.

The business objectives of implementing an operational forensics program are threefold:

1. Maintain maximum system availability (99.999 percent or five-nines "uptime").
2. Quickly restore system operations without losing information related to the interruption.
3. Preserve all information that may be needed as evidence, in an acceptable legal form, should court action be warranted.

The acceptable legal form is what calls for the operational forensics process to be rigorously controlled through standard methods and a coordinated effort by areas outside the traditional IT organization.

Justification Options

The frequent reaction to a request to start an operational forensics program is one of financial concerns. Many stories abound of how forensic investigations of computer crimes have required hundreds or thousands of hours of highly paid investigators pouring over disk drives with a fine-tooth comb — all of this while the business operation is at a standstill. These stories probably have indeed occurred, but the reason they were so disruptive, took so long, or cost so much, was because the operational data or evidence had to be reconstructed. Often, this reconstruction process is difficult and may be effectively challenged in a legal case if not prepared perfectly.

Operational forensics programs can be justified using the age-old 80-20 rule: an investigation cost is 80 percent comprised of recreating lost data and 20 percent actually investigating. An effective operational forensics program nearly eliminates the 80 percent data recreation cost.

A second way in which operational forensics programs have been justified is as a positive closed-loop feedback system for making sure that the investment in IT is effectively utilized. It is wise investment planning and prudent loss reduction. For example, an operational forensics program can quickly and easily determine that the cause of a server crashing frequently is due to an unstable power source, not an improperly configured operating system. A power problem can be resolved for a few hundred dollars, whereas the reinstallation of a new operating system with all options can take several days of expensive staff time, and actually solve nothing.

No matter how the program is justified, organizations are beginning to think about the investment in technology and the huge emphasis on continuous availability, and a finding ways to convince management that a plan for identifying and investigating causes of system problems is a worthwhile endeavor.

Basics of Operational Forensics

Operational forensics includes developing procedures and communicating methods of response so that all flexibility to recover more data or make legal or strategic decisions is preserved. Briefly stated, all the procedures in the world and all the smart investigators that can be found cannot reverse the course of events once they have been put into action. If the Ctrl-Alt-Delete sequence has been started, data lost in that action is difficult and expensive, if not impossible to recover. Operational forensics, therefore, starts with a state of mind. That state of mind prescribes a "think before reacting" mentality. The following are the basic components of the preparation process that accompany that mentality.

For all situations:

- Definition of the process to prioritize the three key actions when an event occurs:
  - — Evidence retention
  - — System recovery
  - — Cause identification
- Guidelines that provide assistance in identifying whether an intrusion has occurred and if it was intentional
- Methods for developing cost-effective investigative methods and recovery solutions
- Maintenance of a secure, provable evidentiary chain of custody

For situations where legal action is warranted:

- Identification or development of professionally trained forensic specialists and interviewers/interrogators, as needed
- Procedures for coordination and referral of unauthorized intrusions and activity to law enforcement and prosecution, as necessary
- Guidelines to assist in ongoing communication with legal representatives, prosecutors, and law enforcement, as necessary
- Instructions for providing testimony, as needed

Notice that the evidence is collected and maintained in a form suitable for use in cases where legal action is possible, even if the event is purely an operational failure. That way, if after the research begins, it is determined that what was thought initially to be operational, turns out to warrant legal action, all the evidence is available.

Consider the following scenario. A Web server has stopped functioning, and upon initial determination, evidence shows that the building had a power outage and when the server rebooted upon restoration, a diskette was left in the drive from a previous software installation. Initial actions in response include purchasing a new UPS (uninterruptable power supply) capable of keeping the server functioning for a longer time, and changing the boot sequence so that a diskette in the drive will not prevent system recovery. All set? Everybody thinks so, until a few days after the recovery, someone has discovered that new operating parameters have taken effect, allowing an intruder to install a "trap door" into the operating system. That change would take effect only after the system rebooted. Is the data still available to identify how the trap door was installed, whether it posed problems prior to this event, and who is responsible for this act of vandalism?

An operational forensics program is designed to identify the risk of changes to the system operation when it is rebooted and conduct baseline quality control, but also to preserve the evidence in a suitable place and manner so that a future investigation can begin if new facts are uncovered.

Building the Operational Forensics Program

Policy

To start building an operational forensics program, the first key element, as in many other technical programs, includes defining a policy. Success in developing this process must be established at the top levels of the organization. Therefore, a policy endorsed by senior management must be written and distributed to the entire organization. This policy both informs and guides.

This policy informs everyone that the organization has corporate endorsement to use appropriate methods to ensure long-term operational stability, and thus ensure that the means to accurately identify and correct problems will be used. It should also inform the organization that methods will be used to take legal action against those who attempt to corrupt, invade, or misuse the technology put in place to accomplish the organization's mission. There is a subtle hint here meant to discourage employees who may be tempted to use the system for questionable purposes (harassing, threatening, or illegal correspondence and actions), that the organization has the means and intent to prosecute violators.

The policy guides in that it describes what to do, under what circumstances, and how to evaluate the results. With this policy, the staff responsible for operating the system components, including mainframes, servers, and even workstations, as well as all other peripherals, will have a definition of the process to prioritize the three key actions when an event occurs:

1. Evidence retention
2. System recovery
3. Cause identification

In general, this policy defines a priority used for establishing irrefutable data that identifies the cause of an interruption. That priority is to first ensure that the evidence is retained; then recover the system operation; and, finally, as time and talent permits, identify the cause.

Guidelines

As a supplement to these policies, guidelines can be developed that provide assistance in identifying whether an intrusion has occurred and if it was intentional. As with all guidelines, this is not a specific set of definitive rules, but rather a checklist of things to consider when conducting an initial response. More detailed guidelines are also provided in the form of a reminder checklist of the process used to secure a site for proper evidence retention. The suggested method for publishing this guideline is to post it on the wall near a server, firewall, or other critical component. Items on this reminder checklist can be constructed to fit the specific installation, but typical entries can include:

Before rebooting this server:

1. Take a photograph of the screen (call Ext xxxx for camera).
2. Verify that the keyboard/monitor switches are set correctly.
3. Record the condition of any lights/indicators.
4. Use the procedure entitled "*Disabling the disk mirror.*"
5. …
6. …
7. etc.

Accompanying these posted instructions are a series of checklists designed to help record and control the information that can be collected throughout the data collection process.

Log Procedures

Policies and guidelines can help provide people with the motivation and method to act thoughtfully and properly when responding to an event, but they are insufficient by themselves to provide all that is needed. Most operating system components and access software (modem drivers, LAN traffic, Internet access software, etc.) provide for log files to be created when the connection is used, changed, or when errors occur. The catch is that usually these logs are not enabled when the component is installed. Furthermore, the log file may be configured to reside on a system device that gets reset when the system restarts. To properly enable these logs, they must be:

- Activated when the service is installed
- Maintained on a safe device, protected from unauthorized viewing or alteration
- Set to record continuously despite system reboots

Additional third-party access management and control logs can and should be implemented to completely record and report system use in a manner acceptable for use as legal evidence. This includes data that can be independently corroborated, non-repudiated, and chain-of-custody maintained.

Configuration Planning

The operational forensics program also includes defining methods for maximizing the data/evidence collection abilities while providing for fast and effective system recovery. That often can be accomplished by planning for operational forensics when system components are configured. One technique often used is to provide a form of disk mirroring on all devices where log files are stored. The intent is to capture data as it exists as close as possible to the event. By maintaining mirrored disks, the "mirror" can be disabled and removed for evidence preservation while the system is restarted. This accomplishes the preservation of evidence and quick recovery required in a critical system.

The process for maintaining and preserving this data is then to create a minimum of three copies of the mirrored data:

1. One copy to be signed and sealed in an evidence locker pending legal action (if warranted)
2. One copy to be used as a control copy for evidence/data testing and analysis
3. One copy to be provided to opposing attorney in the discovery phase, if a criminal investigation proceeds

Linking Operational Forensics to Criminal Investigation

The value of a well-designed operational forensics program is in its ability to have all the evidence necessary to effectively develop a criminal investigation. By far, the most intensive activity in preparing for a legal opportunity is in the preparation of data that is validated and provable in legal proceedings. Three concepts are important to understanding this capacity:

1. Evidence corroboration
2. Non-repudiation
3. Preservation of the chain of custody

Evidence Corroboration

If one is at all familiar with any type of legal proceeding, from the high profile trials of the 1990s to the courtroom-based movies, television programs, or pseudo-legal entertainment of judicial civil cases, evidence that is not validated through some independent means may by inadmissible. Therefore, to provide the maximum potential for critical evidence to be admitted into the record, it should be corroborated through some other means. Therefore, based on the potential for legal action, several log creation utilities can be employed to record the same type of information. When two sources are compared, the accuracy of the data being reported can be assured. For example, access to a system from the outside reported only by a modem log may be questioned that the data was erroneous. However, if the same information is validated by access to the system from system login attempt, or from an application use log, the data is more likely to be admitted as accurate.

Non-Repudiation

A second crucial element necessary for a smooth legal process is establishing evidence in a way that actions cannot be denied by the suspect. This is called "non-repudiation." In many recent cases of attempted system intrusion, a likely suspect has been exonerated by testifying that it could not have been his actions that caused the violation. Perhaps someone masqueraded as him, or perhaps his password was compromised, etc. There is no way to definitely make all transactions pass the non-repudiation test; but in establishing the secure procedures for authenticating all who access the system, non-repudiation should be included as a high-priority requirement.

Preservation of the Chain of Custody

Finally, the last and perhaps most important legal objective of operational forensics is to preserve the chain of custody. In simple terms, this means that the data/evidence was always under the control of an independent source and that it could not have been altered to support one side of the case. This is perhaps the most easily established legal criterion, but the least frequently followed. To establish a proper chain of custody, all data must be properly signed-in and signed-out using approved procedures, and any chance of its alteration must be eliminated — to a legal certainty. Technology has come to the rescue with devices such as read-only CDs, but there are also some low-technology solutions like evidence lockers, instant photography, and voice recorders to track activity related to obtaining, storing, and preserving data.

For all legal issues, it is wise and highly recommended that the organization's legal counsel be included on the forensic team, and if possible, a representative from the local law enforcement agency's (Attorney General, Prosecutor or FBI/state/local police unit) high-tech crime unit. In the case of properly collecting evidence when and if a situation arises, prior planning and preparation is always a good investment.

Linking Operational Forensics to Business Continuity Planning

What makes operational forensics an entity unto itself is the ability to use the time and effort spent in planning for benefits other than prosecuting criminals. The key benefit is in an organization's ability to learn something

from every operational miscue. Countless times, systems stop running because intruders who only partially succeed at gaining access have corrupted the network connections. In most instances, all the information that could have been used to close access vulnerabilities goes away with the Ctrl-Alt-Delete keys. Systems do not crash without cause. If each cause were evaluated, many of them could be eliminated or their probability of reoccurring significantly reduced.

In the current age of continuous availability, maximum network uptime is directly linked to profit or effectiveness. Implementing an operational forensics program can help establish an effective link to business continuity planning risk reduction and can raise the bar of attainable service levels.

Although evidence collected for improving availability does not need to pass all legal hurdles, an effective method of cause identification can help focus the cost of prevention on *real* vulnerabilities, not on the whole universe of possibilities, no matter how remote. Cost justification of new availability features is more readily available, and IT can begin to function more like a well-defined business function than a "black art."

Summary and Conclusion

When a system interruption occurs, operational forensics is a key component of the recovery process and should be utilized to identify the nature and cause of the interruption as well as collecting, preserving, and evaluating the evidence. This special investigation function is essential because it is often difficult to conclusively determine the nature, source, and responsibility for the system interruption. As such, to improve the likelihood of successfully recovering from a system interruption, certain related integral services, such as establishing the data/activity logs, monitoring system, evidence collection mechanisms, intrusion management, and investigative management should be established prior to a system interruptions occurrence. This is the primary benefit of operational forensics. One will see much more of this in the near future.

Computer Crime Investigation and Computer Forensics

Thomas Welch

Incidents of computer-related crime and telecommunications fraud have increased dramatically over the past decade, but due to the esoteric nature of this crime there have been very few prosecutions and even fewer convictions. The same technology that has allowed for the advancement and automation of many business processes has also opened to the door to many new forms of computer abuse. While some of these system attacks merely use contemporary methods to commit older, more familiar types of crime, others involve the use of completely new forms of criminal activity that have evolved along with the technology.

Computer crime investigation and computer forensics are also evolving sciences which are affected by many external factors: continued advancements in technology, societal issues, legal issues, etc. There are many gray areas that need to be sorted out and tested through the courts. Until then, the system attackers will have a clear advantage and computer abuse will continue to increase. We, as computer security practitioners, must be aware of the myriad of technological and legal issues that affect our systems and its users, including issues dealing with investigations and enforcement.

This chapter will take the security practitioner and investigator through each of the areas of computer crime investigation and computer forensics, so that they are better prepared to respond to both internal and external attacks.

COMPUTER CRIME

According to the American Heritage Dictionary a "crime" is any act committed or omitted in violation of the law. This definition causes a perplexing problem for law enforcement when dealing with computer-related crime, since much of today's computer-related crime is without violation of any formal law. This may seem be a contradictory statement, but traditional criminal statutes, in most states, have only been modified throughout the years to reflect the theories of modern criminal justice. These laws generally envision applications to situations involving traditional types of criminal activity, such as burglary, larceny, fraud, etc. Unfortunately, the modern criminal has kept pace with the vast advancements in technology and he has found ways to apply such innovations as the computer to his criminal ventures. Unknowingly and probably unintentionally, he has also revealed the difficulties in applying older traditional laws to situations involving "computer related crimes."

In 1979 the United States Department of Justice established a definition for "computer crime," stating that "a computer crime is any illegal act for which knowledge of computer technology is essential for its perpetration, investigation, or prosecution." This definition was too broad and has since been further refined by new or modified, state and federal criminal statutes.

Criminal Law

Criminal law identifies a crime as being a wrong against society. Even if an individual is victimized, under the law, society is the victim. A conviction under criminal law normally results in a jail term or probation for the defendant. It could also result in a financial award to the victim as restitution for the crime. The main purpose for prosecuting under criminal law is punishment for the offender. This punishment is also meant to serve as a deterrent against future crime. The deterrent aspect of punishment only works if the punishment is severe enough to discourage further criminal activity. This is certainly not the case in the United States, where very few computer criminals ever go to jail. In other areas of the world there are very strong deterrents. For example, in China in 1995, a computer hacker was executed after being found guilty of embezzling $200,000 from a National bank. This certainly will have a dissuading value for other hackers in China!

To be found guilty of a criminal offense under criminal law, the jury must believe, beyond a reasonable doubt, that the offender is guilty of the offense. The lack of technical expertise, combined with the many confusing questions posed by the defense attorney, may cause doubt for many jury members, thus rendering a "not guilty" decision. The only short-term solution to this problem, is to provide simple testimony in layman terms and to use demonstrative evidence whenever possible. Even with this, it will be difficult for many juries to return a guilty verdict.

Criminal conduct is broken down into two classifications depending on severity. A felony is the more serious of the two, normally resulting in a jail term of more than one year. Misdemeanors are normally punishable by a fine or a jail sentence of less than a year. It is important to understand that if we wish to deter future attacks, we must push for the stricter sentencing, which only occurs under the felonious classification. The type of attack and/or the total dollar loss has a direct relationship to the crime classification. As we cover investigation procedures, we will see why it is so important to account for all time and money spent on the investigation.

Criminal law falls under two main jurisdictions: Federal and State. Although there is a plethora of federal and state statutes which may be used against traditional criminal offenses, and even though many of these same statutes may apply to computer related crimes with some measure of success, it is clear that many cases fail to reach prosecution or fail to result in conviction because of the gaps which exists in the Federal Criminal Code and the individual state criminal statutes.

Because of this, every state in the United States, with the exception of one, along with the Federal government, have adopted new laws specific to computer related abuses. These new laws, which have been redefined over the years to keep abreast of the constant changes in the technological forum, have been subjected to an ample amount of scrutiny due to many social issues, which have been impacted by the proliferation of computers in society. Some of these issues, such as privacy, copyright infringement, and software ownership are yet to be resolved, thus we can expect many more changes to the current collection of laws. Some of the computer related crimes, which are addressed by the new state and federal laws, are

- Unauthorized access
- Exceed authorized access
- Intellectual property theft or misuse of information
- Child Pornography
- Theft of services
- Forgery
- Property theft (i.e. Computer hardware, chips, etc.)
- Invasion of privacy
- Denial of services
- Computer fraud
- Viruses
- Sabotage (Data alteration or malicious destruction)
- Extortion
- Embezzlement
- Espionage
- Terrorism

All but one state, Vermont, have created or amended laws specifically to deal with computer-related crime. Twenty-five of the states have enacted specific computer crime statutes, while the other twenty-four states have merely amended their traditional criminal statutes to confront computer crime issues. Vermont has announced legislation under Bill H.0555, which deals with theft of computer services. The elements of proof, which define the basis of the criminal activity, vary from state to state. Security practitioners should be fully cognizant of their own state laws, specifically the elements of proof. Additionally, traditional criminal statutes, such as theft, fraud, extortion and embezzlement, can still be used to prosecute computer crime.

Just as there has been numerous new legislation at the State level, there have also been many new federal policies, such as the:

- Electronic Communications Privacy Act
- Electronic Espionage Act of 1996
- Child Pornography Prevention Act of 1996
- Computer Fraud and Abuse Act of 1986, 18 U.S.C. 1001

These laws and policies have been established, precisely to deal with computer and telecommunications abuses at the Federal level. Additionally, many modifications and updates have been made to the Federal Criminal Code, Sections 1029 and 1030, to deal with a variety of computer related abuses. Even though these new laws have been adopted for use in the prosecution of a computer-related offense, some of the older, proven federal laws, identified below, offer a "simpler" case to present to judges and juries:

- Wire Fraud
- Mail Fraud
- Interstate Transportation of Stolen Property
- Racketeer Influenced & Corrupt Organizations (RICO)

The Electronic Communications Privacy Act (ECPA) is being tested more today than ever before. The ECPA prohibits all monitoring of wire, oral, and electronic communications unless specific statutory exceptions apply. This includes monitoring of e-mail, network traffic, keystrokes, or telephone systems. The ECPA was not meant to prohibit network providers from monitoring and maintaining their networks and connections, thus the ECPA provides an exception for monitoring network traffic for legitimate businesses purposes. Additionally, the ECPA also allows monitoring when the network users are notified of the monitoring process.

The two new Acts enacted in 1996, the Child Pornography Prevention Act (CPPA) and the Electronic Espionage Act (EEA) have proved that the legislative process is working, albeit a bit slower than one would like. The

CPPA is especially impressive in that it eradicates many of the loopholes afforded by newer technology. The CPPA was enacted specifically to combat the use of computer technology to produce pornography that conveys the impression that children were used in the photographs or images, even if the participants are actually adults. The Court held that any child pornography, including simulated or morphed images, stimulate the sexual appetites of pedophiles and that the images themselves may persuade a child to engage in sexual activity by viewing other children. The CPPA was contested by the Freedom of Speech Coalition (FSC), but was upheld by the Court in FSC v. Reno.

The EEA hopefully will curtail some of the industrial espionage that is going on today, but it will also have an impact on how business is conducted in the United States, especially intelligence gathering. According to the EEA, it is a criminal offense to take, download, receive, or possess trade secret information obtained without the owner's authorization. Penalties can reach $10 Million in fines, up to 15 years in prison, and forfeiture of property used in the commission of the crime. This could have tremendous, far-reaching consequences for businesses should an employee improperly use information gained from any previous employment.

Civil Law

Civil law (or tort law) identifies a tort as a wrong against an individual or business, which normally results in damage or loss to that individual or business. The major differences between criminal and civil law, are the type of punishment and the level of proof required to obtain a guilty verdict. There is no jail sentence under the civil law system. A victim may receive financial or injunctive relief as restitution for their loss. An injunction against the offender will attempt to thwart any further loss to the victim. Additionally, a violation of the injunction may result in a Contempt of Court order, which would place the offender in jeopardy of going to jail. The main purpose for seeking civil remedy is for financial restitution, which can be awarded as follows:

- Compensatory Damages
- Punitive Damages
- Statutory Damages

In a civil action, if there is no culpability on the part of the victim, the victim may be entitled to compensatory (restitution), statutory, and punitive damages. Compensatory damages are actual damages to the victim and include attorney fees, lost profits, investigation costs, etc. Punitive damages are just that — damages set by the jury, with the intent to punish the offender. Even if the victim is partially culpable, an award may be made on the victim's behalf, but may be lessened due to the victim's culpable

negligence. Statutory damages are damages determined by law. Mere violation of the law entitles the victim to a statutory award.

Civil cases are much easier to convict under because the burden of proof required for a conviction is much less. To be found guilty of a civil wrong, the jury must believe, based only upon the preponderance of the evidence, that the offender is guilty of the offense. It is much easier to show that the majority (51%) of the evidence is pointing to the defendant's guilt.

Finally, just as a Search Warrant is used by law enforcement as a tool in the criminal investigation, the Court can issue an Inpoundment Order or Writ of Possession, which is a court order to take back the property in question. The investigator should also keep in mind that the criminal and civil case could take place simultaneously, thus allowing items seized during the execution of the Search Warrant to be used in the civil case.

Insurance

An insurance policy is generally part of an organization's overall risk mitigation/management plan. The policy offsets the risk of loss to the insurance company in return for an acceptable level of loss (the insurance premium). Since many computer-related assets (software and hardware) account for the majority of an organization's net worth, they must be protected by insurance. If there is a loss to any of these assets, the insurance company is usually required to pay out on the policy. One important factor to bear in mind, is the principle of culpable negligence. This places part of the liability on the victim if the victim fails to follow a "standard of due care" in the protection of identified assets. If a victim organization is held to be culpably negligent, the insurance company may be required to pay only a portion of the loss. Also, an insurance company can attempt to deny coverage, arguing that an employee's "dishonest" acts caused the damage.

Two important insurance issues related to the investigation are prompt notification of the loss and understanding that the insurance company has a duty to defend. Regarding prompt notification, insurance companies may deny coverage by arguing that the claim was received too late. Some states even allow insurance companies to void its insurance obligations if the notice or claim is proven to be late.

RULES OF EVIDENCE

Before delving into the investigative process and computer forensics, it is essential that the investigator have a thorough understanding of the Rules of Evidence. The submission of evidence in any type of legal proceeding generally amounts to a significant challenge, but when computers are involved, the problems are intensified. Special knowledge is needed to locate and collect evidence and special care is required to preserve and

transport the evidence. Evidence in a computer crime case may differ from traditional forms of evidence inasmuch as most computer-related evidence is intangible—in the form of an electronic pulse or magnetic charge.

Before evidence can be presented in a case, it must be competent, relevant and material to the issue and it must be presented in compliance with the rules of evidence. Anything which tends to prove directly or indirectly, that a person may be responsible for the commission of a criminal offense may be legally presented against him. Proof may include the oral testimony of witnesses or the introduction of physical or documentary evidence.

By definition, **evidence** is any species of proof or probative matter, legally presented at the trial of an issue, by the act of the parties and through the medium of witnesses, records, documents, objects, etc., for the purpose of inducing belief in the minds of the court and jurors as to their contention. In short, evidence is anything offered in court to prove the truth or falsity of a fact at issue. This section will cover each of the Rules of Evidence as they relate to computer crime investigations.

Types of Evidence

There are many types of evidence that can be offered in court to prove the truth or falsity of a given fact. The most common forms of evidence are direct, real, documentary and demonstrative. Direct evidence is oral testimony, whereby the knowledge is obtained from any of the witness's five senses and is, in itself, proof or disproof of a fact in issue. Direct evidence is called to prove a specific act (i.e. Eye Witness Statement). Real Evidence, also known as associative or physical evidence, is made up of tangible objects that prove or disprove guilt. Physical evidence includes such things as tools used in the crime, fruits of the crime, perishable evidence capable of reproduction, etc. The purpose of the physical evidence is to link the suspect to the scene of the crime. It is this evidence which has material existence and can be presented to the view of the court and jury for consideration. Documentary evidence is evidence presented to the court in the form of business records, manuals, printouts, etc. Much of the evidence submitted in a computer crime case is documentary evidence. Finally, demonstrative evidence is evidence used to aid the jury. It may be in the form of a model, experiment, chart, or an illustration offered as proof.

It should be noted that in order to aid the court and the jury in their quest to understand the facts at issue, demonstrative evidence is being used more often, especially in the form of simulation and animation. It is very important to understand the difference between these two types of evidence because the standard of admissibility is affected. A computer simulation is a prediction or calculation about what will happen in the future given known facts. A traffic reconstruction program is a perfect example of

computer simulation. There are many mathematical algorithms used in this type of program, that must be either stipulated to, or proven to the court to be completely accurate. It is generally more difficult to admit a simulation as evidence, because of the substantive nature of the process.

Computer animation, on the other hand, is simply a computer-generated sequence, illustrating an expert's opinion. Animation does not predict future events. It merely supports the testimony of an expert witness through the use of demonstrations. An animation of a hard disk spinning, while the read/write heads are reading data, can help the court or jury understand how a disk drive works. There are no mathematical algorithms that must be proven. The animation solely aids the court and jury through visualization. The key to having animation admitted as evidence is in the strength of the expert witness. Under Rule 702, the expert used to explain evidence must be qualified to do so through skill, training or education.

When seizing evidence from a computer-related crime, the investigator should collect any and all physical evidence, such as the computer, peripherals, notepads, documentation, etc. in addition to computer-generated evidence. There are four types of computer-generated evidence. They are

- Visual output on the monitor
- Printed evidence on a printer
- Printed evidence on a plotter
- Film recorder — Includes magnetic representation on disk, tape or cartridge, and optical representation on CD

Best Evidence Rule

The Best Evidence Rule, which had been established to deter any alteration of evidence, either intentionally or unintentionally, states that the court prefers the original evidence at the trial, rather than a copy, but they will accept a duplicate under the following conditions:

- Original lost or destroyed by fire, flood or other acts of God. This has included such things as careless employees or cleaning staff.
- Original destroyed in the normal course of business
- Original in possession of a third party who is beyond the court's subpoena power

This rule has been relaxed to now allow duplicates unless there is a genuine question as to the original's authenticity, or admission of the duplicate would under the circumstances be unfair.

Exclusionary Rule

Evidence must be gathered by law enforcement in accordance with court guidelines governing search and seizure or it will be excluded (Fourth

Amendment). Any evidence collected in violation of the Fourth Amendment is considered to be "Fruit of the Poisonous Tree," and will not be admissible. Furthermore, any evidence identified and gathered as a result of the initial inadmissible evidence will also be held to be inadmissible. Evidence may also be excluded for other reasons, such as violations of the Electronic Communications Privacy Act (ECPA) or violations related to provisions of Chapters 2500 and 2700 of Title 18 of the United States Penal Code.

Private citizens are not subject to the Fourth Amendment's guidelines on search and seizure, but are exposed to potential exclusions for violations of the ECPA or Privacy Act. Therefore, internal investigators, private investigators, and Computer Emergency Response Team (CERT) team members should take caution when conducting any internal search, even on company computers. For example, if there were no policy in place explicitly stating the company's right to electronically monitor network traffic on company systems, then internal investigators would be well advised not to set up a sniffer on the network to monitor such traffic. To do so may be a violation of the ECPA.

Hearsay Rule

A legal factor of computer-generated evidence is that it is considered hearsay. Hearsay is second-hand evidence; evidence which is not gathered from the personal knowledge of the witness but from another source. Its value depends on the veracity and competence of the source. The magnetic charge of the disk or the electronic bit value in memory, which represents the data, is the actual, original evidence. The computer-generated evidence is merely a representation of the original evidence.

Under the US Federal Rules of Evidence, all business records, including computer records, are considered "hearsay" because there is no firsthand proof that they are accurate, reliable, and trustworthy. In general, hearsay evidence is not admissible in court. However, there are some well-established exceptions (Rule 803) to the hearsay rule for business records. In Rosenberg v. Collins, the court held that if the computer output is used in the regular course of business, then the evidence shall be admitted.

Business Record Exemption to the Hearsay Rule

US Federal Rules of Evidence 803(6) allows a court to admit a report or other business document made at or near the time by, or from information transmitted by, a person with knowledge, if kept in the course of regularly conducted business activity, and if it was the regular practice of that business activity to make the [report or document], all as shown by testimony of the custodian or other qualified witness, unless the source of information or the method or circumstances of preparation indicate lack of trustworthiness.

To meet Rule 803 (6) the witness must:

- Have custody of the records in question on a regular basis
- Rely on those records in the regular course of business
- Know that they were prepared in the regular course of business

Audit trails would meet the criteria if they were produced in the normal course of business. The process to produce the output will have to be proven to be reliable. If computer-generated evidence is used and admissible, the court may order disclosure of the details of the computer, logs, maintenance records, etc. in respect to the system generating the printout, and then the defense may use that material to attack the reliability of the evidence. If the audit trails are not used or reviewed (at least the exceptions — i.e., failed log-on attempts) in the regular course of business, then they may not meet the criteria for admissibility.

US Federal Rules of Evidence 1001 (3) provides another exception to the Hearsay Rule. This rule allows a memory or disk dump to be admitted as evidence, even though it is not done in the regular course of business. This dump merely acts as statement of fact. System dumps (in binary or hexidecimal) would not be hearsay because it is not being offered to prove the truth of the contents, but only the state of the computer.

Chain of Evidence (Custody)

Once evidence is seized, the next step is to provide for its accountability and protection. The Chain of Evidence, which provides a means of accountability, must be adhered to by law enforcement when conducting any type of criminal investigation, including a computer crime investigation. It helps to minimize the instances of tampering. The Chain of Evidence must account for all persons who handled or who had access to the evidence in question.

The Chain of Evidence shows:

- Who obtained the evidence
- Where and when the evidence was obtained
- Who secured the evidence
- Who had control or possession of the evidence

It may be necessary to have anyone associated with the evidence testify at trial. Private citizens are not required to maintain the same level of control of the evidence as law enforcement although they would be well advised to do so. Should an internal investigation result in the discovery and collection of computer-related evidence, the investigation team should follow the same, detailed chain of evidence as required by law enforcement. This will help to dispel any objection by the defense, that the evidence is unreliable, should the case go to court.

Admissibility of Evidence

The admissibility of computer-generated evidence is, at best, a moving target. Computer-generated evidence is always suspect because of the ease with which it can be tampered—usually without a trace! Precautionary measures must be taken in order to ensure that computer-generated evidence has not been tampered with, erased, or added to. In order to ensure that only relevant and reliable evidence is entered into the proceedings, the judicial system has adopted the concept of admissibility.

- *Relevancy of Evidence* — evidence tending to prove or disprove a material fact. All evidence in court must be relevant and material to the case.
- *Reliability of Evidence* — The evidence and the process to produce the evidence must be proven to be reliable. This is one of the most critical aspects of computer-generated evidence.

Once computer-generated evidence meets the Business Record Exemption to the hearsay rule, is not excluded for some technicality or violation, follows the Chain of Custody, and is found to be both relevant and reliable, then it is held to be admissible. The defense will attack both the relevancy and reliability of the evidence, so great care should be taken to protect both.

Evidence Life Cycle

The Evidence Life Cycle starts with the discovery and collection of the evidence. It progresses through the following series of states until it is finally returned to the victim or owner:

- Collection and Identification
- Analysis
- Storage, Preservation and Transportation
- Presented in Court
- Returned to Victim (Owner)

Collection and Identification. As the evidence is obtained or collected, it must be properly marked so that it can be identified as being the particular piece of evidence gathered at the scene. The collection must be recorded in a logbook identifying the particular piece of evidence, the person who discovered it, and the date, time and location discovered. The location should be specific enough for later recollection in court. All other types of identifying marks, such as make, model or serial number, should also be logged. It is of paramount importance to list any type of damage to the particular piece of evidence. This is not only for identification purposes, but it will also limit any potential liability should a claim be made later on that you damaged the evidence. When marking evidence, the following guidelines should be followed:

- Mark the actual piece of evidence if it will not damage the evidence, by writing or scribing your initials, the date and the case number if known. Seal this evidence in the appropriate container and again, mark the container by writing or scribing your initials, the date and the case number, if known.
- If the actual piece of evidence cannot be marked, then seal the evidence in an appropriate container, then mark the container by writing or scribing your initials, the date and the case number, if known.
- The container should be sealed with evidence tape and your marking should write over the tape, so that if the seal is broken it can be noticed.
- Be extremely careful not to damage the evidence while engraving or marking the piece.

When marking glass or metal, a diamond scriber should be used. For all other objects, a felt tip pen with indelible ink is recommended. Dependent on the nature of the crime, the investigator may wish to preserve latent fingerprints. If so, static free gloves should be used if working with computer components, instead of standard latex gloves.

Try to always mark evidence the same way, because you will be asked to testify to the fact that you are the person identified by the evidence markings. Keep in mind, that the defense is going to try to discredit you as a witness or try some way to keep the evidence out of court, so something as simple as quick, positive identification of your mark is largely beneficial to the your case.

Storage, Preservation, and Transportation. All evidence must packed and preserved to prevent contamination. It should be protected against heat, extreme cold, humidity, water, magnetic fields, and vibration. The evidence must be protected for future use in court and for return to the original owner. It the evidence is not properly protected, the person or agency responsible for the collection and storage of the evidence may be held liable for damages. Therefore, the proper packing materials should be used whenever possible. Documents and disks (hard, floppy, optical, tapes, etc.) should be seized and stored in appropriate containers to prevent their destruction. For example, hard disks should be packed in a sealed, static-free bag, within a cardboard box with a foam container. The box should be sealed with evidence tape and an Electromagnetic Field (EMF) warning label should be affixed to the box. It may be wise to defer to the system administrator or a technical advisor on how to best protect a particular type of system, especially mini-systems or mainframes.

Finally, evidence should be transported to a location where it can be stored and locked. Sometimes the systems are too large to transport, thus the forensic examination of the system may need to take place on site.

Presented in Court. Each piece of evidence that is used to prove or disprove a material fact needs to be presented in court. After the initial seizure, the evidence is stored until needed for trial. Each time the evidence is transported to and from the courthouse for the trial, it needs to be handled with the same care as with the original seizure. Additionally, the Chain of Custody must continue to be followed. This process will continue until all testimony related to the evidence is completed. Once the trail is over, the evidence can be returned to the victim (owner).

Returned to Victim (Owner). The final destination of most types of evidence is back with its original owner. Some types of evidence, such as drugs or paraphernalia (i.e. contraband) are destroyed after the trial. Any evidence gathered during a search, even though maintained by law enforcement, is legally under the control of the courts. Even though a seized item may be yours and may even have your name on it, it may not be returned to you unless the suspect signs a release or after a hearing by the court. Unfortunately, many victims don't want to go to trial. They just want to get their property back.

Many investigations merely need the information on a disk to prove or disprove a fact in question, thus there is no need to seize the entire system. Once a schematic of the system is drawn or photographed, the hard disk can be removed and then transported to a forensic lab for copying. Mirror copies of the suspect disk are obtained using forensic software and then one of those copies can be returned to the victim so that business operations can resume.

COMPUTER CRIME INVESTIGATION

The computer crime investigation should start immediately following the report of any alleged criminal activity. Many processes ranging from reporting and containment to analysis and eradication need to be accomplished as soon as possible after the attack. An Incident Response Plan should be formulated and a Computer Emergency Response Team (CERT) should be organized prior to the attack. The Incident Response Plan will help set the objective of the investigation and will identify each of the steps in the investigative process.

The use of a Corporate CERT Team is invaluable. Due to the numerous complexities of any computer-related crime, it is extremely advantageous to have a single group that is acutely familiar with the Incident Response Plan to call upon. The CERT team should be a technically astute group, that is knowledgeable in the area of legal investigations, the Corporate Security Policy (especially the Incident Response Plan), the severity levels of various attacks, and the company position on information dissemination and disclosure.

The Incident Response Plan should be part of the overall Corporate Computer Security Policy. The plan should identify reporting requirements, severity levels, guidelines to protect the crime scene and preserve evidence, etc. The priorities of the investigation will vary from organization to organization but the issues of containment and eradication are reasonably standard, that is to minimize any additional loss and resume business as quickly as possible. The following sections describe the investigative process starting with the initial detection.

Detection and Containment

Although intrusion detection is covered elsewhere in this manual, it must be mentioned that before any investigation can take place, the system intrusion or abusive conduct must first be detected. The closer the detection is to the actual intrusion event will not only help to minimize system damage, but will also assist in the identification of potential suspects.

To date, most computer crimes have either been detected by accident or through the laborious review of lengthy audit trails. While audit trails can assist in providing user accountability, their detection value is somewhat diminished because of the amount of information that must be reviewed and because these reviews are always post-incident. Accidental detection is usually made through observation of increased resource utilization or inspection of suspicious activity, but again, is not effective due to the sporadic nature of this type of detection.

These types of reactive or passive detection schemes are no longer acceptable. Proactive and automated detection techniques need to be instituted in order to minimize the amount of system damage in the wake of an attack. Real-time intrusion monitoring can help in the identification and apprehension of potential suspects and automated filtering techniques can be used to make audit data more useful.

Once an incident is detected it is essential to minimize the risk of any further loss. This may mean shutting down the system and reloading clean copies of the operating system and application programs. It should be noted, that failure to contain a known situation (i.e. system penetration) might result in increased liability for the victim organization. For example, if a company's system has been compromised by an external attacker and the company failed to shut down the intruder, hoping to trace him, the company may be held liable for any additional harm caused by the attacker.

Report to Management

All incidents should be reported to management as soon as possible. Prompt internal reporting is imperative in order to collect and preserve

potential evidence. It is important that information about the investigation be limited to as few people as possible. This should be done on a need-to-know basis. This limits the possibility of the investigation being leaked. Additionally, all communications related to the incident should be made via an out-of-band method to ensure the intruder does not intercept any incident-related information. In other words, do not use e-mail to discuss the investigation on a compromised system. Based on the type of crime and type of organization it may be necessary to notify:

- Executive Management
- Information Security Department
- Physical Security Department
- Internal Audit Department
- Legal Department

Preliminary Investigation

A preliminary internal investigation is necessary for all intrusions or attempted intrusions. At a minimum, the investigator must ascertain if a crime has occurred; and if so, he must identify the nature and extent of the abuse. It is important for the investigator to remember that the alleged attack or intrusion may not be a crime at all. Even if it appears to be some form of criminal conduct, it could merely be an honest mistake. Most internal losses occur from errors, not from overt criminal acts. There is no quicker way to initiate a lawsuit than to mistakenly accuse an innocent person of criminal activity.

The preliminary investigation usually involves a review of the initial complaint, inspection of the alleged damage or abuse, witness interviews, and, finally, examination of the system logs. If during the preliminary investigation, it is determined that some alleged criminal activity has occurred, the investigator must address the basic elements of the crime to ascertain the chances of successfully prosecuting a suspect either civilly or criminally. Additionally, the investigator must identify the requirements of the investigation (dollars and resources). If it is believed that a crime has been committed, neither the investigator nor any other company personnel should confront or talk with the suspect. Doing so would only give the suspect the opportunity to hide or destroy evidence.

Determine if Disclosure is Required

It must be determined if a disclosure is required or warranted, due to laws or regulations. Disclosure may be required by law or regulation or may be required if the loss affects a corporation's financial statement. Even if disclosure is not required, it is sometimes better to disclose the attack to possibly deter future attacks. This is especially true if the victim organization

prosecutes criminally and/or civilly. Some of the following attacks would probably result in disclosure:

- Large Financial Loss of a Public Company
- Bank Fraud
- Public Safety Systems (i.e. Air Traffic Control)

The Federal Sentencing Guidelines also require organizations to report criminal conduct. The stated goals of the Commission were to "provide just punishment, adequate deterrence, and incentives for organizations to maintain internal mechanisms for preventing, detecting, and reporting criminal conduct." The Guidelines also state that organizations have a responsibility to "maintain internal mechanism for preventing, detecting, and reporting criminal conduct." The Federal Sentencing Guidelines do not prevent an organization from conducting preliminary investigations to ascertain if, in fact, a crime has been committed. One final note of the Federal Sentencing Guidelines, is that they were designed to punish computer criminals for acts of recidivism and using their technical skills and talents to engage in criminal activity.

If the decision is made to disclose an alleged incident or intrusion, be sure to be especially careful when dealing with the media. The media has a history of sensationalizing these types of events and can easily distort the facts that could portray the victim organization as the "Goliath," using the "David v. Goliath" analogy. Make sure that you have all the facts and provide the media with the "slant" that best serves your purposes. Do not lie to the media! A "No Comment" is better then lying.

Investigation Considerations

Once the preliminary investigation is complete and the victim organization has made a decision related to disclosure, the organization must decide on the next course of action. The victim organization may decide to do nothing or they may attempt to eliminate the problem and just move on. Deciding to do nothing is not a very good course of action as the organization may be held to be culpably negligent should another attack or intrusion occur. The victim organization should at least attempt to eliminate the security hole that allowed the breach, even if they do not plan to bring the case to court. If the attack is internal, the organization may wish to conduct an investigation that might only result in the dismissal of the subject. If they decide to further investigate the incident, they must also determine if they are going to prosecute criminally or civilly, or are they merely conducting the investigation for insurance purposes. If an insurance claim is to be submitted, a police report is usually necessary.

When making the decision to prosecute a case, the victim must clearly understand the overall objective. If the victim is looking to make a point by

punishing the attacker, then a criminal action is warranted. This is one of the ways to deter potential future attacks. If the victim were seeking financial restitution or injunctive relief, then a civil action would be appropriate. Keep in mind that a civil trial and criminal trial can happen in parallel. Information obtained during the criminal trial can be used as part of the civil trial. The key is to know what you want to do at the outset, so all activity can be coordinated.

The evidence or lack thereof, may also hinder the decision to prosecute. Evidence is a significant problem in any legal proceeding, but the problems are compounded when computers are involved. Special knowledge is needed to locate and collect the evidence while special care is required to preserve the evidence.

There are many factors to consider when deciding upon whether or not to further investigate an alleged computer crime. For many organizations, the primary consideration will be the cost associated with an investigation. The next consideration will probably be the impact to operations or the impact to business reputation. The organization must answer the following questions:

- Will productivity be stifled by inquiry process?
- Will the subject system have to be shut down to conduct an examination of the evidence or crime scene?
- Will any of the system components be held as evidence?
- Will proprietary data be subject to disclosure?
- Will there be any increased exposure for failing to meet a "standard of due care"?
- Will there be any adverse publicity related to the loss?
- Will a disclosure invite other perpetrators to commit similar acts or will an investigation and subsequent prosecution deter future attacks?

The answers to these questions may have an impact on how the investigation is handled and who is called in to conduct the investigation. Furthermore, these issues must be addressed early on, so that the proper authorities can be notified if required. Prosecuting an alleged criminal offense is a very time consuming task. Law enforcement and the prosecutor will expect a commitment of time and resources for the following:

- Interviews to prepare crime reports and search warrant affidavits
- Engineers or computer programmers to accompany law enforcement on search warrants
- Assistance of the victim company to identify and describe documents, source code, and other found evidence
- A company expert who may needed for explanations and assistance during the trial

- Discovery — Documents may need to be provided to the defendant's attorney for discovery. They may ask for more than you want to provide. Your attorney will have to argue against broad ranging discovery. Defendants are entitled to seek evidence they need for their defense.
- You and other company employees will be subpoenaed to testify.

Who Should Conduct the Investigation?

Based upon the type of investigation (i.e. civil, criminal, insurance, or administrative) and extent of the abuse, the victim must decide who is to conduct the investigation. This used to be a fairly straightforward decision, but high-technology crime has altered the decision making process. Inadequate and untested laws combined with the lack of technical training and technical understanding, has severely hampered the effectiveness of our criminal justice system when dealing with computer-related crimes.

In the past, society would adapt to change, usually at the same rate of that change. Today, this is no longer true. The information age has ushered in dramatic technological changes and achievements, which continue to evolve at exponential rates. The creation, the computer itself, is being used to create new technologies or advance existing ones. This cycle means that changes in technology will continue to occur at an ever-increasing pace. What does this mean to the system of law? It means we have to take a look at how we establish new laws. We must adapt the process to account for the excessive rate of change. Unfortunately, this is going to take time! In the mean time, if they are to launch an investigation, the victim must choose from the following options:

- Conduct an internal investigation
- Bring in external private consultants/investigations
- Bring in local/state/federal law enforcement

Exhibit 1 identifies each of the tradeoffs.

Law enforcement officers have greater search and investigative capabilities than private individuals, but they also have more restrictions than private citizens. For law enforcement to conduct a search, a warrant must first be issued. No warrant is needed if the victim or owner of compromised system gives permission to conduct the search. Issuance of the search warrant is based upon probable cause (reason to believe the something is true). Once probable cause has been identified, law enforcement officers have the ability to execute search warrants, subpoenas and wire taps. The warrant process was formed in order to protect the rights of the people. The Fourth Amendment to the Constitution of the United States established the following:

Group	Cost	Legal Issues	Information Dissemination	Investigative Control
Internal Investigators	Time/People Resources	Privacy Issues Limited Knowledge of Law and Forensics	Controlled	Complete
Private Consultants	Direct Expenditure	Privacy Issues	Controlled	Complete
Law Enforcement Officers	Time/People Resources	Fourth Amendment Issues Jurisdiction Miranda Privacy Issues	Uncontrolled Public Information (FOIA)	None

Exhibit 1. Tradeoffs for Three Options Compensating for Rate of Change

---

The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no Warrants shall issue, but upon probable cause, supported by oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.

There are certain exceptions to this. The "exigent circumstances" doctrine allows for a warrantless seizure, by law enforcement, when the destruction of evidence is impending. In United States v. David, the court held that "When destruction of evidence is imminent, a warrantless seizure of that evidence is justified if there is probable cause to believe that the item seized constitutes evidence of criminal activity."

Internal investigators (non-government) or private investigators, acting as private citizens, have much more latitude in conducting a warrantless search, due to a ruling by the Supreme Court, in Burdeau v. McDowell. In this case, the Supreme Court held that evidence obtained in a warrantless search, could be presented to a grand jury by a government prosecutor, because there was no unconstitutional government search and hence no violation of the fourth amendment.

Normally, a private (party) citizen is not subject to the rules and laws governing search and seizure, but a private citizen becomes a police agent, and the Fourth Amendment applies, when:

- the private party performs a search which the government would need a search warrant to conduct;
- the private party performs that search to assist the government, as opposed to furthering its own interest; and
- the government is aware of that party's conduct and does not object to it.

The purpose of this doctrine is to eliminate the opportunity for government to circumvent the warrant process by eliciting the help of a private citizen. If a situation required law enforcement to obtain a warrant, due to the subject's expectations of privacy, and the government knowingly allowed a private party to conduct a search in order to disclose evidence, the court would probably rule that the private citizen acted as a police agent. A victim acting to protect its property by assisting police to prevent or detect a crime does not become a police agent.

Law enforcement personnel are not alone in their ability to obtain a warrant. A private party can also obtain a warrant, albeit a civil one, to search and seize specifically identified property which they make claim to. This civil warrant, also known as a Writ of Possession, allows the plaintiff to seize property that is rightfully theirs. In order to obtain such a court order, the plaintiff much prove to a judge or magistrate that the property in question is theirs and that an immediate seizure is essential to minimizing any collateral monetary loss. Additionally, the plaintiff must also post a bond, double the value of the property in question. This places an enormous burden on the plaintiff, should they be unsuccessful in their endeavor, but it also protects individuals and businesses against frivolous requests made to the court.

The biggest issues affecting the decision on who to bring in (in order of priority) are information dissemination, investigative control, cost, and the associated legal issues. Once an incident is reported to law enforcement, information dissemination becomes uncontrolled. The same holds true for investigative control. Law enforcement controls the entire investigation, from beginning to end. This is not always bad, but the victim organization may have a different set of priorities. Cost is always a concern and the investigation costs only add to the loss initially sustained by the attack or abuse. Even law enforcement agencies, which are normally considered "free," add to the costs because of the technical assistance they require during the investigation.

Another area that affects law enforcement is jurisdiction. Jurisdiction is the geographic area where the crime had been committed and any portion

of the surrounding area over, or through which the suspect passed, is enroute to, or going away from, the actual scene of the crime. Any portion of this area adjacent to the actual scene over which the suspect, or the victim, might have passed, and where evidence might be found, is considered part of the crime scene. When a system is attacked remotely, where did the crime occur? Most courts submit that the crime scene is the victim's location. But what about "enroute to"? Does this suggest that a crime scene may also encompass the telecommunications path used by the attacker? If so, and a theft occurred, is this interstate transport of stolen goods? There seem to be more questions than answers but only through cases being presented in court can precedence be set. It will take time for the answers to shake out.

There are advantages and disadvantages to each of the groups identified above. Internal investigators will know your systems the best, but may lack some of the legal and forensic training. Private investigators, who specialize in high-technology crime, also have a number of advantages, but usually result in higher costs. Private security practitioners and private investigators are also private businesses and may be more sensitive to business resumption than law enforcement. If you elect to retain the services of a private investigator or computer consultant, it is best if your corporate counsel retains them. This protects the victim organization from unwarranted or untimely disclosure. All communications are treated as privileged communications, under the Attorney-Client Privilege. Additionally, all work product is protected by the same privilege and is protected from disclosure. This includes details of the investigation, witness interviews, forensic analysis, etc. It also includes any past criminal activity, by the victim organization, which may be uncovered during the investigation.

Should you decide to contact your local police department, call the detective unit directly. Chances are you will get someone who is more experienced and knowledgeable and someone who can be more discrete. If you call 911, a uniformed officer will arrive on your doorstep and possibly alert the attacker. Furthermore, the officer must create a report of the incident that will become part of a public log. Now the chances for a discretionary dissemination of information and a covert investigation are gone.

Ask the detective to meet with you in plain clothes. When they arrive at your business have them announce themselves as consultants. If you decide that you would like Federal authorities to be present, do so, but you should inform the local law enforcement authorities. Be aware that your local law enforcement agency may not be well equipped to handle high-tech crime. The majority of law enforcement agencies have limited budgets and, as such, place an emphasis on problems related to violent crime and drugs. Also, with technology changing so rapidly, most law enforcement

officers lack the technical training to adequately investigate an alleged intrusion.

The same problems hold true for the prosecution and the judiciary. To successfully prosecute a case, both the prosecutor and the judge must have a reasonable understanding of high-tech laws and the crime in question. This is not always the case. Additionally, many of the current laws are woefully inadequate. Even though an action may be morally and ethically wrong, it is still possible that no law is violated (i.e. LaMacchia case). Even when there is a law that has been violated, many of these laws remain untested and lack precedence. Because of this many prosecutors are reluctant to prosecute high-tech crime cases.

Many recent judicial decisions have indicated that judges are lenient towards techno-criminal just as with other white-collar criminals. Furthermore, the lack of technology expertise may cause "doubt," thus rendering "not guilty" decisions. Since many of the laws concerning computer crime are new and untested, many judges have a concern with setting precedence, which may later be overturned in an appeal. Some of the defenses that have been used, and accepted by the judiciary, are

- If you have no system security or lax system security, then you are implying that there is no company concern. Thus there should be no court concern.
- If a person is not informed that access is unauthorized, then it can be used as a defense.
- If an employee is not briefed and does not acknowledge understanding of policy and procedures, then they can use it as a defense.

The Investigative Process

As with any type of criminal investigation the goal of the investigation is to know who, what, when, where, why, and how. It is important that the investigator logs all activity and account for all time spent on the investigation. The amount of time spent on the investigation has a direct impact on the total dollar loss for the incident. This may result in greater criminal charges and, possibly, stiffer sentencing. Finally, the money spent on investigative resources can be reimbursed as compensatory damages in a successful civil action.

Once the decision is made to further investigate the incident, the next course of action for the investigative team is to establish a detailed investigative plan, including the search and seizure plan. The plan should consists of an informal strategy that will be employed throughout the investigation, including the search and seizure:

- Identify any potential suspects
- Identify potential witnesses
- Identify what type of system is to be seized
- Identify the Search and Seizure Team Members
- Obtain a Search Warrant (if required)
- Determine if there is risk of the suspect destroying evidence or causing greater losses

Identify Any Potential Suspects. The type of crime and the type of attacker will set the stage for the overall investigation. Serious attacks against government sites, military installations, financial centers, or a telecommunications infrastructure must be met with the same fervor as that of a physical terrorist attack. Costs will not be the issue. On the other hand, when an organization plans to conduct an investigation pertaining to unauthorized access or a violation of company policy all the factors should considered. This includes the anticipated cost and the chances of success. In either case, there will always be the usual suspects: insiders and outsiders.

Insiders are usually trusted users who abuse their level of authorized access to the system. They are normally the greatest source of loss. They know the value of your assets! They are usually motivated by greed, need (i.e. drug habit, gambling problem, divorce, etc.), or perceived grievance. Most importantly that have the access and the opportunity. Outsiders, as the name implies, attack your systems and networks from the outside. They attack systems for a variety of reasons, with attacks increasing at alarming rates because of advancements such as the Internet. Some examples of Outsiders are as follows:

- Hackers and Crackers
- Organized Crime
- Terrorists
- Pedophiles
- Industrial/Corporate Spies

While, individually, each of these groups continue to be a problem, it is especially disturbing to realize the potential for collaboration between any two or more of the groups. When organized crime groups or terrorist factions gain access to the technical expertise provided by hackers and crackers, the potential for widespread harm and exorbitant financial losses is intensified. Albert Einstein said it best when he said, "Technological progress is like an axe in the hands of a pathological criminal."

When commencing with the investigation, it is important to understand how and why a system is being attacked. The how will provide you with information pertaining to technical expertise required to conduct the

attack. The why will potentially indicate motive. The how and why together, along with the when and the where, may provide the who.

Identify Potential Witnesses. It is important to identify potential witnesses early on in the investigation. It is just as important not to alert the suspect to the investigation, therefore selecting whom will be interviewed and when may have an impact on the investigation. The key to obtaining good witness statements is to ascertain the facts in the case, not opinions. Also, it is wise not to ask leading questions. Sources of information may be staff members, expert witnesses, associates, etc. Interviews are not the same as interrogations and great care should go into not confusing the two. If a hostile witness does not want to be interviewed, then the process should cease immediately. If a witness or potential witness is detained against their will, there may be criminal and/or civil liability to the individuals and business responsible for the investigation. Never intimidate, coerce, or harass a potential witness.

Technically competent personnel should conduct interviews of technical witnesses or suspects. A potential suspect, who is technically competent, will have a field day if interviewed by a non-technical investigator. Many times these individuals are arrogant to start with. If they feel that they have the upper hand, because of their "esoteric knowledge," they may be less inclined to provide a truthful statement. Also, it is sometimes better to interview a technical suspect (i.e. programmer) first, before seizing his system. If you advise the suspect that you will be seizing his systems if he does not cooperate, he may assist in the investigation.

One final note on conducting interviews. It is always a good idea to have the witness write out and sign their statement, in their own handwriting. This statement can then be types for better readability, but you can always point to the original. This helps to counter statements made by the witness in court, that that is not what they meant.

Identify the Type of System That Is to Be Seized. It is imperative to learn as much as possible about the target computer system(s). If possible, obtain the configuration of the system, including the network environment (if any), hardware, and software. The following data should be acquired prior to the seizure:

- Identify system experts. Make them part of the team.
- Is a security system in place on the system, If so, what kind? Are passwords used? Can a root password be obtained?
- Where is the system located? Will simultaneous raids be required?
- Obtain the required media supplies in advance of the operation
- What law has been violated? Discuss the elements of proof. These should be the focus of the search and seizure.

- What is your Probable Cause? Obtain a warrant if necessary.
- Determine if the analysis of the computer system will be conducted on site or back in the office or forensics lab.

Identify the Search and Seizure Team Members. There are different rules for Search and Seizure based upon who's conducting the search. Under the Fourth Amendment, law enforcement must obtain a warrant, which must be based on probable cause. Regardless of who's conducting the Search and Seizure, a team should be identified and should consist of the following members:

- Lead Investigator
- Information Security Department
- Legal Department
- Technical Assistance — System Administrator as long as he is not a suspect

If a Corporate CERT Team is already organized, then this process is already complete. A Chain of Command needs to be established and it must be determined who is to be in charge. This person is responsible for delegating assignments to each of the team members. A media liaison should be identified if the attack is to be disclosed. This will control the flow of information to the media.

Obtaining and Serving Search Warrants. If it is believed that the suspect has crucial evidence at his home or office, then a search warrant will be required to seize the evidence. If a search warrant is going to be needed, then it should be done as quickly as possible before the intruder can do further damage. The investigator must establish that a crime has been committed and that the suspect is somehow involved in the criminal activity. He must also show why a search of the suspect's home or office is required. The victim may be asked to accompany law enforcement when serving the warrant to identify property or programs.

If you must take along documents with you when serving the Search Warrant, consider coping them onto a colored paper to prevent the defense from inferring that what you might have found was left by you.

Is the System at Risk. Prior to the execution of the plan, the investigative team should ascertain if the suspect, if known, is currently working on the system. If so, the team must be prepared to move swiftly, so that evidence is not destroyed. The investigator should determine if the computer is protected by any physical or logical access control systems and be prepared to respond to such systems. It should also be decided early on, what will be done if the computer is on at the commencement of the seizure. The goal

of this planning is to minimize any risk of evidence contamination or destruction.

Executing the Plan

The first step in executing the plan is to secure and control the scene. This includes securing the power, network servers, and telecommunications links. If the suspect is near the system, it may be necessary to physically remove him. It may be best to execute the search and seizure after normal business hours to avoid any physical confrontation. Keep in mind, that even if a search is conducted after hours, the suspect may still have remote access to the system via a LAN-based modem connection, PC-based modem connection, wireless modem connection, or Internet connection. Many times it is required to seize a disk from the suspects computer, mirror image a copy of the disk and then replace the original with a copy of the disk, all without the suspect knowing what is happening. This allows the investigative team to protect the evidence and continue with the investigation, while retaining secrecy of the investigation.

Enter the area slowly so as not to disturb or destroy evidence. Evaluate the entire situation. In no other type of investigation, can evidence be destroyed more quickly. Do not touch the keyboard as this may invoke a Trojan Horse or some other rogue or malicious program. Do not turn off the computer unless it appears to be active (i.e. formatting the disk, deleting files, initiating some I/O process, etc.). Look for the disk activity light and listen for disk usage. If you must turn off the computer, pull the plug from the wall, rather than using the on/off switch. Look for notes, documentation, passwords, encryption codes, etc. The following questions must be answered in order to effectively control the scene:

- Is the subject system turned on?
- Is there a modem attached? If so,
  - Check for internal and wireless modems
  - Check for telephone lines connected to the computer
- Is the system connected to a LAN?

The investigator may wish to videotape the entire evidence collection process. There are two schools of thought on this. The first is that if you videotape the search and seizure, any mistakes can nullify the whole operation. The second school of thought is that if you videotape the evidence collection process, many of the claims by the defense can be silenced. In either case, be careful what you say if the audio is turned on!

Sketch and photograph the crime scene before touching anything. Sketches should be drawn to scale. Take still photographs of critical pieces of evidence. At a minimum, the following should be captured:

- The layout of desks and computers (Include dimensions and measurements)
- The configuration of the all computers on the network
- The configuration of the suspect computer, including network connections, peripheral connections, internal and external components, and system backplane
- The suspect computer display

A drawing package, such a Visio — Technical Edition, is excellent for these types of drawings. Visio allows the investigator to sketch the scene using a drag and drop graphical user interface (GUI). Most computer and network graphics, desk and furniture graphics, etc., are included with the application. The output is a professional product that is made part of the report and can be used later to recreate the environment or to present the case in court.

If the computer is on, the investigator should capture what is on the monitor. This can be accomplished by video taping what is on the screen. The best way to do this, without getting the "scrolling effect" caused by the video refresh, is to use a National Television Standards Committee (NTSC) adapter. Every monitor has a specific refresh rate (i.e. Horizontal: 30-66 KHz, Vertical: 50-90 Hz), which identifies how frequently the screen's image is redrawn. It is this redrawing process that causes the videotaped image to appear as if the vertical hold is not properly adjusted. The NTSC adapter is connected between the monitor and the monitor cable, and directs the incoming signal into the camcorder directly. The adapter converts the computer's analog signal (VGA) to a NTSC format. Still photos are a good idea too. Do not use a flash, because it can "white out" the image. Even if the computer is off, check the monitor for burnt-in images. This does not happen as much with the new monitors, but it may still help in the discovery of evidence.

Once you have reviewed and captured what's on the screen, pull the plug on the system. This is for PC-based systems only. Mini-systems or mainframes must be logically power-downed. It is best to conduct a forensic analysis (technical system review with a legal basis focused on evidence gathering) on a forensic system, in a controlled environment. If necessary, a forensic analysis can be conducted on site, but never using the suspect system's operating system or system utilities. See the section on forensic analysis for the process that should be followed.

Once the computer is turned off, remove the cover and photograph and sketch the inside of the computer. The analyst or investigator should use a static-dissipative grounding kit when working inside of the computer. You should note any peculiarities, such as booby traps. Identify each drive and its logical ID (i.e. C: drive) by tracing the ribbon cables to the I/O board.

Also identify any external drives. Once this has been completed, remove, label and pack all drives. Check the floppy drives for any media. If a disk is in the drive, remove the disk, and mark on the evidence label where it was found. Next, place a blank diskette into the floppy drive(s). Place evidence tape over the floppy drives and the on/off switch, once it is placed in the off position.

Identify, mark and pack all evidence according to the collection process under the Rules of Evidence. Identify and label all computer systems, cables, documents, disks, etc. The investigator should also seize all diskettes, backup tapes, PCMCIA disks, magnetic cartridges, optical disks, and printouts. All diskettes should be write protected. Make an entry for each in the evidence log. Check the printer. If it uses ribbons, make sure it (or at least the ribbon) is taken as evidence. Keep in mind that many of the peripheral devices may contain crucial evidence in their memory and/or buffers. Some items to consider are LAN servers, routers, printers, etc. You must check with the manufacturer on how to output the memory buffers for each device. Also, keep in mind that most buffers are stored in volatile memory. Once the power is cut, the information may be lost.

Additionally, check all drawers, closets and even the garbage for any forms of magnetic media (i.e. hard drives, floppy diskettes, tape cartridges, optical disks, etc.) or documentation. It seems that many computer literate individuals conduct most of their correspondence and work product on a computer. This is an excellent form of leads, but take care to avoid an invasion of privacy. Even media that appears to be destroyed can turn out to be quite useful. One case involved an American serviceman, who contracted to have his wife killed and wrote the letter on his computer. In an attempt to destroy all the evidence, he cut up the floppy disk, containing the letter, into 17 pieces. The Air Force Office of Special Investigations (AFOSI)was able to reconstruct the diskette and read almost all the information.

Don't overlook the obvious, especially hacker tools and any ill-gotten gains (i.e. password or credit card lists). This will help your case when trying to show motive and opportunity. The State of California has equated hacker tools to that of burglary tools; the mere possession constitutes a crime. Possession of a Red Box, or any other telecommunications instrument that has been modified with the intent to defraud, is also prohibited under U.S.C. Section 1029. Some of the hacker tools that you should be aware of are:

- Password crackers
- Network sniffers
- Automated probing tools (i.e. SATAN)
- Anonymous remailers
- War dialers
- Encryption and Steganography tools

Finally, phones, answering machines, desk calendars, day-timers, fax machines, pocket organizers, electronic watches, etc. are all sources of potential evidence. If the case warrants, seize and analyze all sources of data, both, electronic and manual. Document all activity in an Activity Log and if necessary secure the crime scene.

Surveillance

There are two forms of surveillance used in computer crime investigations. They are physical surveillance and computer surveillance. The physical surveillance can be generated at the time of the abuse, via CCTV security camera, or after the fact. When done after the fact, physical surveillance is usually performed undercover. It can be used in an investigation to determine a subject's personal habits, family life, spending habits, or associates.

Computer surveillance is achieved in a number of ways. It is done passively through audit logs or actively by way of electronic monitoring. Electronic monitoring can be accomplished via keyboard monitoring, network sniffing, or line monitoring. In any case, it generally requires a warning notice and/or explicit statement in the security policy, indicating that the company can and will electronically monitor any and all system or network traffic. Without such a policy or warning notice, a warrant is normally required.

Before you conduct electronic monitoring, make sure you review Chapters 2500 & 2700 of the Electronic Communications Privacy Act, Title 18 of the US Code as it relates to keystroke monitoring or system administrators looking into someone's account. If you do not have a banner or if the account holder has not been properly notified, the system administrator and the company can be guilty of a crime and liable for, both, civil and criminal penalties. Failure to obtain a warrant could result in the evidence being suppressed or worse yet, litigation by the suspect for invasion of privacy or violation of the ECPA.

One other method of computer surveillance that is used are "sting operations." These operations are established so as to continue to track the attacker, on-line. By baiting a trap or setting up "Honey Pots," the victim organization lures the attacker to a secured area of the system. This is what was done in the Cuckoo's Egg. The system attackers were enticed into accessing selected files. Once these files or their contents are downloaded to another system, their mere presence can be used as evidence against the suspect. This enticement is not the same as entrapment as the intruder is already predisposed to commit the crime. Entrapment only occurs when a law enforcement officer induces a person to commit a crime that the person had not previously contemplated.

It is very difficult to track and identify a hacker or remote intruder, unless there is a way to trace the call (i.e. Caller ID, wire tap, etc.). Even with these resources, many hackers meander through communication networks, hopping from one site to the next, via a multitude of telecommunications gateways and hubs, such as the Internet! Bill Cheswick, author of Firewalls and Internet Security, refers to this a "connection laundering." Additionally, the organization can not take the chance of allowing the hacker to have continued access to their system and potentially cause any additional harm.

Telephone traps require the equivalent of a search warrant. Additionally, the victim will be required to file a criminal report with law enforcement and must show probable cause. If sufficient probable cause is shown, a warrant will be issued and all incoming calls can be traced. Once a trace is made, a pen register is normally placed on the suspects phone to log all calls placed by the suspect. These entries can be tied to the system intrusions based upon the time of the call and the time the system was accessed.

Investigative and Forensic Tools

Exhibit 2, although not exhaustive, identifies some of the investigative and forensic tools that are commercially available. The first table identifies the hardware and software tools that should be part of the investigator's toolkit, while the second table identifies forensic software and utilities.

Other Investigative Information Sources

When conducting an internal investigation it is important to remember that the witness statements and computer-related evidence are not the only sources of information useful to the investigation. Personnel files provide a wealth of information related to an employee's employment history. It may show past infractions by the employee or disciplinary action by the company. Telephone and fax logs can possibly identify any accomplices or associates of subject. At a minimum they will identify the suspects most recent contacts. Finally, security logs, time cards, and check-in sheets will determine when a suspected insider had physical access to a particular system.

Investigative Reporting

The goal of the investigation is to identify all available facts related to the case. The investigative report should provide a detailed account of the incident, highlighting any discrepancies in witness statements. The report should be a well organized document that contains a description of the incident, all witness statements, references to all evidentiary articles, pictures of the crime scene, drawings and schematics of the computer and the computer network (if applicable), and finally, a written description of the forensic analysis. The report should state final conclusions, based solely

Investigative Tools	
Investigation and Forensic Toolkit Carrying Case	Static Charge Meter
Cellular Phone	EMF/ELF Meter (Magnetometer)
Laptop Computer	Gender Changer (9 Pin and 25 Pin)
Camcorder w/NTSC adapter	Line Monitor
35mm Camera (2)	RS232 Smart Cable
Wide Angle & Telephoto Lens	Nitrile Anti-static Gloves
Night Vision Adapter for Camera and Camcorder	Alcohol Cleaning Kit
Polaroid Camera	CMOS Battery
Tape Recorder (VOX)	Extension Cords
Scientific Calculator	Power Strip
Label Maker	Keyboard Key Puller
Crime Scene/Security Barrier Tape	Cable Tester
PC Keys	Breakout Box
IC Removal Kit	Transparent Static Shielding Bags (100 Bags)
Compass	Anti-Static Sealing Tape
Diamond Tip Engraving Pen Extra Diamond Tips	Serial Port Adapters (9 Pin 25 Pin & 25 Pin 9 Pin)
Felt Tip Pens	Foam-Filled Carrying Case
Evidence Seals (250 Seals/Roll)	Static-Dissipative Grounding Kit w/Wrist Strap
Plastic Evidence Bags (100 Bags)	Foam-Filled Disk Transport Box
Evidence Labels (100 Labels)	Computer Dusting System (Air Spray)
Evidence Tape--2" X 165'	Small Computer Vacuum
Tool Kit containing: Screwdriver Set (inc. Precision Set) Torx Screwdriver Set 25' Tape Measure Razor Knife Nut Driver Pliers Set LAN Template Probe Set Neodymium Telescoping Magnetic Pickup Allen Key Set Alligator Clips Wire Cutters Small Pry Bar Hammer Tongs and/or Tweezers	Printer and Ribbon Cables 9 Pin Serial Cable 25 Pin Serial Cable Null Modem Cable Centronics Parallel Cable 50 Pin Ribbon Cable LapLink Parallel Cable Telephone Cable for Modem
Cordless Driver w/Rechargeable Batteries (2)	Batteries for Camcorder, Camera, Tape Recorder, etc. (AAA, AA, 9-volt)
Pen Light Flashlight	
Magnifying Glass 3 1/4"	
Inspection Mirror	

Exhibit 2. Investigative and Forensic Tools (continues)

Computer Supplies	Software Tools
Diskettes: 3 1/2" Diskettes (Double & High Density Format) 5 1/4" Diskettes (Double & High Density Format)	Sterile O/S Diskettes
Diskette Labels	Virus Detection Software
5 1/2" Floppy Diskette Sleeves	SPA Audit Software
3 1/2" Floppy Diskette Container	Little-Big Endian Type Application
CD-ROM Container	Password Cracking Utilities
Write Protect labels for 5 1/4" Floppies	Disk Imaging Software
Tape and Cartridge Media 1/4" Cartridges 4mm & 8mm DAT Travan 9-Track/1600/6250 QIC Zip Drives Jazz Drives	Auditing Tools Test Data Method Integrated Test Facility (ITF) Parallel Simulation Snapshot Mapping Code Comparison Checksum
Hard Disks IDE SCSI	File Utilities (DOS, Windows, 95, NT, Unix)
Paper 8 1/2 x 11 Laser Paper 80 Column Formfeed 132 Column Formfeed	Zip/Unzip Utilities
Miscellaneous Supplies	**Miscellaneous Supplies**
Paper Clips	MC60 Microcassette Tapes
Scissors	Camcorder Tapes
Rubber Bands	35mm Film (Various Speeds)
Stapler and Staples	Polaroid Film
Masking Tape	Graph Paper
Duct Tape	Sketch Pad
Investigative Folders	Evidence Checklist
Cable Ties/Labels	Blank Forms -- Schematics
Numbered and Colored Stick-on Labels	Label Maker Labels

Exhibit 2. (continued)

on the facts. It should not include the investigator's opinions, unless he is an expert. Keep in mind that all documentation related to the investigation is subject to discovery by the defense, so be careful about what is written down!

COMPUTER FORENSICS

Computer forensics is the study of computer technology as it relates to the law. The objective of the forensic process is to learn as much about the

suspect system as possible. This generally means analyzing the system using a variety of forensic tools and processes. Bear in mind that the examination of the suspect system may lead to other victims and other suspects. The actual forensic process will be different for each system analyzed, but the following guidelines should help the investigator/analyst conduct the forensic analysis.

There are many tools available to the forensic analyst to assist in the collection, preservation and analysis of computer-based evidence. The make-up of a forensic system will vary from lab to lab, but at a minimum, each forensic system must have the ability to:

- Conduct a Disk Image Backup of the Suspect System
- Authenticate the File System
- Conduct Forensic Analysis in a Controlled Environment
- Validate Software and Procedures

Before analyzing any system it is extremely important to protect the systems and disk drives from static electricity. The analyst should always use an anti-static or static-dissapative wristband and mat before conducting any forensic analysis.

Conduct a Disk Image Backup of the Suspect System

A disk image backup is different from a file system backup in that it conducts a bit level copy of the disk, sector-by-sector, rather then merely copying the system files. This process provides the capability to back up deleted files, unallocated clusters and slackspace. The backup process can be accomplished by using either disk imaging hardware, such as the ImageMaster 1000, or through a variety of software programs. Most of these programs run under DOS or Windows and will back up most any type of hard disk or floppy disk, regardless of the operating system. The image backup process is conducted as depicted in Exhibit 3.

Authenticate the File System

File system authentication helps to ensure the integrity of the seized data and the forensic process. Before actually analyzing the suspect disk, a message digest is generated for all system directories, files and disk sectors. A message digest is a signature that uniquely identifies the content of a file or disk sector. It is created using a one-way hashing algorithm. In the past a 32-bit CRC32 algorithm was used, but due to the advancements in cryptographic research and along with more powerful machines, two more advanced, one-way hashing algorithms are now being used. MD5 is a 128-bit hash, while SHA is a 160-bit hash. These strong cryptographic hashing algorithms virtually guarantee the integrity of the processed data. Doing

Step	Disk Image Backup Procedure
1	Remove the internal hard disk(s) from suspect machine and label (if not already done). Make a note of which logical disk you are removing. Follow the ribbon cables from the disk to the I/O board to accomplish this task. It is a good idea to photograph the inside of the system including the connections to the I/O boards and disk drives.
2	Identify the type of disk (i.e. IDE or SCSI). Identify the make and model.
3	Identify the disk capacity. Make a note of cylinders, heads and sectors.
4	Place each disk, one at a time, in a clean forensic examination machine as the next available drive. Beware that the suspect disk may have a virus (keep only the minimal amount of software on the forensic examination machine). Note, if you are using a hardware-based disk duplication method (i.e. ImageMaster 1000), then this step is not necessary.
5	Backup (Disk Image) the suspect disk(s) to tape—Make at least 4 copies of each suspect disk
6	Check the disk image backup logs to make sure that there were no errors during the backup process.
7	Place the original suspect disk(s), along with one of the backup tapes, and backup logs, in the appropriate container. Seal, mark and log into evidence.
8	Return a copy of the original disk to the victim (if applicable)
9	Use the last two copies for the forensic analysis (one is used for file authentication)

Exhibit 3. Image Backup Process

this now will help refute any argument by the defense, that the evidence was tampered with.

The concept of a one-way hash, using MD5 for example, is that a file is read into memory. The file is then processed, bit by bit, until it reaches the end of the file. The hashing process creates a 128-bit signature for the file that is based upon the file content. Even the change of a single bit will change the signature produced by the hashing algorithm. The significance of the one-way hash is that it only works one way. Knowledge of the hash value can not produce the file content itself.

The only problem with executing the authentication process is that it will change the file's last access time. The mere process of reading the file, to produce the hash value will change this time. That is why a separate backup is used for the authentication process.

Conduct Forensic Analysis in a Controlled Environment

After restoring at least one of the backup tapes to a disk, of equal capacity to the original disk (identical disk, if possible), the restored data should be analyzed. This should be done in a controlled environment on a forensic system. Everything on the system must be checked, starting with the file system and directory structure. The analyst should create an organizational chart of the disk file system and then inventory all files on the disk.

There are a number of commercially available utilities that allow the analyst to quickly create a directory tree, list system files, identify hidden files, and to conduct keyword searches. The analyst should make notes during each step in the process, especially when restoring hidden or deleted files, or modifying the suspect system (i.e. repairing a corrupted disk sector w/Norton Utilities). The analyst should also note that what may have happened on the system may have resulted from error or incompetence rather from than a malicious user. It is a good idea to check for viruses at this point to, first, note their existence, and secondly, to avoid potential contamination.

Since forensic analysis can be a laborious and time-consuming process, it is sometimes better to distribute the workload to, both, other analyst and case agents. Since it would be too costly to have multiple forensic systems and to have to replicate the suspect data on multiple hard drives, it may be more effective to make CD copies of the hard disk contents that can be distributed and analyzed by different individuals. This is certainly more cost effective and may possibly accelerate the analysis process.

When using CD-R or WORM (Write Once Read Many) technology, the data should be structured in way that will enhance the forensic process. One method of data organization that works quite well is to create a logical directory structure that will store and organize all data from the target disk. This should include all files and directories from the original file structure, deleted files, hidden files, data in slack space, data in unallocated space, compressed data, encrypted data and data generated from search results.

To initiate this process, the analyst should copy (file copy) the complete file structure, starting from the root directory, from the image copy to a newly created hard disk partition. This type of copy will not pick up deleted files, data in slack space, or data in unallocated space, therefor the analyst must manually copy this data from the target system to the new disk partition. Before copying this data, individual sub-directories must be created for each data type: DELETED, SLACK, UNALLOC. The file copy process will copy the swap file, but it may be best to move the file to a SWAP sub-directory. The next step in the process is to review the information in the original file system, looking for files with hidden file attributes, compressed files, encrypted files and files that meet the criteria of key-word searches. These, too, should be copied to specific directories, so that later it is understood where the data came from. The following directories should be created to store and organize this data: HIDDEN, COMPRESS, ENCRYPT, and SEARCH.

The final process is to use a disk editor utility to look for “BAD” clusters that have data in them and to run key-word searches at the disk editor level

(below the operating system). Any data found during this analysis should be copied to the newly created file system. A BAD sub-directory can be created under the HIDDEN sub-directory and an EDITOR sub-directory can be created under the SEARCH sub-directory. Once the new file system is populated with all the data, the information can be burnt into a CD-R or WORM drive. This information can then be made available to other forensic analysts or case agents. If damaging evidence is discovered upon review of the data stored on the CD-R or WORM drive, the original information can easily be recovered from the original image copy.

A quick background on file times should be given before continuing on. Most computer systems, including Windows 95, NT and UNIX store three values for file times: creation time, update time, last access time. Any or all of these file times may have an impact on the investigation. The access time is the one most susceptible to modification because any read to access to the file changes this time. The image backup will not change this time, but the file authentication process will! The creation time is the time the file was originally created. It is not accessible from the file manager or the DIR command. The update time is the time the file was last modified (written to). This is the time the file manager displays. The last access time is recorded whenever any other program or command, including read, copy, etc touches the file. This time is also not accessible from the file manager but can be seen in the under file properties.

When searching through files and directories, the first things to look for are file names or document content that have case-relevant names. For example, if the case you are working is an espionage or theft of trade secrets case, then look for file names with the word (or partial word) of the trade secret item itself. If trade secret was related to the release of a new, database software product, called SplitDB, then look for files with the name "split.xls," "db.doc," or "database.ppt." Another search may find the word "split," "db," or "database" in the body of a word processing document (i.e. a hidden file named sys.dll with the following phase, "For this database structure to work effectively..."). Another indicator that something is afoul, is when the file extension doesn't match the file signature. All files have a signature, which identify the type of file, somewhere in the first 50 characters of the file. This file signature normally correlates to a particular file extension. For example, a bitmap graphic file normally has a file extension of .bmp and a file signature of BM as the first two bytes of the file. If these two items do not match up, then it may mean that someone modified the file extension to hide the presence of the file. A pedophile can use this technique to hide a bitmap image containing child pornography in the c:\windows\system directory as system.dll. A cursory review of the system may miss this file completely, thinking that it is a Windows system file, when in fact it is damaging evidence.

Search Tools. There are many search tools that can assist the forensic analyst in his endeavor to locate damaging evidence. Most of these tools are commercial off-the-shelf (COTS) applications that were created for some other reason, other than forensics. It just so happens that these applications work well in a forensic environment. Norton Utilities, although not the end all, is a must for all forensic investigators. Norton provides file searching utilities, disk editor functions, data recovery, etc. Some other tools are listed below:

- Quick View Plus
- Expert Witness
- Computer Forensics Laboratory
- Drag and View
- Rescue Professional
- Super Sleuth
- Outside/In

Searching for Obscure Data. Once the basic analysis is complete, the next step is to conduct a more detail analysis of more obscure data. It may be necessary to use forensic data recovery techniques to locate and recover:

- Hidden files
  - Hidden by attributes
  - Hidden through steganography
  - Hidden in slack space
  - Hidden in good clusters marked as BAD
- Modifying the size of the file in the directory entry
- Hidden directories
- Erased or deleted files
- Reformatted media
- Encrypted data
- Overwritten (wiped) files

The fact that a file is hidden is a good indicator of its evidentiary value. If someone took the time to hide the file, it was probably hidden for a reason. The simplest way to hide a file is to alter the file attribute to Hidden, System, or Volume Label. Files with these attributes do not normally appear in a DIR listing or even in the Windows file manager. Simply changing the attribute back will make the file accessible. Files with the Hidden attribute set are usually further hidden in a hidden directory. An example of hidden directory would be the .directory in UNIX or creating a directory with the ALT 255 character in a Windows or DOS system. Many times these hidden directories are deeply nested to avoid discovery. The “chkdsk” utility will display the number of hidden files on the DOS system, while Norton Utilities will display a listing of the hidden file and its location.

A file can also be hidden in slack space. Slack space is the area left over in a cluster that is not utilized by a file. For example, if a 2K file is stored in a 32K cluster, then there is 30K of slack space, which may contain data from a previous file. This area can also be used to hide data. A cluster, which is the basic allocation unit, is the smallest unit of space that DOS uses for a file. The amount of slack space for a given file varies based upon the file size and cluster size. The cluster size usually expands as hard disk capacity increases.

Another, more elaborate way to hide data is to first, write data to a file in the normal way. When this is complete, the suspect can use a disk editor to ascertain the sector and cluster of the newly created file, go to that cluster and mark the cluster as BAD. When the operating system sees a BAD cluster, it simply ignores the area. The data is still present on the disk even though it can not be accessed. The analyst will need to locate the cluster by using a sector searching utility, then go to the specific cluster and remove the BAD label.

Files and directories can also be deleted. But when DOS or Windows deletes a file, it only changes the first character of the file name to 0xE5, which merely makes the file space available. The file is not actually removed. The data in the cluster previously allocated by the file is still available until overwritten by a new file. On DOS and Windows systems, the analyst can use the un-erase utility to recover deleted files. These utilities only recover the first cluster that the file occupied. If the file occupied multiple clusters, this data may be lost, as the cluster chain is no longer available. Cluster chains can be re-built although not reliably.

If the disk is formatted, the analyst can attempt to use the "un-format" command in the DOS or Windows environment. If the disk has been wiped, which is also known as shredding, the data is not easily recoverable. The cost of recovery is usually exorbitant, far exceeding the initial loss.

Steganography. Steganography is the art of hiding communications. Unlike encryption, which utilizes an algorithm and a seed value to scramble or encode a message in order to make it unreadable, Steganography makes the communication invisible. This takes concealment to the next level — that is to deny that the message even exists. If a forensic analyst were to look at an encrypted file, it would be obvious that some type of cypher process has been used. It is even possible to determine what type of encryption process was used to encrypt the file, based upon a unique signature. However, Steganography hides data and messages in a variety of picture files, sound files and even slack space on floppy diskettes. Even the most trained security specialist or forensic analyst may miss this type of concealment during a forensic review.

Steganography simply takes one piece of information and hides it within another. Computer files, such as images, sound recordings, and slack space contain unused or insignificant areas of data. For example, the least significant bits of a 24-bit bitmap image can be used to hide messages, usually without any material change in the original file. Only through a direct, visual comparison of the original and processed image can the analyst detect the possible use of Steganography. Since many times the suspect system only stores the processed image, the analyst has nothing to use as a comparison and generally has no way to tell that the image in question contains hidden data. There is research underway that will help in the forensic process when dealing with Steganography. New tools are being developed that will look at the file contents to determine if there is a Steganographic signature within the file. But with over 25 different types of Steganography being used today, this new research may take some time.

Review Communications Programs. A good source of contact and associate information can many times be found on-line. Since many technically competent individuals use technology for the same reasons businesses do, electronic Rolodexes, databases of contacts, and communication programs should be searched. Applications like Microsoft Outlook, ACT and others can be tremendously beneficial during an investigation to link your suspect to other individuals or businesses. Some computers store Caller ID files, while others may contain war dialer (or demon dialer) logs. Review communications programs, such as Procomm, to ascertain if any numbers are stored in the application.

Microprocessor Output. One final note, before moving on to the next step in the forensic process, is to understand that not all microprocessors are created equal. If a forensic analyst is forced to dump the contents of a file in binary or hexadecimal format, he must not only understand how to read these hieroglyphic notations, but must know the type of microprocessor that produced the output. For example, the Intel 30286 is a 16-bit, little endian processor. A 16-bit microprocessor is capable of working with binary numbers of up to 16 places or bits. That translates to the decimal number 65,536. The Intel 30486 and newer Pentium processors are 32-bit computers, capable of handling binary numbers of up 32 bits or up to the decimal number 4,294,967,296. The little endian attribute of the Intel chip signifies the byte, not bit, ordering sequence. In this case the bytes are reversed, where the high order byte(s) is stored low order byte location. A big endian processor does not reverse the byte order. It is important to understand that the same value dumped out on two different systems may produce different results.

Reassemble and Boot Suspect System (with Clean Operating System)

The next step in the process is to reassemble the suspect system, using one of the copies of the suspect disk. Place a clean copy of the forensic operating system (usually DOS or Windows) into the floppy drive. Start the boot process and enter the CMOS setup. Check the CMOS to make sure that the boot sequence looks to the floppy drive first, then the hard disk second. This will allow the investigator to boot from the clean operating system diskette. Also, if the system is password protected at the CMOS level, remove and reinstall or short out the CMOS battery. Continue with the boot process and pay particular attention to the Boot-up process, looking for a modified BIOS or EPROM.

It is very important to boot from a clean operating system, as the target system utilities may contain a Trojan Horse or Logic Bomb that will do other than what's intended. (e.g. Modified command.com—conducting a Delete with the Dir command). The first thing to do once the system is booted is to check the system time. This time, even if not accurate, will give the analyst or investigator a reference for all file times. After the system time is obtained, run a complete Systems Analysis Report. This report should, at a minimum, provide the following:

- System Summary—contains basic system configuration
- Disk Summary
- Memory Usage w/Task List
- Display Summary
- Printer Summary
- TSR Summary
- DOS Driver Summary
- System Interrupts
- CMOS Summary
- Listing of all environment variables as set by Autoexec.bat, config.sys, win.ini, system.ini, etc.

Audit trails can be viewed any time subsequent to the image backup, but before a through analysis can be completed, the analyst will need a time reference, which is obtained from booting the suspect system. Check the audit logs for system and account activity. Check with the victim organization to ascertain if the Audit logs are used in the normal course of business. The following questions must be asked:

- Is there a corporate security policy on how the logs are to be used? If so, has the policy been followed?
- What steps have been taken to ensure the integrity of the audit trail?
- Has the audit trail been tampered with? If so, when?

Boot Suspect System (with Original Operating System)

The next step in the forensic process is to boot the target system using the original, target system operating system. This is done to see if any rouge programs were left on the system. The analyst should let the system install all background programs (Set by autoexec.bat and config.sys). Once this has been done, the analyst should check what programs (including TSR's) are running and what system interrupts have been set. The goal is to learn if there are any Trojan Horses or other rouge programs, such as keystroke monitors, activated. Execute some of the basic operating system commands to see if the command.com file had been altered.

Searching Backup Media

Remember that if the data is not on the hard disk, it may be on backup tapes or some other form of backup media. Even if the data was recently deleted from the hard disk, there may be a backup that has all of the original data. Many times a "snapshot" of the system is taken on a weekly or monthly basis and saved in the long term archives for disaster contingency purposes. Search for PCMCIA flash disks, floppy diskettes, optical disks, Ditto tapes, Zip and Jazz cartridges, Kangaroo drives, or any other form of backup media. Restore and review all data. Many organizations store backups off-site, and although a warrant may be required to obtain the media, don't forget to ascertain if this practice is being done. Before analyzing floppy diskettes, always write-protect the media.

Searching Access Controlled Systems and Encrypted Files

During a search the investigator may be confronted with a system which is secured physically and/or logically. Some physical security devices, such as CPU key locks, prevent only a minor obstacle, whereas other types of physical access control systems may be harder to break.

Logical access control systems may pose a more challenging problem. The analyst may be confronted with a software security program that requires a unique user-name and password. Some of these systems can be simply bypassed by entering a control-c or some other interrupt command. The analyst must be cautious that any of these commands may invoke a Trojan horse routine that may destroy the contents of the disk. A set of "password cracker" programs should be part of the forensic tool-kit. The analyst can always try to contact the publisher of the software program in an effort to gain access. Most security program publishers leave a back door into their systems.

The investigator should look around the suspects work area for documents that may provide him with a clue to the proper user-name/password combination. Check desk drawers, the suspect's Rolodex, acquaintances,

friends, etc. It may be possible to compel a suspect to provide access information. It is a good idea to first ask the suspect for his password, before going through the process of compelling him to do so. The following cases set precedence for ordering a suspect, whose computer is in the possession of law enforcement, to divulge password or decryption key:

- Fisher v US (1976), 425 US 391, 48 LED2 39
- US v Doe (1983), 465 US 605, 79 LED2d 552
- Doe v US (1988), 487 US 201, 101 LED2d 184
- People v Sanchez (1994) 24 CA4 1012

The caveat is that the suspect might use this opportunity to command the destruction of potential evidence. The last resort may be that the system needs to be hacked. This can be done as follows:

- Search for passwords written down (It may be part of the evidence collected)
- Try words, names or numbers that are related to the suspect
- Call the software vendor and request their assistance (Some charge for this)
- Try to use password cracking programs which are readily available on the net
- Try a brute force or dictionary attack

LEGAL PROCEEDINGS

A brief description of the legal proceedings that occur subsequent to the investigation are necessary so the victim and the investigative team understand the full impact of their decision to prosecute. The post-incident legal proceedings generally result in additional cost to the victim, until the outcome of the case, at which time they may be reimbursed.

Discovery and Protective Orders

Discovery is the process whereby the prosecution provides all investigative reports, information on evidence, list of potential witnesses, any criminal history of witnesses, and any other information except how their going to present the case to the defense. Any property or data recovered by law enforcement will be subject to discovery if a person is charged with a crime. However, a protective order can limit who has access, who can copy, and the disposition of the certain protected documents. These protective orders allow the victim to protect proprietary or trade secret documents related to a case.

Grand Jury and Preliminary Hearings

If the defendant is held to answer in a preliminary hearing or the grand jury returns an indictment, a trial will be scheduled. If the case goes to trail,

interviews with witnesses will be necessary. The victim company may have to assign someone to work as the law enforcement liaison.

The Trial

The trail may not be scheduled for some time based upon the backlog of the court that has jurisdiction in the case. Additionally, the civil trial and criminal trial will occur at different times, although much of the investigation can be run in parallel. The following items provide tips on courtroom testimony:

- The prosecutor does not know what the defense attorney will ask.
- Listen to the questions carefully to get the full meaning and to determine that this is not a multiple part or contradictory question.
- Do not answer quickly; Give the prosecutor time to object to the defense questions that are inappropriate, confusing, contradictory or vague.
- If you do not understand the question, ask the defense attorney for an explanation, or answer the question by stating "I understand your question to be . . ."
- You can not give hearsay answers. This generally means that you can not testify to what someone has told you.
- Do not lose your temper and get angry as this may affect your credibility.
- You may need to utilize expert witnesses.

Recovery of Damages

To recover the costs of damages, such as reconstructing data, re-installing an uncontaminated system, repairing a system, or investigating a breach, you can file a civil law suit against the suspect in either Superior Court or Small Claims Court.

Post Mortem Review

The purpose of the Post Mortem review is to analyze the attack and close the security holes that led to the initial breach. In doing so, it may also be necessary to update the corporate security policy. All organizations should take the necessary security measures to limit their exposure and potential liability. The security policy should include an:

- Incident Response Plan
- Information Dissemination Policy
- Incident Reporting Policy
- Electronic Monitoring Statement
- Audit Trail Policy

- Inclusion of a Warning Banner—This should:
  - Prohibit unauthorized access and;
  - Give notice that all electronic communications will be monitored

One final note is that many internal attacks can be avoided by conducting background checks on potential employees and consultants.

SUMMARY

As you probably gleaned from this chapter, computer crime investigation is more an art than a science. It is a rapidly changing field that requires knowledge in many disciplines. But although it may seem esoteric, most investigations are based on traditional investigative procedures. Planning is integral to a successful investigation. For the internal investigator, an Incident Response Plan should be formulated prior to an attack. The Incident Response Plan will help set the objective of the investigation and will identify each of the steps in the investigative process. For the external investigator, investigative planning may have to happen post incident. It is also important to realize that no one person will have all the answers and that teamwork is essential. The use of a Corporate CERT Team is invaluable, but when no team is available, the investigator may have the added responsibility of building a team of specialists.

The investigator's main responsibility is to determine the nature and extent of the system attack. From there, with knowledge of the law and forensics, the investigative team may be able to piece together who committed the crime, how and why the crime was committed, and maybe more importantly, what can be done to minimize the potential for any future attacks. For the near term, convictions will probably be few, but as the law matures and as investigations become more thorough, civil and criminal convictions will increase. In the mean time, it is extremely important that investigations be conducted so as to better understand the seriousness of the attack and the overall impact to business operations

Finally, to be successful, the computer crime investigator must, at a minimum, have a thorough understanding of the law, the rules of evidence as they relate to computer crime, and computer forensics. With this knowledge, the investigator should be able to adapt to any number of situations involving computer abuse.

148

What Happened?

Kelly J. Kuchta, CPP, CFE

Envision coming across the dead bodies and the related carnage of a crime scene at night. It is a place of chaos and confusion, smoke, shadows, and debris. Victims wander around dazed and stumble into each other; bystanders and the curious mill around in anxious speculation and anticipation. No one really knows what happened, or even when. They just know that it has happened. The authorities are supposedly on the way. Then suddenly, someone runs up to you and puts you in charge. Why? Because you know the neighborhood.

This sounds like a nightmare and in reality, it is — especially when the crime scene is somewhere in your network and involves your information systems.

I use the crime scene analogy because forensics issues involving information systems are like a crime scene. From decades of watching TV cop shows (or the O.J. trial), most people know that you do not trample over evidence because valuable information and clues about the crime could be inadvertently destroyed or tainted. At the crime scene, we know to check to see whether there is anyone who needs medical assistance and then just pick up the phone and dial 911 — thereby letting those who have the requisite training, background, and expertise analyze the crime scene and work it.

What do you do when you find out later that something bad has happened in your network and you need information about an event in the past? If someone in your organization has the appropriate skills, that person will appreciate early notice about the incident and your efforts to leave the crime scene intact.

As emergency personnel will often tell you, the initial decisions made following an incident have the greatest impact on the outcome. Today's information systems usually do not leave many outward signs that something is terribly wrong. Actually, it is the people that using the system who will provide insight into incidents.

With increasing frequency, we are seeing theft of confidential data and other misuse of computers. The best advice I can give people and corporations in handling future incidents is to develop a "behavioral pattern matrix" (see Exhibit 148.1) of personnel security-related events that need closer scrutiny (more on this later) and when in doubt preserve the evidence by removing the hard drive of the victimized computer. Hard drives are inexpensive, and the amount of downtime from pulling a hard drive and installing a new hard drive with your organizations' standard loadset is minimal. The effort to do this can save the organization money and headaches.

Consider the employee who resigns after working in a sensitive area of your business. If anything illegal or unethical has taken place, you will probably not find out about it until 30 to 60 days after the employee has left, if ever. I suggest saving the hard drives from laptops or desktops of resigned and terminated employees for a minimum of 60 to 90 days and longer if possible. At the end of this period of time, cleanse the disk and put it back into production. Why? Because once the hard drive and the residing data is reformatted and placed back into circulation, the chances of recovering any usable information from that hard drive for forensic analysis will be next to impossible and limited by the amount of time and money you have to spend.

In most instances when evidence is tainted, it is through ignorance, not through intentional acts of deception. I have witnessed corporations and individuals who attempt to use their investigative skills after an incident by having the system administrator look for clues or evidence. In one case, they were able to find incriminating data; however, after finding the information, they opened the file and copied it to a floppy disk. This action modified the key dates and contaminated the electronic evidence, preventing its use in a court of law.

EXHIBIT 148.1 Sample of Behavioral Matrix

Employee	Risk Score **Yes = 1 No = 0**	Weight 100 (percent)	Weighted Score
Did the employee work with sensitive information?	1	5	0.05
Was the separation hostile?	0	20	0
Did the employee go to work for a competitor?	1	20	0.2
Could the employee have been involved in any unexplained events?	0	5	0
Was the separation unexpected?	0	10	0
Is there a chance that the employee's actions might be involved in litigation?	0	25	0
Has the entity been the target of intelligence gathering?	1	15	0.15
Evidence preservation score			40 percent

Guidelines for evidence preservation
0 to 24 percent no apparent need to preserve evidence
25 to 49 percent good reason to preserve evidence
>49 percent strong reason to preserve evidence
This is a sample behavioral matrix you can customize to your needs.

Computer forensic professionals view the system dates as vital pieces of information. Created, last written, and last access dates are used to establish a chain of events that give important insight into what happened in the past. Computer forensics methodology dictates that computer forensics professionals must not change any piece of evidence, including the dates. When reviewing the data on a suspected system, great lengths are taken to prevent the operating system from writing to the hard drive. Even if you are not a computer forensics professional, you owe your organization the opportunity to fight back by preserving the original evidence.

When a computer is started, right away the operating system changes or modifies a large number of file dates on the system. The actual number of files may vary depending on what type of system, anti-virus applications, or network protocols the organization is using. A typical Windows 98 machine will have over 12,000 files loaded on it. During the start-up process, hundreds of these files may be changed during the POST (power on self test) process. If the anti-virus application is set to inoculate any viruses found, having the malicious code removed will modify the file. This process will change the last access and last written dates.

To keep as many options available as possible, consider setting the hard drives aside for a reasonable amount of time. If you think that putting each hard drive in a probationary period will not work because of the potential expense, consider doing it on a limited basis. Earlier I mentioned developing a behavioral pattern matrix for exiting employees who might give you reason to preserve their hard drives. The objective is to find predictors that would indicate the future need to review the hard drive of the computer.

My experience has shown that human behavior is a key predictor that must be considered at a digital crime scene. Each organization will experience different behaviors that constitute a warning. Each organization will need to develop its own behavior pattern that fits its culture. In this case, past events can be good indicators of future events. The sources of information to consider should come from human resources, corporate investigations, information security, as well as the legal department and the business units themselves.

Some factors that might weigh into your behavioral pattern matrix are as follows: Did the employee work with sensitive data? Was the resignation a surprise? Is the termination likely to result in legal proceedings? Is the employee going to work for a competitor? Have there been any events that are of concern to the organization in which the employee might have been involved? Was the employee vague about why he or she was leaving? The answer "yes" to any of these questions should trigger at least considering saving the hard drive for a reasonable period of time.

I often hear, "I knew there was something suspicious about the person!" when working on employee or former employee issues. There are other signs that are frequently overlooked, but by considering all the facts,

organizations realize in retrospect that they missed the warning signs. The warnings are generally spread out over multiple areas, such as human resources, corporate investigations, business units, and information security.

Human resources and the business units hold the keys about the behavior of the individual and the possible reason for the departure of the employee. Corporate investigations might be able to provide insight on external events and intelligence information. This could include events under investigation but not publicly known, attempts by competitors to gain proprietary information, and other possible related matters. Information security might have some information about suspicious behavior the individual demonstrated recently. Examples of suspicious behavior could be linked to attempting access to restricted information, copying large amount of data, allegations of technology misuse, or browsing suspicious Internet sites.

The best process I have witnessed was to have the human resources personnel in charge of the employee exit process give notice to the three groups listed previously. They should give each group a reasonable amount of time to respond that they would like the hard drive held for the proscribed period of time or want immediate analysis relating to a specific event. Of course, Human Rresources personnel might make this request themselves based on their information.

Do not forget the importance of having an "acceptable use" policy to guide new employees. As exiting employees are getting ready to turn in their PCs, they should be instructed on what they can or cannot do. Depending on business needs and culture, you might establish a policy that restricts the employee's ability to use wipe utilities (especially nonstandard products) or other products that could sabotage forensics results. Although this is a difficult subject to deal with in corporate America, it is vitally important. On more than one occasion, I have seen cases in which a mildly disgruntled employee deliberately erased valuable client information and used a wipe utility to make the information unrecoverable. You should make a conscious decision about this issue, even if it is to have no policy on this issue!

To develop a process that is customized to your organization, consider getting input from the above-named individuals and your legal counsel. If the employee is part of a unionized labor force, special rules may apply. There may also be special considerations based on state law or if the organization fulfills government contracts. The preserved evidence is probably discoverable with a subpoena. Your legal counsel can help you determine what legal requirements you need to adhere to.

Assume that you have adopted a process similar to the one outlined. The organization has made the decision to preserve a hard drive. How do you go about it? The major concerns are establishing a chain of custody, documenting specific details, and securing the hard drive. Each of these areas is vitally important if there is a chance that the electronic evidence you have preserved will be presented in a court of law.

You must establish a chain of custody to prove authentication and refute allegations of evidence tampering. Many defense attorneys have successfully argued that if you cannot prove that the evidence has been under your control, you cannot prove that it has not been changed or modified to construct the incriminating evidence. To establish the chain of custody, you must document possession from the point of acquiring it until the matter is resolved. This includes an appeals process through the court system.

Part of the documentation process will be to identify as many details about the original PC that the evidence came from as possible. This is important because an analysis completed later will go much smoother if a few key pieces of information are known. You should document the following:

- What types of operating system are on the hard drive?
- What are other systems specifications (RAM, SCSI, or IDE; processor type)?
- Are any partitions likely to be found?
- What applications are known to be on the hard drive?
- What, if any, encryption was used?
- Is there a list of any known passwords, keys, or certificates?
- To what systems did the owner of the hard drive have access?
- What type of system did the hard drive come from (manufacturer, model)?
- Is there a history of hardware problems, including any maintenance logs?

Having the answers to these questions will make the forensics analysis a much faster and efficient process.

A master log should accompany the evidence from the time it is acquired. It should include date and time, a detailed description of the evidence, and who seized the evidence. The log should also include a transfer-of-custody section, which should include reason for transfer, method (hand-delivered or courier), released by and date (signature and date of person transferring custody), and received by and date (signature and date of

person taking custody). People listed as having custody of the evidence will need to demonstrate that the evidence was under their control and secured to prevent tampering.

A secured location is a lockable container that has limited access. It can be a file cabinet with locks, a safe, an evidence locker, or even a room with a lock. The best possible scenario is to have only one person with access to the evidence. If that is not possible, the evidence must be stored in a limited area and everyone with access to the area should be documented. The more persons with access to evidence will mean more people testifying that they did not modify the evidence. It is easier to provide a lockable container with single access than one with multiple access. If you will be securing evidence on a regular basis, consider purchasing an evidence locker. Your evidence locker should also include a master log of evidence it holds. When evidence is stored, it should be logged in. Each time it is removed, custody should be transferred out to the individual removing it. The design of the log outlined above can also be utilized here. The purpose of this log is to document each and every time the evidence locker is accessed as well as to provide supporting documentation about particular evidence.

If it is necessary to send evidence to another location, I recommend using a courier service that can provide documentation of its custody. This should include tracking forms and numbers. Most of the traditional delivery services provide this service. The senders should seal the package themselves and the recipients observe that the package has not been breached. For additional protection, it is suggested that the evidence is sealed in a container so that the recipient can attest that the document has not been tampered with. Reasonable steps should be taken to protect the evidence during shipping. The evidence will do little good if it has been damaged.

Taking these steps will increase the odds of determining what happened in the past. Understanding history to change the future is the ultimate goal. To understand the history we must have good information. To preserve information, you do not need to be a computer forensics professional — just understand the process and why it is important. Also, practice techniques that will work for your company and be prepared to have good information on "what happened."

Computer Abuse Methods and Detection

Donn B. Parker

This chapter describes 17 computer abuse methods in which computers play a key role. Several of the methods are far more complex than can be described here in detail; in addition, it would not be prudent to reveal specific details that criminals could use. These descriptions should facilitate a sufficient understanding of computer abuse for security practitioners to apply to specific instances. Most technologically sophisticated computer crimes are committed using one or more of these methods. The results of these sophisticated and automated attacks are loss of information integrity or authenticity, loss of confidentiality, and loss of availability or utility associated with the use of services, computer and communications equipment or facilities, computer programs, or data in computer systems and communications media. The abuse methods are not necessarily identifiable with specific statutory offenses. The methods, possible types of perpetrators, likely evidence of their use, and detection and prevention methods are described in the following sections.

EAVESDROPPING AND SPYING

Eavesdropping includes wiretapping and monitoring of radio frequency emanations. Few wiretap abuses are known, and no cases of radio frequency emanation eavesdropping have been proved outside government intelligence agencies. Case experience is probably so scarce because industrial spying and scavenging represent easier, more direct ways for criminals to obtain the required information.

On the other hand, these passive eavesdropping methods may be so difficult to detect that they are never reported. In addition, opportunities to pick up emanations from isolated small computers and terminals, microwave circuits, and satellite signals continue to grow.

One disadvantage of eavesdropping, from the eavesdropper's point of view, is that the perpetrators often do not know when the needed data will be sent. Therefore, they must collect relatively large amounts of data and search for the specific items of interest. Another disadvantage is that identifying and isolating the communications circuit can pose a problem for perpetrators. Intercepting microwave and satellite communications is even more difficult, primarily because complex, costly equipment is needed for interception and because the perpetrators must determine whether active detection facilities are built into the communications system.

Clandestine radio transmitters can be attached to computer components. They can be detected by panoramic spectrum analysis or second-harmonic radar sweeping. Interception of free-space radiation is not a crime in the United States unless disclosure of the information thus obtained violates the Electronic Communications Privacy Act of 1986 (the ECPA) or the Espionage Act. Producing radiation may be a violation of FCC regulations.

Intelligible emanations can be intercepted even from large machine rooms and at long distances using parametric amplifiers and digital filters. Faraday-cage shielding can be supplemented by carbon-filament adsorptive covering on the walls and ceilings. Interception of microwave spillage and satellite footprints is different because it deals with intended signal data emanation and could be illegal under the ECPA if it is proved that the information obtained was communicated to a third party.

Spying consists of criminal acquisition of information by covert observation. For example, shoulder surfing involves observing users at computer terminals as they enter or receive displays of sensitive information (e.g., observing passwords in this fashion using binoculars). Frame-by-frame analysis of video recordings can also be used to determine personal ID numbers entered at automatic teller machines.

Solutions to Eavesdropping and Spying

The two best solutions to eavesdropping are to use computer and communications equipment with reduced emanations and to use cryptography to scramble data. Because both solutions are relatively costly, they are not used unless the risks are perceived to be sufficiently great or until a new level of standard of due care is met through changes in practices, regulation, or law.

In addition, electronic shielding that uses a Faraday grounded electrical conducting shield helps prevent eavesdropping, and physical shielding helps prevent spying. Detecting these forms of abuse and obtaining evidence require that investigators observe the acts and capture the equipment used to perpetrate the crime.

Potential Perpetrators	Methods of Detection	Evidence
• Communications technicians and engineers • Communications employees	• Voice wiretapping methods • Observation • Tracing sources of equipment used	• Voice wiretapping evidence

Exhibit 1. Detection of Eavesdropping

Eavesdropping should be assumed to be the least likely method used in the theft or modification of data. Detection methods and possible evidence are the same as in the investigation of voice communications wiretapping. Exhibit 1 summarizes the potential perpetrators, detection, and evidence in eavesdropping acts.

SCANNING

Scanning is the process of presenting information sequentially to an automated system to identify those items that receive a positive response (e.g., until a password is identified). This method is typically used to identify telephone numbers that access computers, user IDs, and passwords that facilitate access to computers as well as credit card numbers that can be used illegally for ordering merchandise or services.

Computer programs that perform the automatic searching, called demon programs, are available from various hacker electronic bulletin boards. Scanning may be prosecuted as criminal harassment and perhaps as trespassing or fraud if the information identified is used with criminal intent. For example, scanning for credit card numbers involves testing sequential numbers by automatically dialing credit verification services. Access to proprietary credit rating services may constitute criminal trespass.

Prevention of Scanning

The perpetrators of scanning are generally malicious hackers and system intruders. Many computer systems can deter scanners by limiting the number of access attempts. Attempts to exceed these limits result in long delays that discourage the scanning process.

Identifying perpetrators is often difficult, usually requiring the use of pen registers or dialed number recorder equipment in cooperation with communication companies. Mere possession of a demon program may constitute possession of a tool for criminal purposes, and printouts from demon programs may be used to incriminate a suspect.

Potential Perpetrators	Methods of Detection	Evidence
• Authorized computer users • Hackers	• Audit log analysis • Password violations • Observation • Report by person impersonated	• Computer audit log • Notes and documents in possession of suspects • Pen register and records of number dialed • Witnesses • Access control package exception or violation reports

Exhibit 2. Detection of Masquerading

MASQUERADING

Physical access to computer terminals and electronic access through terminals to a computer require positive identification of an authorized user. The authentication of a user's identity is based on a combination of something the user knows (e.g., a secret password), a physiological or learned characteristic of the user (e.g., a fingerprint, retinal pattern, hand geometry, keystroke rhythm, or voice), and a token the user possesses (e.g., a magnetic-stripe card, smart card, or metal key). Masquerading is the process of an intruder's assuming the identity of an authorized user after acquiring the user's ID information. Anybody with the correct combination of identification characteristics can masquerade as another individual.

Playback is another type of masquerade, in which user or computer responses or initiations of transactions are surreptitiously recorded and played back to the computer as though they came from the user. Playback was suggested as a means of robbing ATMs by repeating cash dispensing commands to the machines through a wiretap. This fraud was curtailed when banks installed controls that placed encrypted message sequence numbers, times, and dates into each transmitted transaction and command.

Detection of Masquerading

Masquerading is the most common activity of computer system intruders. It is also one of the most difficult to prove in a trial. When an intrusion takes place, the investigator must obtain evidence identifying the masquerader, the location of the terminal the masquerader used, and the activities the masquerader performed. This task is especially difficult when network connections through several switched telephone systems interfere with pen register and direct number line tracing. Exhibit 2 summarizes the methods of detecting computer abuse committed by masquerading.

PIGGYBACK AND TAILGATING

Piggyback and tailgating can be done physically or electronically. Physical piggybacking is a method for gaining access to controlled access areas when control is accomplished by electronically or mechanically locked doors. Typically, an individual carrying computer-related objects (e.g., tape reels) stands by the locked door. When an authorized individual arrives and opens the door, the intruder goes in as well. The success of this method of piggybacking depends on the quality of the access control mechanism and the alertness of authorized personnel in resisting cooperation with the perpetrator.

Electronic piggybacking can take place in an online computer system in which individuals use terminals and the computer system automatically verifies identification. When a terminal has been activated, the computer authorizes access, usually on the basis of a secret password, token, or other exchange of required identification and authentication information (i.e., a protocol). Compromise of the computer can occur when a covert computer terminal is connected to the same line through the telephone switching equipment and is then used when the legitimate user is not using the terminal. The computer cannot differentiate between the two terminals; it senses only one terminal and one authorized user.

Electronic piggybacking can also be accomplished when the user signs off or a session terminates improperly, leaving the terminal or communications circuit in an active state or leaving the computer in a state in which it assumes the user is still active. Call forwarding of the victim's telephone to the perpetrator's telephone is another means of piggybacking.

Tailgating involves connecting a computer user to a computer in the same session as and under the same identifier as another computer user, whose session has been interrupted. This situation happens when a dial-up or direct-connect session is abruptly terminated and a communications controller (i.e., a concentrator or packet assembler/disassembler) incorrectly allows a second user to be patched directly into the first user's still-open files.

This problem is exacerbated if the controller incorrectly handles a modem's data-terminal-ready signal. Many network managers set up the controller to send data-terminal-ready signals continually so that the modem quickly establishes a new session after finishing its disconnect sequence from the previous session. The controller may miss the modem's drop-carrier signal after a session is dropped, allowing a new session to tailgate onto the old session.

In one vexing situation, computer users connected their office terminal hardwired cables directly to their personal modems. This allowed them to connect any outside telephone directly to their employer's computers

Potential Perpetrators	Methods of Detection	Evidence
• Employees and former employees • Vendor's employees • Contracted persons • Outsiders	• Access observations • Interviewing witnesses • Examination of journals and logs • Out-of-sequence messages • Specialized computer programs that analyze characteristics of on line computer user accesses	• Logs, journals, and equipment usage meters • Photographs and voice and video recordings • Other physical evidence

Exhibit 3. Detection of Piggybacking and Tailgating

through central data switches, thus avoiding all dial-up protection controls (e.g., automatic callback devices). Such methods are very dangerous and have few means of acceptable control.

Prevention of Piggybacking and Tailgating

Turnstiles, double doors, or a stationed guard are the usual methods of preventing physical piggybacking. The turnstile allows passage of only one individual with a metal key, an electronic or magnetic card key, or the combination to a locking mechanism. The double door is a double-doored closet through which only one person can move with one key activation.

Electronic door access control systems frequently are run by a microcomputer that produces a log identifying each individual gaining access and the time of access. Alternatively, human guards may record this information in logs. Unauthorized access can be detected by studying these logs and interviewing people who may have witnessed the unauthorized access. Exhibit 3 summarizes the methods of detecting computer abuse committed by piggybacking and tailgating methods.

FALSE DATA ENTRY

False data entry is usually the simplest, safest, and most common method of computer abuse. It involves changing data before or during its input to computers. Anybody associated with or having access to the processes of creating, recording, transporting, encoding, examining, checking, converting, and transforming data that ultimately enters a computer can change this data. Examples of false data entry include forging, misrepresenting, or counterfeiting documents; exchanging computer tapes or disks; keyboard entry falsifications; failure to enter data; and neutralizing or avoiding controls.

Potential Perpetrators	Methods of Detection	Evidence
• Transaction participants • Data preparers • Source data suppliers • Nonparticipants with access	• Data comparison • Document validation • Manual controls • Audit log analysis • Computer validation • Report analysis • Computer output comparison • Integrity tests (e.g., for value limits, logic consistencies, hash totals, crossfoot and column totals, and forged entry)	• Data documents: —Source —Transactions • Computer-readable output • Computer data media: —Tapes —Disks —Storage modules • Manual logs, audit logs, journals, and exception reports • Incorrect computer output • Control violation alarms

Exhibit 4. Detection of False Data Entry

Preventing False Data Entry

Data entry typically must be protected using manual controls. Manual controls include separation of duties or responsibilities, which force collusion among employees to perpetrate fraudulent acts.

In addition, batch control totals can be manually calculated and compared with matching computer-produced batch control totals. Another common control is the use of check digits or characters embedded in the data on the basis of various characteristics of each field of data (e.g., odd or even number indicators or hash totals). Sequence numbers and time of arrival can be associated with data and checked to ensure that data has not been lost or reordered. Large volumes of data can be checked with utility or special-purpose programs.

Evidence of false data entry is data that does not correctly represent data found at sources, does not match redundant or duplicate data, and does not conform to earlier forms of data if manual processes are reversed. Further evidence is control totals or check-digits that do not check or meet validation and verification test requirements in the computer.

Exhibit 4 summarizes the likely perpetrators of false data entry, methods of detection, and sources of evidence.

SUPERZAPPING

Computers sometimes stop, malfunction, or enter a state that cannot be overcome by normal recovery or restart procedures. In addition, computers occasionally perform unexpectedly and need attention that normal

access methods do not allow. In such cases, a universal access program is needed.

Superzapping derives its name from Superzap, a utility program used as a systems tool in most IBM mainframe centers. This program is capable of bypassing all controls to modify or disclose any program or computer-based data. Many programs similar to Superzap are available for micro-computers as well.

Such powerful utility programs as Superzap can be dangerous in the wrong hands. They are meant to be used only by systems programmers and computer operators who maintain the operating system and should be kept secure from unauthorized use. However, they are often placed in program libraries, where they can be used by any programmer or operator who knows how to use them.

Detection of Superzapping

Unauthorized use of Superzap programs can result in changes to data files that are usually updated only by production programs. Typically, few if any controls can detect changes in the data files from previous runs. Applications programmers do not anticipate this type of fraud; their realm of concern is limited to the application program and its interaction with data files. Therefore, the fraud is detected only when the recipients of regular computer output reports from the production program notify management that a discrepancy has occurred.

Furthermore, computer managers often conclude that the evidence indicates data entry errors, because it would not be a characteristic computer or program error. Considerable time can be wasted in searching the wrong areas. When management concludes that unauthorized file changes have occurred independent of the application program associated with the file, a search of all computer use logs might reveal the use of a Superzap program, but this is unlikely if the perpetrator anticipates the possibility. Occasionally, there may be a record of a request to have the file placed online in the computer system if it is not typically in that mode. Otherwise, the changes would have to occur when the production program using the file is being run or just before or after it is run.

Superzapping may be detected by comparing the current file with parent and grandparent copies of the file. Exhibit 5 summarizes the potential perpetrators, methods of detection, and sources of evidence in superzapping abuse.

SCAVENGING

Scavenging is a method of obtaining or reusing information that may be left after processing. Simple physical scavenging could involve searching

Potential Perpetrators	Methods of Detection	Evidence
• Programmers with access to Superzap programs • Computer operation staff with applications knowledge	• Comparison of files with historical copies • Discrepancies in output reports, as noted by recipients • Examination of computer usage logs	• Output report discrepancies • Undocumented transactions • Computer usage or file request logs

Exhibit 5. Detection of Superzapping

trash barrels for copies of discarded computer listings or carbon paper from multiple-part forms. More technical and sophisticated methods of scavenging include searching for residual data left in a computer, computer tapes, and disks after job execution.

Computer systems are designed and operators are trained to preserve data, not destroy it. If computer operators are requested to destroy the contents of disks or tapes, they most likely make backup copies first. This situation offers opportunities for both criminals and investigators.

In addition, a computer operating system may not properly erase buffer storage areas or cache memories used for the temporary storage of input or output data. Many operating systems do not erase magnetic disk or magnetic tape storage media because of the excessive computer time required to do this. (The data on optical disks cannot be electronically erased, though additional bits could be burned into a disk to change data or effectively erase them by, for example, changing all zeros to ones.).

In a poorly designed operating system, if storage were reserved and used by a previous job and then assigned to the next job, the next job might gain access to the same storage area, write only a small amount of data into that storage area, and then read the entire storage area back out, thus capturing data that was stored by the previous job.

Detection of Scavenging

Exhibit 6 lists the potential perpetrators of, methods of detection for, and evidence in scavenging crimes.

TROJAN HORSES

The Trojan horse method of abuse involves the covert placement or alteration of computer instructions or data in a program so that the computer will perform unauthorized functions. Typically, the computer still allows the program to perform most or all of its intended purposes.

Potential Perpetrators	Methods of Detection	Evidence
• Users of the computer system • Persons with access to computer or backup facilities and adjacent areas	• Tracing of discovered proprietary information back to its source • Testing of an operating system to reveal residual data after job execution	• Computer output media • Type font characteristics • Proprietary information produced in suspicious ways and appearing in computer output media

Exhibit 6. Detection of Scavenging

Trojan horse programs are the primary method used to insert instructions for other abusive acts (e.g., logic bombs, salami attacks, and viruses). This is the most commonly used method in computer program-based frauds and sabotage.

Instructions may be placed in production computer programs so that they will be executed in the protected or restricted domain of the program and have access to all of the data files that are assigned for the program's exclusive use. Programs are usually constructed loosely enough to allow space for inserting the instructions, sometimes without even extending the length or changing the checksum of the infected program.

Detecting and Preventing Trojan Horse Attacks

A typical business application program can consist of more than 100,000 computer instructions and data items. The Trojan horse can be concealed among as many as 5 or 6 million instructions in the operating system and commonly used utility programs. It waits there for execution of the target application program, inserts extra instructions in it for a few milliseconds of execution time, and removes them with no remaining evidence.

Even if the Trojan horse is discovered, there is almost no indication of who may have done it. The search can be narrowed to those programmers who have the necessary skills, knowledge, and access among employees, former employees, contract programmers, consultants, or employees of the computer or software suppliers.

A suspected Trojan horse might be discovered by comparing a copy of the operational program under suspicion with a master or other copy known to be free of unauthorized changes. Although backup copies of production programs are routinely kept in safe storage, clever perpetrators may make duplicate changes in them. In addition, programs are frequently changed for authorized purposes without the backup copies being updated, thereby making comparison difficult.

A program suspected of being a Trojan horse can sometimes be converted from object form into assembly or higher-level form for easier examination or

Potential Perpetrators	Methods of Detection	Evidence
• Programmers with detailed knowledge of a suspected part of a program and its purpose as well as access to it • Employee technologists • Contracted programmers • Vendor programmers • Computer operators	• Program code comparison • Testing of suspected programs • Tracing of unexpected events or possible gain from the act to suspected programs and perpetrators • Examination of computer audit logs for suspicious programs or pertinent entries	• Unexpected results of program execution • Foreign code found in a suspected program • Audit logs • Uncontaminated copies of suspected programs

Exhibit 7. Detection of Trojan Horses and Viruses

comparison by experts. Utility programs are usually available to compare large programs; however, their integrity and the computer system on which they are executed must be verified by trusted experts.

A Trojan horse might be detected by testing the suspect program to expose the purpose of the Trojan horse. However, the probability of success is low unless exact conditions for discovery are known. (The computer used for testing must be prepared in such a way that no harm will be done if the Trojan horse is executed.) Furthermore, this testing may prove the existence of the Trojan horse but usually does not identify its location. A Trojan horse may reside in the source language version or only in the object form and may be inserted in the object form each time it is assembled or compiled — for example, as the result of another Trojan horse in the assembler or compiler. Use of foreign computer programs obtained from untrusted sources (e.g., shareware bulletin board systems) should be restricted, and the programs should be carefully tested before production use.

The methods for detecting Trojan horse frauds are summarized in Exhibit 7. The Exhibit also lists the occupations of potential perpetrators and the sources of evidence of Trojan horse abuse.

COMPUTER VIRUSES

A computer virus is a set of computer instructions that propagates copies of versions of itself into computer programs or data when it is executed within unauthorized programs. The virus may be introduced through a program designed for that purpose (called a pest) or through a Trojan horse. The hidden virus propagates itself into other programs when they are executed, creating new Trojan horses, and may also execute harmful processes under the authority of each unsuspecting computer user whose programs or system have become infected. A worm attack is a variation in

which an entire program replicates itself throughout a computer or computer network.

Although the virus attack method has been recognized for at least 15 years, the first criminal cases were prosecuted only in November 1987. Of the hundreds of cases that occur, most are in academic and research environments. However, disgruntled employees or ex-employees of computer program manufacturers have contaminated products during delivery to customers.

Preventing, Detecting, and Recovering from Virus Attacks

Prevention of computer viruses depends on protection from Trojan horses or unauthorized programs, and recovery after introduction of a virus entails purging all modified or infected programs and hardware from the system. The timely detection of Trojan horse virus attack depends on the alertness and skills of the victim, the visibility of the symptoms, the motivation of the perpetrator, and the sophistication of the perpetrator's techniques. A sufficiently skilled perpetrator with enough time and resources could anticipate most know methods of protection from Trojan horse attacks and subvert them.

Prevention methods consist primarily of investigating the sources of untrusted software and testing foreign software in computers that have been conditioned to minimize possible losses. Prevention and subsequent recovery after an attack are similar to those for any Trojan horse. The system containing the suspected Trojan horse should be shut down and not used until experts have determined the sophistication of the abuse and the extent of damage. The investigator must determine whether hardware and software errors or intentionally produced Trojan horse attacks have occurred.

Investigators should first interview the victims to identify the nature of the suspected attack. They should also use the special tools available (not resident system utilities) to examine the contents and state of the system after a suspected event. The original provider of the software packages suspected of being contaminated should be consulted to determine whether others have had similar experiences. Without a negotiated liability agreement, however, the vendor may decide to withhold important and possibly damaging information.

The following are examples of possible indications of a virus infection:

- The file size may increase when a virus attaches itself to the program or data in the file.
- An unexpected change in the time of last update of a program or file may indicate a recent unauthorized modification.

- If several executable programs have the same date or time in the last update field, they have all been updated together, possibly by a virus.
- A sudden unexpected decrease in free disk space may indicate sabotage by a virus attack.
- Unexpected disk accesses, especially in the execution of programs that do not use overlays or large data files, may indicate virus activity.

All current conditions at the time of discovery should be documented, using documentation facilities separate from the system in use. Next, all physically connected and inserted devices and media that are locally used should be removed if possible. If the electronic domain includes remote facilities under the control of others, an independent means of communication should be used to report the event to the remote facilities manager. Computer operations should be discontinued; accessing system functions could destroy evidence of the event and cause further damage. For example, accessing the contents or directory of a disk could trigger the modification or destruction of its contents.

To protect themselves against viruses or indicate their presence, users can:

- Compare programs or data files that contain checksums or hash totals with backup versions to determine possible integrity loss.
- Write-protect diskettes whenever possible, especially when testing an untrusted computer program. Unexpected write-attempt errors may indicate serious problems.
- Boot diskette-based systems using clearly labeled boot diskettes.
- Avoid booting a hard disk drive system from a diskette.
- Never put untrusted programs in hard disk root directories. Most viruses can affect only the directory from which they are executed; therefore, untrusted computer programs should be stored in isolated directories containing a minimum number of other sensitive programs or data files.
- When transporting files from one computer to another, use diskettes that have no executable files that might be infected.
- When sharing computer programs, share source code rather than object code, because source code can more easily be scanned for unusual contents.

The best protection against viruses, however, is to frequently back up all important data and programs. Multiple backups should be maintained over a period of time, possibly up to a year, to be able to recover from uninfected backups. Trojan horse programs or data may be buried deeply in a computer system — for example, in disk sectors that have been declared by the operating system as unusable. In addition, viruses may contain counters for logic bombs with high values, meaning that the virus may be spread many times before its earlier copies are triggered to cause visible damage.

The perpetrators, detection, and evidence are the same as for Trojan horse attacks (see Exhibit 7).

SALAMI TECHNIQUES

A salami technique is an automated form of abuse involving Trojan horses or secret execution of an unauthorized program that causes the unnoticed or immaterial debiting of small amounts of assets from a large number of sources or accounts. The name of this technique comes from the fact that small slices of assets are taken without noticeably reducing the whole. Other methods must be used to remove the acquired assets from the system.

For example, in a banking system, the demand deposit accounting system of programs for checking accounts could be changed (using the Trojan horse method) to randomly reduce each of a few hundred accounts by 10 cents or 15 cents by transferring the money to a favored account, where it can be withdrawn through authorized methods. No controls are violated because the money is not removed from the system of accounts. Instead, small fractions of the funds are merely rearranged, which the affected customers rarely notice. Many variations are possible. The assets may be an inventory of products or services as well as money. Few cases have been reported.

Detecting Salami Acts

Several technical methods for detection are available. Specialized detection routines can be built into the suspect program, or snapshot storage dump listings could be obtained at crucial times in suspected program production runs. If identifiable amounts are being taken, these can be traced; however, a clever perpetrator can randomly vary the amounts or accounts debited and credited. Using an iterative binary search of balancing halves of all accounts is another costly way to isolate an offending account.

The actions and lifestyles of the few people with the skills, knowledge, and access to perform salami acts can be closely watched for deviations from the norm. For example, the perpetrators or their accomplices usually withdraw the money from the accounts in which it accumulates in legitimate ways; records will show an imbalance between the deposit and withdrawal transaction. However, all accounts and transactions would have to be balanced over a significant period of time to detect discrepancies. This is a monumental and expensive task.

Many financial institutions require employees to use only their financial services and make it attractive for them to do so. Employees' accounts are more completely and carefully audited than others. Such requirements usually force the salami perpetrators to open accounts under assumed

Potential Perpetrators	Methods of Detection	Evidence
• Financial system programmers	• Detailed data analysis using a binary search	• Many small financial losses
• Employee technolgists	• Program comparison	• Unsupported account balance buildups
• Former employees	• Transaction audits	• Trojan horse code
• Contracted programmers	• Observation of financial activities of possible suspects	• Changed or unusual personal financial practices of possible suspects
• Vendor's programmers		

Exhibit 8. Detection of Salami Acts

names or arrange for accomplices to commit the fraud. Therefore, detection of suspected salami frauds might be more successful if investigators concentrate on the actions of possible suspects rather than on technical methods of discovery.

Exhibit 8 lists the methods of detecting the use of salami techniques as well as the potential perpetrators and sources of evidence of the use of the technique.

TRAPDOORS

Computer operating systems are designed to prevent unintended access to them and unauthorized insertion of modification of code. Programmers sometimes insert code that allows them to compromise these requirements during the debugging phases of program development and later during system maintenance and improvement. These facilities are referred to as trapdoors, which can be used for Trojan horse and direct attacks (e.g., false data entry).

Trapdoors are usually eliminated in the final editing, but sometimes they are overlooked or intentionally left in to facilitate future access and modification. In addition, some unscrupulous programmers introduce trapdoors to allow them to later compromise computer programs. Furthermore, designers or maintainers of large complex programs may also introduce trapdoors inadvertently through weaknesses in design logic.

Trapdoors may also be introduced in the electronic circuitry of computers. For example, not all of the combinations of codes may be assigned to instructions found in the computer and documented in the programming manuals. When these unspecified commands are used, the circuitry may cause the execution of unanticipated combinations of functions that allow the computer system to be compromised.

Typical known trapdoor flaws in computer programs include:

- Implicit sharing of privileged data.
- Asynchronous change between time of check and time of use.
- Inadequate identification, verification, authentication, and authorization of tasks.
- Embedded operating system parameters in application memory space.
- Failure to remove debugging aids before production use begins.

During the use and maintenance of computer programs and computer circuitry, ingenious programmers invariably discover some of these weaknesses and take advantage of them for useful and innocuous purposes. However, the trapdoors may be used for unauthorized, malicious purposes as well.

Functions that can be performed by computer programs and computers that are not in the specifications are often referred to as negative specifications. Designers and implementers struggle to make programs and computers function according to specifications and to prove that they do. They cannot practicably prove that a computer system does not perform functions it is not supposed to perform.

Research is continuing on a high priority basis to develop methods of proving the correctness of computer programs and computers according to complete and consistent specifications. However, commercially available computers and computer programs probably will not be proved correct for many years. Trapdoors continue to exist; therefore, computer systems are fundamentally insecure because their actions are not totally predictable.

Detecting Trapdoors

No direct technical method can be used to discover trapdoors. However, tests of varying degrees of complexity can be performed to discover hidden functions used for malicious purposes. The testing requires the expertise of systems programmers and knowledgeable applications programmers. Investigators should always seek out the most highly qualified experts for the particular computer system or computer application under suspicion.

The investigator should always assume that the computer system and computer programs are never sufficiently secure from intentional, technical compromise. However, these intentional acts usually require the expertise of only the technologists who have the skills, knowledge, and access to perpetrate them. Exhibit 9 lists the potential perpetrators, methods of detection, and sources of evidence of the abuse trapdoors.

Potential Perpetrators	Methods of Detection	Evidence
• Expert application programmers	• Exhaustive testing • Comparison of specification to performance • Specific testing based on evidence	• Computer performance or output reports indicating that a computer system performs outside of its specifications

Exhibit 9. Detection of Trapdoors

Potential Perpetrators	Methods of Detection	Evidence
• Programmers with detailed knowledge of a suspected part of a program and its purpose as well as access to it • Employees • Contracted programmers • Vendor's programmers • Computer users	• Program code comparisons • Testing of suspected programs • Tracing of possible gains from the act	• Unexpected results of program execution • Foreign code found in a suspected program

Exhibit 10. Detection of Logic Bombs

LOGIC BOMBS

A logic bomb is a set of instructions in a computer program periodically executed in a computer system that determines conditions or states of the computer, facilitating the perpetration of an unauthorized, malicious act. In one case, for example, a payroll system programmer put a logic bomb in the personnel system so that if his name were ever removed from the personnel file, indicating termination of employment, secret code would cause the entire personnel file to be erased.

A logic bomb can be programmed to trigger an act based on any specified condition or data that may occur or be introduced. Logic bombs are usually placed in the computer system using the Trojan horse method. Methods of discovering logic bombs are the same as for Trojan horses. Exhibit 10 summarizes the potential perpetrators, methods of detection, and kinds of evidence of logic bombs.

ASYNCHRONOUS ATTACKS

Asynchronous attacks take advantage of the asynchronous functioning of a computer operating system. Most computer operating systems function asynchronously on the basis of the services that must be performed for the various computer programs executed in the computer system. For

example, several jobs may simultaneously call for output reports to be produced. The operating system stores these requests and, as resources become available, performs them in the order in which resources are available to fit the request or according to an overriding priority scheme. Therefore, rather than executing requests in the order they are received, the system performs then asynchronously on the basis of the available resources.

Highly sophisticated methods can confuse the operating system to allow it to violate the isolation of one job from another. For example, in a large application program that runs for a long time, checkpoint/restarts are customary. These automatically allow the computer operator to set a switch manually to stop the program at a specified intermediate point and later restart it in an orderly manner without losing data.

To avoid the loss, the operating system must save the copy of the computer programs and data in their current state at the checkpoint. The operating system must also save several system parameters that describe the mode and security level of the program at the time of the stop. Programmers or computer operators might be able to gain access to the checkpoint restart copy of the program, data, and system parameters. They could change the system parameters such that on restart, the program would function at a higher-priority security level or privileged level in the computer and thereby give the program unauthorized access to data, other programs, or the operating system. Checkpoint/restart actions are usually well documented in the computer operations or audit log.

Even more complex methods of attack could be used besides the one described in this simple example, but the technology is too complex to present here. The investigator should be aware of the possibilities of asynchronous attacks and seek adequate technical assistance if suspicious circumstances result from the activities of highly sophisticated and trained technologists. Evidence of such attacks would be discernible only from unexplained deviations from application and system specifications in computer output, or characteristics of system performance. Exhibit 11 lists the potential perpetrators, methods of detecting, and evidence of asynchronous attacks.

DATA LEAKAGE

A wide range of computer crime involves the removal of data or copies of data from a computer system or computer facility. This part of a crime may offer the most dangerous exposure to perpetrators. Their technical act may be well hidden in the computer; however, to convert it to economic gain, they must get the data from the computer system. Output is subject to examination by computer operators and other data processing personnel, who might detect the perpetrators' activity.

Potential Perpetrators	Methods of Detection	Evidence
• Sophisticated advanced system programmers • Sophisticated and advanced computer operators	• System testing of suspected attack methods • Repeat execution of a job under normal and secured circumstances	• Output that deviates from expected output or logs containing records of computer operation

Exhibit 11. Detection of Asynchronous Attacks

Several techniques can be used to secretly leak data from a computer system. The perpetrator may be able to hide the sensitive data in otherwise innocuous-looking output reports — for example, by adding to blocks of data or interspersing the data with otherwise routine data. A more sophisticated method might be to encode data to look like something else. For example, a computer listing may be formatted so that the secret data is in the form of different lengths of printer lines, number of characters per line, or locations of punctuation; is embedded in the least significant digits of engineering data; and uses code words that can be interspersed and converted into meaningful data.

Sophisticated methods of data leakage might be necessary only in high-security, high-risk environments. Otherwise, much simpler manual methods might be used. It has been reported that hidden in the central processors of many computers used in the Vietnam War were miniature radio transmitters capable of broadcasting the contents of the computers to a remote receiver. These were discovered when the computers were returned to the United States.

Detecting Data Leakage

Data leakage would probably best be investigated by interrogating IS personnel who might have observed the movement of sensitive data. In addition, computer operating system usage logs could be examined to determine whether and when data files have been accessed. Because data leakage can occur through the use of Trojan horses, logic bombs, and scavenging, the use of these methods should be investigated when data leakage is suspected.

Evidence will most likely be in the same form as evidence of the scavenging activities described in a preceding section. Exhibit 12 summarizes the detection of crimes resulting from data leakage.

SOFTWARE PIRACY

Piracy is the copying and use of computer programs in violation of copyright and trade secret laws. Commercially purchased computer programs are protected by what is known as a shrink-wrap contract agreement,

Potential Perpetrators	Methods of Detection	Evidence
• Computer programmers • Employees • Former employees • Contracted workers • Vendor's employees	• Discovery of stolen information • Tracing computer storage media back to the computer facility	• Computer storage media • Computer output forms • Type font characteristics • Trojan horse or scavenging evidence

Exhibit 12. Detection of Data Leakage

which states that the program is protected by copyright and its use is restricted.

Since the early 1980s, violations of these agreements have been widespread, primarily because of the high price of commercial programs and the simplicity of copying the programs. The software industry reacted by developing several technical methods of preventing the copying of disks; however, these have not always been successful because of hackers' skills at overcoming this protection and because they are seen as inconvenient to customers.

The software industry has now stabilized and converged on a strategy of imposing no technical constraints to copying, implementing an extensive awareness program to convince honest customers not to engage in piracy, pricing their products more reasonably, and providing additional benefits to purchasers of their products that would not be obtainable to computer program pirates. In addition, computer program manufacturers occasionally find gross violations of their contract agreements and seek highly publicized remedies.

Malicious hackers commonly engage in piracy, sometimes even distributing pirated copies on a massive scale through electronic bulletin boards. Although criminal charges can often be levied against malicious hackers and computer intruders, indictments are most often sought against educational and business institutions, in which gross violations of federal copyright laws and state trade secret laws are endemic.

Detecting Piracy

Investigators can most easily obtain evidence of piracy by confiscating suspects' disks, the contents of their computer hard disks, paper printouts from the execution of the pirated programs, and pictures of screens produced by the pirated programs. Recent court decisions indicate that piracy can also occur when programs are written that closely duplicate the look and feel of protected computer programs, which includes the use of similar command structures and screen displays. Exhibit 13 summarizes the potential perpetrators, detection methods, and evidence of computer program piracy.

Potential Perpetrators	Methods of Detection	Evidence
• Any purchasers and users of commercially available computer programs • Hackers	• Observation of computer users • Search of computer users' facilities and computers • Testimony of legitimate computer program purchasers • Receivers of copied computer programs	• Pictures of computer screens while pirated software is being executed • Copies of computer media on which pirated programs are found • Memory contents of computers containing pirated software • Printouts produced by execution of pirated computer programs

Exhibit 13. Detection of Software Piracy

COMPUTER LARCENY

The theft, burglary, and sale of stolen microcomputers and components are increasing dramatically — a severe problem because the value of the contents of stolen computers often exceeds the value of the hardware taken. The increase in computer larceny is becoming epidemic, in fact, as the market for used computers in which stolen merchandise may be fenced also expands.

It has been suggested that an additional method of protection be used along with standard antitheft devices for securing office equipment. If the user is to be out of the office, microcomputers can be made to run antitheft programs that send frequent signals through modems and telephones to a monitoring station. If the signals stop, an alarm at the monitoring station is set off.

Investigation and prosecution of computer larceny fits well within accepted criminal justice practices, except for proving the size of the loss when a microcomputer worthy only a few hundred dollars is stolen. Evidence of far larger losses (e.g., programs and data) may be needed.

Minicomputers and mainframes have been stolen as well, typically while equipment is being shipped to customers. Existing criminal justice methods can deal with such thefts.

USE OF COMPUTERS FOR CRIMINAL ENTERPRISE

A computer can be used as a tool in a crime for planning, data communications, or control. An existing process can be simulated on a computer, a planned method for carrying out a crime can be modeled, or a crime can be monitored by a computer (i.e., by the abuser) to help guarantee its success.

Potential Perpetrators	Methods of Detection	Evidence
• Computer application programmers • Simulation and modeling experts • Managers in positions to engage in large, complex embezzlement • Criminal organizations	• Investigation of possible computer use by suspects • Identification of equipment	• Computer programs • Computer and communications equipment and their contents • Computer program documentation • Computer input • Computer-produced reports • Computer and data communications usage logs and journals

Exhibit 14. Detection of Simulation and Modeling

In one phase of a 1973 insurance fraud in Los Angeles, a computer was used to model the company and determine the effects of the sale of large numbers of insurance policies. The modeling resulted in the creation of 64,000 fake insurance policies in computer-readable form that were then introduced into the real system and subsequently resold as valid policies to reinsuring companies.

The use of a computer for simulation, modeling, and data communications usually requires extensive amounts of computer time and computer program development. Investigation of possible fraudulent use should include a search for significant amounts of computer services used by the suspects. Their recent business activities, as well as the customer lists of locally available commercial time-sharing and service bureau companies, can be investigated. If inappropriate use of the victim's computer is suspected, logs may show unexplained computer use.

Exhibit 14 lists the potential perpetrators, methods of detection, and kinds of evidence in simulation and modeling techniques.

SUMMARY

Computer crimes will change rapidly along with the technology. As computing becomes more widespread, maximum losses per case are expected to grow. Ultimately, all business crimes will be computer crimes.

Improved computer controls will make business crime more difficult, dangerous, and complex, however. Computers and workstations impose absolute discipline on information workers, forcing them to perform within set bounds and limiting potential criminal activities. Managers receive

improved and more timely information from computers about their businesses and can more readily discern suspicious anomalies indicative of possible wrongdoing.

Although improved response rates from victims, improvements in security, modification of computer use, reactions from the criminal justice community, new laws, and saturation of the news media warning of the problems will cause a reduction of traditional types of crime, newer forms of computer crime will proliferate. Viruses and malicious hacking will eventually be superseded by other forms of computer abuse, including computer larceny, desktop forgery, voice mail and E-mail terrorism and extortion, fax graffiti, phantom computers secretly connected to networks, and repudiation of EDI transactions.

149

The International Dimensions of Cyber-Crime*

Ed Gabrys, CISSP

It is Monday morning and you begin your prework ritual by going to the World Wide Web and checking the morning electronic newspapers. In the past you might have read the paper edition of *The New York Times* or *The Wall Street Journal;* but with free news services and robust search features available on the Internet, you have decided to spare the expense and now the Internet is your primary news source. Your browser automatically opens to the electronic edition of your favorite news site, where you see the latest headline, "Electronic Terrorist Group Responsible for Hundreds of Fatalities." Now wishing that you had the paper edition, you wonder if this news story is real or simply a teenage hacker's prank. This would not be the first time that a major news service had its Web site hacked. You read further and the story unfolds. A terrorist group, as promised, has successfully struck out at the United States. This time, the group did not use conventional terrorist weapons such as firearms and explosives, but instead has attacked state infrastructure using computers. Electronically breaking into electric power plants, automated pipelines, and air-traffic-control systems, in one evening they have successfully caused havoc and devastation across the United States, including mid-air collisions over major U.S. city airports. To top it off, the U.S. government is unable to locate the culprits. The only thing that authorities know for sure is that the perpetrators are not physically located in the United States.

Is this science fiction or a possible future outcome? As an information security specialist, you have probably heard variations on this theme many times; but now, in the light of both homegrown and foreign terrorism striking the United States, the probability needs to be given serious thought. Considering the growing trends in computer crime, world dependence on computers and communication networks, and the weaknesses in the world's existing laws, it may soon be history. Kenneth A. Minihan, Director of the National Security Agency, has called the Information Superhighway "the economic lifeblood of our nation."[1] When you consider that order, economic prosperity is as important to state security as military power in the New World, an attack on a country's infrastructure may be as devastating as a military attack. This could be the next Pearl Harbor — an *electronic* Pearl Harbor!

To successfully combat the cyber-crime threat, a global solution must be addressed. To date, the only far-reaching and coordinated global response to the cyber-crime problem has been the Convention on Cyber-Crime developed by the Council of Europe (CoE). Unfortunately, the treaty has the potential to achieve its goals at the loss of basic human rights and innovation, and by extending state powers. Those who drafted the treaty have violated an important principle of regime theory — disallowance of the participation of all relevant actors in its decision making by drafting a convention that only represents the voice of the actors in power.

To clarify the arguments outlined above, this chapter first defines the scale and extent of the growing global cyber-crime threat. The second section illustrates how organizations are currently responding and highlights the Council of Europe's solution. In the third section, regime theory is defined and applied to the global cyber-

*A look at the Council of Europe's Cyber-Crime Convention and the need for an international regime to fight cyber-crime

crime problem; then an argument of how the CoE's convention fails to embrace an important element of regime-theory principles is made. Finally, in the last section, an adjusted Council of Europe convention is offered as an alternative and will be compared to a notable and successful international regime.

Part I: Global Cyber-Crime

The Cyber-Crime Threat

> Look at how many clueless admins are out there. Look at what kind of proprietary data they are tasked to guard. Think of how easy it is to get past their pathetic defenses…. 'The best is the enemy of the good.'
>
> **— Voltaire[2]**
>
> **Posted on *The New York Times* Web site**
>
> **by the computer hacking group, Hacking 4 Girliez**

A New Age and New Risks

The human race has passed through a number of cultural and economic stages. Most of our progress can be attributed to the ideas and the tools we have created to develop them. Wielding sticks and stones, we began our meager beginnings on a par with the rest of the animal kingdom, as hunters and gatherers. We then graduated to agrarian life using our picks and shovels, through an industrial society with our steam engines and assembly lines, and have arrived in today's Digital Age. Computers and communication networks now dominate our lives. Some may argue that a vast number of people in the world have been overlooked by the digital revolution and have never made a phone call, let alone e-mailed a friend over the Internet. The advent of computers has had far-reaching effects; and although some people may not have had the opportunity to navigate the digital highway, they probably have been touched in other ways. Food production, manufacturing, education, health care, and the spread of ideas have all been beneficiaries of the digital revolution. Even the process of globalization owes its far and rapid reach to digital tools.

For all of the benefits that the computer has brought us, like the tools of prior ages, we have paid little attention to the potential harm they bring until after the damage has been done. On one hand, the Industrial Age brought industrialized states greater production and efficiency and an increase in standards of living. On the other, it also produced mechanized warfare, sweatshops, and a depleting ozone layer, to name a few. Advocates of the Digital Age and its now most famous invention, the Internet, flaunt dramatic commercial growth, thriving economies, and the spread of democracy as only a partial list of benefits. The benefits are indeed great, but so are the costs. One such cost that we now face is a new twist on traditional crime — cyber-crime.

An International Threat

> Because of its technological advancements, today's criminals can be more nimble and more elusive than ever before. If you can sit in a kitchen in St. Petersburg, Russia, and steal from a bank in New York, you understand the dimensions of the problem.[3]
>
> **— Former Attorney General Janet Reno**

Cyber-crime is an extension of traditional crime, but it takes place in cyberspace[4] — the nonphysical environment created by computer systems. In this setting, cyber-crime adopts the nonphysical aspects of cyberspace and becomes borderless, timeless, and relatively anonymous. By utilizing globally connected phone systems and the world's largest computer network, the Internet, cyber-criminals are able to reach out from nearly anywhere in the world to nearly any computer system, as long as they have access to a communications link. Most often, that only needs to be a reliable phone connection. With the spread of wireless and satellite technology, location will eventually become totally irrelevant. In essence, the global reach of computer networks has created a borderless domain for cyber-crimes. Add in automation, numerous time zones, and 24/7 access to computer systems, and now time has lost significance. A famous *New Yorker* cartoon shows a dog sitting at

a computer system speaking to his canine companion, saying, "On the Internet, nobody knows you're a dog."[5] In this borderless and timeless environment, only digital data traverses the immense digital highway, making it difficult to know who or what may be operating a remote computer system. As of today there are very few ways to track that data back to a person, especially if the person is skilled enough to conceal his tracks. Moreover, cyber-criminals are further taking advantage of the international aspect of the digital domain by networking with other cyber-criminals and creating criminal gangs. Being a criminal in cyber space takes technical know-how and sophistication. By dividing up the work, cyber-gangs are better able to combat the sophistication and complexities of cyber space. With computers, telecommunications networks, and coordination, the cyber-criminal has achieved an advantage over his adversaries in law enforcement. Cyber-crime, therefore, has an international aspect that creates many difficulties for nations that may wish to halt it or simply mitigate its effects.

Cyber-Crime Defined

Cyber-crime comes in many guises. Most often, people associate cyber-crime with its most advertised forms — Web hacking and malicious software such as computer worms and viruses, or *malware* as it is now more often called. Who can forget some of these more memorable events? Distributed denial-of-service attacks in early 2000 brought down E-commerce sites in the United States and Europe, including Internet notables Yahoo!, Amazon.com, and eBay. The rash of computer worms that are becoming more sophisticated spread around the world in a matter of hours and cost businesses millions — or by some estimates, billions — in damages related to loss and recovery. Also in 2000, a Russian hacker named "Maxus" stole thousands of credit card numbers from the online merchant CD Universe and held them for ransom at $100,000 (U.S.). When his demands were not met, he posted 25,000 of the numbers to a public Web site. These are just a sample of the more recent and widely publicized events. These types of cyber-crimes are often attributed to hackers — or, as the hacker community prefers them to be called, crackers or criminals.

Most often, the hackers associated with many of the nuisance crimes such as virus writing and Web site defacements are what security experts refer to as script kiddies. They are typically males between the ages of 15 and 25, of whom Jerry Schiller, the head of network security at the Massachusetts Institute of Technology, said, "… are usually socially maladjusted. These are not the geniuses. These are the misfits."[6] Although these so-called misfits are getting much of the public attention, the threat goes deeper. The annual CSI/FBI Computer Crime and Security Survey,[7] as shown in Exhibit 149.1, cited foreign governments and corporations, U.S. competitors, and disgruntled employees as other major players responsible for cyber-attacks.[8]

Because cyber-crime is not bound by physical borders, it stands to reason that cyber-criminals can be found anywhere around the world. They do, however, tend to concentrate in areas where education is focused on mathematics (a skill essential to hacking), computer access is available, and the country is struggling economically, such as Russia, Romania, or Pakistan. Although this does not preclude other countries such as the United

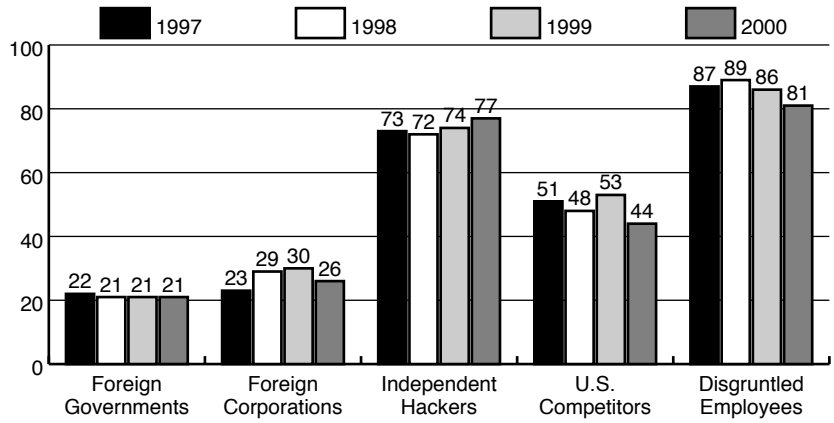


EXHIBIT 149.1 CSI/FBI 2000 Computer Crime and Security Survey. (*Source*: Computer Security Institute)

EXHIBIT 149.2 Ten Foreign Hot Spots for Credit Card Fraud

City	Percent of Fraudulent Foreign Orders
Bucharest, Romania	12.76
Minsk, Belarus	8.09
Lasi, Romania	3.14
Moscow, Russia	2.43
Karachi, Pakistan	1.23
Krasnogorsk, Russia	0.78
Cairo, Egypt	0.74
Vilnius, Lithuania	0.74
Padang, Indonesia	0.59
Sofia, Bulgaria	0.56

Source: *Internet World*, February 1, 1999.

Kingdom or United States from having their share of computer criminals, recent trends suggest that the active criminal hackers tend to center in these specific areas around the globe. This is an indication that, if their talented minds cannot be occupied and compensated as they may be in an economically prosperous country, then they will use their skills for other purposes. Sergie Pokrovsky, an editor of the Russian hacker magazine *Khaker*, said hackers in his circle "… have skills that could bring them rich salaries in the West, but they expect to earn only about $300 a month working for Russian companies."[9] An online poll on a hacker-oriented Web site asked respondents to name the world's best hackers and awarded hackers in Russia top honors, with 82 percent of the vote. Compare that to the paltry five percent given to American hackers.[10] Looking at online credit card fraud, a 1999 survey of Yahoo! stores (see Exhibit 149.2) reported that nearly a third of foreign orders placed with stolen credit cards could be traced to ten international cities, which is an indicator of the geographic centers of major international hacker concentrations.[11]

Cyber-crime is quite often simply an extension of traditional crimes; and, similarly, there are opportunities for everyone — foreign spies, disgruntled employees, fraud perpetrators, political activists, conventional criminals, as well as juveniles with little computer knowledge. It is easy to see how crimes such as money laundering, credit card theft, vandalism, intellectual property theft, embezzlement, child pornography, and terrorism can exist both in and outside of the cyber-world. Just think about the opportunities that are available to the traditional criminal when you consider that cyber-crime promises the potential for a greater profit and a remote chance of capture. According to the FBI crime files, the average bank robbery yields $4000; the average computer heist can turn around $400,000.[12] Furthermore, the FBI states that there is less than a 1:20,000 chance of a cyber-criminal being caught. This is more evident when you take into consideration that employees — who, as you know, have access to systems, procedures, and passwords — commit 60 percent of the thefts.[13] Adding insult to injury, in the event that a cyber-criminal is actually caught, there is still only a 1:22,000 chance that he will be sent to prison.[14]

Here are just a few examples of traditional crimes that have made their way to the cyber-world. In 1995, a Russian hacker, Vladimir Levin, embezzled more than $10 million from Citibank by transferring electronic money out of the bank's accounts.[15] Copyright infringement or information theft has reached mass proportions with wildly popular file-sharing programs such as Limewire, Morpheous, and the notorious Napster. Millions of copies of copyrighted songs are freely traded among these systems' users all over the globe, which the record companies are claiming cost them billions of dollars.[16] In August 2000, three Kazakhs were arrested in London for allegedly breaking into Bloomberg L.P.'s computer system in Manhattan in an attempt to extort money from the company.[17] A 15-year-old boy was arrested for making terrorist threats and possessing an instrument of crime after he sent electronic mail death threats to a U.S. judge. He demanded the release of three Arab men imprisoned in connection with the failed 1993 plot to blow up several New York City landmarks. If they were not released, he threatened that a *jihad* would be proclaimed against the judge and the United States. Beginning in 1985 until his capture in 2001, Robert Philip Hanssen, while working for the Federal Bureau of Investigation, used computer systems to share national secrets with Russian counterparts and commit espionage.[18] In 1996, members of an Internet chat room called "KidsSexPics" executed a horrific offense involving child pornography and international computer crime. Perpetrators, who included citizens of the United States, Finland, Australia, and Canada, were arrested for orchestrating a child molestation that was broadcast over the Internet.[19]

Computers Go to War: Cyber-Terrorism

> The modern thief can steal more with a computer than with a gun. Tomorrow's terrorist may be able to do more damage with a keyboard than with a bomb.[20]
>
> **— National Research Council, 1991**

> We are picking up signs that terrorist organizations are looking at the use of technology.[21]
>
> **— Ronald Dick**
> **Head of the FBI's Anti-Cyber-Crime Unit**

One of the most frightening elements of cyber-crime is a threat that has fortunately been relatively absent in the world — cyber-terrorism. Cyber-terrorism is, as one may expect, the marriage of terrorism and cyber space. Dorothy Denning, a professor at Georgetown University and a recognized expert in cyber-terrorism, has described it as "unlawful attacks and threats of attack against computer's networks, and the information stored therein when done to intimidate or coerce a government or its people in furtherance of political or social objectives."[22] Although there have been a number of cyber-attacks over the past few years of a political or social nature, none have been sufficiently harmful or frightening to be classified by most authorities as cyber-terrorism. Most of what has occurred, such as threatening e-mails, e-mail bombs, denial-of-service attacks, and computer viruses, are more analogous to street protests and physical sit-ins.

The threat, however, is still very real. In a controlled study, the Department of Defense attacked its own machines. Of the 38,000 machines attacked, 24,700 (or 65 percent) were penetrated. Only 988 (or four percent) of the penetrated sites realized they were compromised; and only 267 (or 27 percent) of those reported the attack.[23] Keep in mind that the Department of Defense has mandatory reporting requirements and a staff that recognizes the importance of following orders, which makes those numbers even more ominous.

Although government systems may have deficiencies, a greater vulnerability may lie with critical infrastructures. Finance, utilities, and transportation systems are predominately managed by the private sector and are far more prone to an attack because those organizations are simply unprepared. A survey by the U.K.-based research firm Datamonitor shows that businesses have been massively underspending for computer security. Datamonitor estimates that $15 billion is lost each year through E-security breaches, while global spending on defense is only $8.7 billion. Moreover, even if business were to improve security spending habits and correct the weaknesses in computer systems, it is effectively impossible to eliminate all vulnerabilities. Administrators often ignore good security practices or are unaware of weaknesses when they configure systems. Furthermore, there is always the possibility that an insider with knowledge may be the attacker. In March 2000, Japan's Metropolitan Police Department reported that software used by the police department to track 150 police vehicles, including unmarked cars, was developed by the Aum Shinryko cult — the same group that gassed the Tokyo subway in 1995, killing 12 people and injuring 6000 others. At the time of the discovery, the cult had received classified tracking data on 115 vehicles.[24]

Experts believe that terrorists are looking at the cyber-world as an avenue to facilitate terrorism. The first way in which terrorists are using computers is as part of their infrastructure, as might any other business trying to take advantage of technological advancements. They develop Web sites to spread messages and recruit supporters, and they use the Internet to communicate and coordinate action.[25]

Clark State, executive director of the Emergency Response and Research Institute in Chicago, testified before a Senate judiciary subcommittee that "members of some Islamic extremist organizations have been attempting to develop a 'hacker network' to support their computer activities and may engage in offensive information warfare attacks in the future."[26] This defines their second and more threatening use of computer systems — that of a weapon. Militant and terrorist groups such as the Indian separatist group Harkat-ul-Ansar and the Provisional Irish Republican Army have already used computer systems to acquire classified military information and technology. In all of the related terrorist cases, there have been no casualties or fatalities directly related to the attack. For those who doubt that a computer attack may be fatal, consider the following real incident. A juvenile from Worcester, Massachusetts, took control of a local telephone switch. Given the opportunity, he disabled local phone service. That alone is not life-threatening. That switch, however, controlled the activation of landing lights for a nearby airport runway that were subsequently rendered inoperable.[27] Luckily, it was a small airport. If it had been the Newark or Los Angeles airport, the effects could have been devastating.

It is believed that most terrorist groups are not yet prepared to stage a meaningful cyber-attack but that they can be in the near future. Understanding that these groups are preparing, critical systems are and will be

vulnerable to an attack; and a successful attack in the cyber-world will gain them immediate and widespread media attention — it should be expected that a cyber-terrorist attack is imminent.

The Threat is Growing

> Every one of us either has been or will be attacked in cyber space. A threat against one is truly a threat against all.[28]
>
> **— Mary Ann Davidson**
> **Security Product Manager at Oracle**

It is difficult to determine what the real scope of the cyber-crime threat is. Most successful computer crimes go unreported to law enforcement or undetected by the victims. If a business has systems that are compromised by a cyber-criminal, they are hard-pressed to make that information public. The cost of the break-in may have been a few thousand, tens of thousands, or possibly hundreds of thousands of dollars. If that cost is not substantial enough, the cost associated with a loss of customer trust and negative public opinion can bankrupt a company.

The statistics that are available illustrate that cyber-crime is undeniably on the rise. The number of Web sites that are reported vandalized each year is reaching numbers close to 1000 a month.[29] ICSA.net reported that the rate of virus infections doubled annually from 1997 to 1999, starting at 21 incidents per month per 1000 computers up to 88.[30] In the United Kingdom, there was a 56-percent increase in cyber-crime for 2000, with most cyber-criminals seeking financial gain or hacking for political reasons.[31] In the first six months of 2000, cyber-crime accounted for half of all U.K. fraud. The FBI has approximately 1400 active investigations into cyber-crime, and there are at least 50 new computer viruses generated weekly that require attention from federal law enforcement or the private sector.[32] According to a Gartner Group study, smaller companies stand a 50:50 chance of suffering an Internet attack by 2003; and more than 60 percent of the victimized companies will not know that they have been attacked.[33] In the event that an attack is undetected, a cyber-criminal can utilize the pirated system to gather information, utilize system capacity, launch further attacks internally or externally to the organization, or leave behind a logic bomb. A logic bomb is a computer program that will wait until triggered and then release a destructive payload. This can include destruction of data, capturing and broadcasting sensitive information, or anything else that a mischievous programmer may be able to devise.

Beyond the increase in incidents, the costs of dealing with cyber-crime are rising as well. A joint study by the American Society for Industrial Security (ASIS) and consulting firm PricewaterhouseCoopers found that Fortune 1000 companies incurred losses of more than $45 billion in 1999 from the theft of proprietary information. That number is up from roughly $24 billion a year in the middle 1990s.[34] Furthermore, the average Fortune 1000 company reported 2.45 incidents with an estimated loss per incident in excess of $500,000.[35] If these numbers are truly accurate, that is a cost of over $1 trillion.

International Issues

> We cannot hope to prevail against our criminal adversaries unless we begin to use the same interactive mechanisms in the pursuit of justice as they use in the pursuit of crime and wealth.[36]
>
> **— Former Attorney General Janet Reno**

Cyber-criminals and cyber-terrorists are chipping away at the cyber-world, weakening the confidentiality, integrity, and availability of our communications channels, computer systems, and the information that traverses or resides in them. As illustrated, the costs are high in many ways. Moreover, if a nation cannot protect its critical infrastructure, the solvency of its businesses, or the safety of its citizens from this growing threat, then it is possible that the nations most dependent on the cyber-world are jeopardizing their very sovereignty. So what is preventing the world from eliminating or at least reducing the cyber-crime threat? The primary challenges are legal and technical.

Whether a cyber-criminal is the proverbial teenage boy hacker or a terrorist, the borderless, timeless, and anonymous environment that computers and communications networks provide creates an international problem for law enforcement agencies. With most crimes, the physical presence of a perpetrator is necessary. This makes investigation of a crime and identification, arrest, and prosecution of a criminal much simpler.

Imagine for a moment that a group of cyber-criminals located in a variety of countries including Brazil, Israel, Canada, and Chile decide to launch an attack to break into an E-commerce Web site that is physically located in California but maintained for a company in New York City. In an attempt to foil investigators, the cyber-gang first takes control of a computer system in South Africa, which in turn is used to attack a system in France. From the system in France, the attackers penetrate the system in California and steal a listing of credit card numbers that they subsequently post to a Web site in England. If California law enforcement is notified, how are they able to investigate this crime? What laws apply? What technology can be used to investigate such a crime?

Legal Issues

> Currently, at least 60 percent of INTERPOL membership lacks the appropriate legislation to deal with Internet/computer-related crime.[37]
>
> **— Edgar Adamson**
> **Head of the U.S. Customs Service**

Traditional criminal law is ill-prepared for dealing with cyber-crime in many ways. The elements that we have taken for granted, such as jurisdiction and evidence, take on a new dimension in cyber space. Below are some of the more important legal issues concerning cyber-crime. This is not intended to be a comprehensive list but rather a highlight.

Criminalizing and Coordinating Computer-Related Offenses

Probably the most important legal hurdle in fighting cyber-crime is the criminalizing and coordinating of computer-related offenses among all countries. Because computer crime is inherently a borderless crime, fighting cyber-criminals cannot be effective until all nations have established comprehensive cyber-crime laws. A report by Chief Judge Stein Schjolberg of Norway highlights a number of countries that still have "no special penal legislation."[38] According to a study that examined the laws of 52 countries and was released in December 2000, Australia, Canada, Estonia, India, Japan, Mauritius, Peru, Philippines, Turkey, and the United States are the top countries that have "fully or substantially updated their laws to address some of the major forms of cyber-crimes."[39] There are still many countries that have not yet adequately addressed the cyber-crime issue, and others are still just considering the development of cyber-security laws.[40]

An excellent example of this issue involves the developer of the "ILOVEYOU" or Love Bug computer worm that was launched from the Philippines in May 2000 and subsequently caused damages to Internet users and companies worldwide calculated in the billions of dollars. A suspect was quickly apprehended, but the case never made it to court because the Philippines did not have adequate laws to cover computer crimes. Because the Philippines did not have the laws, the United States and other countries that did were unable extradite the virus writer to prosecute him for the damage done outside of the Philippines. Within six weeks after the Love Bug attack, the Philippines outlawed most computer crimes.

Investigations and Computer Evidence

Once an incident has occurred, the crime must be investigated. In most societies, the investigation of any crime deals with the gathering of evidence so that guilt or innocence may be proven in a court of law. In cyber space, this often proves very difficult. Evidence is the "testimony, writings, material objects, or other things presented to the senses that are offered to prove the existence or nonexistence of a fact."[41] Without evidence, there really is no way to prove a case. The problem with electronic evidence, unlike evidence in many traditional crimes, is that it is highly perishable and can be removed or altered relatively easily from a remote location. The collection of useful evidence can be further complicated because it may not be retained for any meaningful duration, or at all, by involved parties. For example, Internet service providers (ISPs) may not maintain audit trails, either because their governments may not allow extended retention for privacy reasons, or the ISP may delete it for efficiency purposes. At this time, most countries do not require ISPs to retain electronic information for evidentiary purposes. These audit trails can be essential for tracing a crime back to a guilty party.

In instances where the investigation involves more than one country, the investigators have further problems because they now need to coordinate and cooperate with foreign entities. This often takes a considerable amount of time and a considerable amount of legal wrangling to get foreign authorities to continue with or cooperate in the investigation.

Assuming that it is possible to locate evidence pertaining to a cyber-crime, it is equally important to have the ability to collect and preserve it in a manner that maintains its integrity and undeniable authenticity. Because the evidence in question is electronic information, and electronic information is easily modified, created, and deleted, it becomes very easy to question its authenticity if strict rules concerning custody and forensics are not followed.

Jurisdiction and Venue

After the evidence has been collected and a case is made, a location for trial must be chosen. Jurisdiction is defined as "the authority given by law to a court to try cases and rule on legal matters within a particular geographic area and/or over certain types of legal cases."[42] Because cyber-crime is geographically complex, jurisdiction becomes equally complex — often involving multiple authorities, which can create a hindrance to an investigation. The venue is the proper location for trial of a case, which is most often the geographic locale where the crime was committed. When cyber-crime is considered, jurisdiction and venue create a complex situation. Under which state or nation's laws is a cyber-criminal prosecuted when the perpetrator was physically located in one place and the target of the crime was in another? If a cyber-criminal in Brazil attacked a system in the United States via a pirated system in France, should the United States or France be the venue for the trial? They were both compromised. Or should Brazil hold the trial because the defendant was physically within its geographic boundaries during the crime?

Extradition

Once jurisdiction is determined and a location for trial is set, if the defendant is physically located in a different state or nation than the venue for trial, that person must be extradited. *Black's Law Dictionary* defines extradition as "the surrender by one state or country to another of an individual accused or convicted of an offense outside its own territory and within the territorial jurisdiction of the other, which being competent to try and punish him, demands the surrender."[43] As seen by the Love Bug case, extradition efforts can become unpredictable if cyber-crime laws are not criminalized and especially if extradition laws are not established or modified to take cyber-crime into consideration. As an example, the United States requires, by constitutional law, that an extradition treaty be signed and that these treaties must either list the specific crimes covered by it or require dual criminality, whereby the same law is recognized in the other country.[44] Because the United States only has approximately 100 extradition treaties, and most countries do not yet have comprehensive computer crime laws, extradition of a suspected cyber-criminal to the United States may not be possible.

Technical Issues

The technical roadblocks that may hinder the ability of nations to mitigate the cyber-crime threat primarily concern the tools and knowledge used in the electronic domain of cyber space. Simply put, law enforcement often lacks the appropriate tools and knowledge to keep up with cyber-criminals.

The Internet is often referenced as the World Wide Web (WWW). However, information security professionals often refer to the WWW acronym as the *Wild Wild Web*. Although some countries do their best to regulate or monitor usage of the Internet, it is a difficult environment for any one country to exercise power over. For every control that is put in place, a workaround is found. One example exists for countries that wish to restrict access to the Internet. Saudi Arabia restricts access to pornography, sites that the government considers defamatory to the country's royal family or to Islam, and usage of Yahoo! chat rooms or Internet telephone services on the World Wide Web.[45] Reporters Without Borders, a media-rights advocacy group based in France, estimates that at least 20 countries significantly restrict Internet access.[46] SafeWeb, a small Oakland, California, company, provides a Web site that allows Internet users to mask the Web site destination. SafeWeb is only one of many such companies; and although the Saudi government has retaliated by blocking the SafeWeb site, other sites appear quickly that either offer the same service as SafeWeb or mirror the SafeWeb site so that it is still accessible. This is one example of a service that has legitimate privacy uses and is perfectly legal in its country of origin. However, it is creating a situation for Saudi Arabia and other countries whereby they are unable to enforce their own laws. Although some may argue that Saudi citizens should have the ability to freely access the Internet, the example given is not intended for arguing ethics but purely to serve as an example of the increasing inability of law enforcement to police what is within its jurisdiction. The same tool in the hands of a criminal can prevent authorities with legal surveillance responsibilities from monitoring criminal activity.

SafeWeb is but one example of a large number of tools and processes used for eluding detection. Similarly, encryption can be used to conceal most types of information. Sophisticated encryption programs were once

solely used by governments but are now readily available for download off of the Internet. If information is encrypted with a strong cryptography program, it will take authorities months or possibly years of dedicated computing time to reveal what the encryption software is hiding. Also available from the Internet is software that not only searches for system vulnerabilities, but also proceeds to run an attack against what it has found; and if successful, it automatically runs subsequent routines to hide traces of the break-in and to ensure future access to the intruder.

These types of tools make investigation and the collection of evidence increasingly more difficult. Until more effective tools are developed and made available to facilitate better detection and deterrence of criminal activities, criminals will continue to become more difficult to identify and capture.

Part 2: International Efforts to Mitigate Cyber-Crime Risk

The cyber-crime threat has received the attention of many different organizations, including national and local governments, international organizations such as the Council of Europe and the United Nations, and nongovernmental organizations dealing with issues such as privacy, human rights, and those opposed to government regulation.

General Government Efforts

> We are sending a strong signal to would-be attackers that we are not going to let you get away with cyber-terrorism.[47]
>
> **— Norman Mineta**
> **Former Secretary of Commerce**

> One thing that we can learn from the Atomic Age is that preparation, a clear desire and a clear willingness to confront the problem, and a clear willingness to show that you are prepared to confront the problem is what keeps it from happening in the first place.[48]
>
> **— Condoleezza Rice**
> **National Security Advisor**

Governments around the world are in an unenviable position. On one hand, they need to mitigate the risk imposed by cyber-crime in an environment that is inherently difficult to control; on the other hand, nongovernmental organizations are demanding limited government interference.

The first order of business for national governments is to take the lead in creating a cyber-crime regime that can coordinate the needs of all the world's citizens and all of the nation's interests in fighting the cyber-crime threat. To date, industry has taken the lead; and in effect, government has in a large part ceded public safety and national security to markets.

Many efforts have been made by various nations to create legislation concerning computer crime. The first was a federal bill introduced in 1977 in the Congress by Senator Ribikoff, although the bill was not adopted.[49] The United States later passed the 1984 Computer Fraud and Abuse Law, the 1986 Computer Fraud and Abuse Act, and the Presidential Decision Directive 63 (PDD-63), all of which resulted in strengthened U.S. cyber-crime laws. Internationally, in 1983 the OECD made recommendations for its member countries to ensure that their penal legislation also applied to certain categories of computer crime. The Thirteenth Congress of the International Academy of Comparative Law in Montreal, the U.N.'s Eighth Criminal Congress in Havana, and a Conference in Wurzburg, Germany, all approached the subject in the early 1990s from an international perspective. The focus of these conferences included modernizing national criminal laws and procedures; improvement of computer security and prevention measures; public awareness; training of law enforcement and judiciary agencies; and collaboration with interested organizations of rules and ethics in the use of computers.[50] In 1997, the High-Tech Subgroup of the G-8's Senior Experts on Transnational Organized Crime developed Ten Principles and a plan of action for combating computer crime. This was followed in 1999 by the adoption of principles of transborder access to stored computer data by the G-8 countries. The Principles and action plan included:[51]

- A review of legal systems to ensure that telecommunication and computer system abuses are criminalized

- Consideration of issues created by high-tech crimes when negotiating mutual assistance agreements and arrangements
- Solutions for preserving evidence prior to investigative actions
- Creation of procedures for obtaining traffic data from all communications carriers in the chain of a communication and ways to expedite the passing of this data internationally
- Coordination with industry to ensure that new technologies facilitate national efforts to combat high-tech crime by preserving and collecting critical evidence

Around the globe, countries are slowly developing laws to address cyber-crime, but the organization that has introduced the most far-reaching recommendations has been the Council of Europe (CoE). The Convention on Cyber-Crime was opened for signature on November 23, 2001, and is being ratified by its 41 member states and the observing states — Canada, United States, and Japan — over a one- to two-year period. The treaty will be open to all countries in the world to sign once it goes into effect. The impact of the treaty has the potential to be significant considering that CoE members and observing countries represent about 80 percent of the world's Internet traffic.[52]

Council of Europe Convention

The objective of the Council of Europe's Convention on Cyber-Crime is aimed at creating a treaty to harmonize laws against hacking, fraud, computer viruses, child pornography, and other Internet crimes and ensure common methods of securing digital evidence to trace and prosecute criminals.[53] It will be the first international treaty to address criminal law and procedural aspects of various types of criminal behavior directed against computer systems, networks or data, and other types of similar misuse.[54] Each member country will be responsible for developing legislation and other measures to ensure that individuals can be held liable for criminal offenses as outlined in the treaty. The Convention has been drafted by the Committee of Experts on Crime in Cyberspace (PC-CY) — a group that is reportedly made up of law enforcement and industry experts. The group worked in relative obscurity for three years, released its first public draft — number 19 — in April 2000, and completed its work in December 2000 with the release of draft number 25. The Convention was finalized by the Steering Committee on European Crime Problems and submitted to the Committee of Ministers for adoption before it was opened to members of the Council of Europe, observer nations, and the world at large.

The Convention addresses most of the important issues outlined in this chapter concerning cyber-crime. As previously described, the major hurdles in fighting cyber-crime are the lack of national laws applicable to cyber-crime and the inability for nations to cooperate when investigating or prosecuting perpetrators.

National Law

At a national level, all signatory countries will be expected to institute comprehensive laws concerning cyber-crime, including the following:

- Criminalize "offenses against the confidentiality, integrity and availability of computer data and systems," "computer-related offenses," and "content-related offenses."
- Criminalize the "attempt and aiding or abetting" of computer-related offenses.
- Adopt laws to expedite the preservation of stored computer data and "preservation and partial disclosure of traffic data."
- Adopt laws that empower law enforcement to order the surrender of computer data, computer systems, and computer data storage media, including subscriber information provided by an ISP.
- Adopt laws that provide law enforcement with surveillance powers over "content data" and require ISPs to cooperate and assist.
- Adopt legislation that establishes jurisdiction for computer-related offenses.

International Cooperation

The section of the Convention dealing with international cooperation concerns the development and modification of arrangements for cooperation and reciprocal legislation. Some of the more interesting elements include the following:

- Acceptance of criminal offenses within the Convention as extraditable offenses even in the absence of any formal extradition treaties. If the extradition is refused based on nationality or jurisdiction over the offense, the "requested Party" should handle the case in the same manner as under the law of the "requesting Party."
- Adoption of legislation to provide for mutual assistance to the "widest extent possible for the purpose of investigations or proceedings concerning criminal offenses related to computer systems and data, or for the collection of evidence in electronic form of a criminal offense."[55]
- In the absence of a mutual assistance treaty, the "requested Party" may refuse if the request is considered to be a political offense or that execution of the request may likely risk its "sovereignty, security or other essential interests."

NGO Responses and Criticisms

> We don't want to pass a text against the people.[56]
>
> **— Peter Csonka**
> **Deputy Head of the Council of**
> **Europe's Economic Crime Division**

> The experts should be proud of themselves. They have managed during the past eight months to resist pernicious influence of hundreds if not thousands of individual computer users, security experts, civil liberties groups, ISPs, computer companies and others outside of their select circle of law enforcement representatives who wrote, faxed and e-mailed their concerns about the treaty.[57]
>
> **— David Banisar**
> **Deputy Director of Privacy International**

> We don't have any comment regarding these protestings. Everyone is entitled to their own opinion, but we have no comment.[58]
>
> **— Debbie Weierman**
> **FBI Spokeswoman**

Within days of the CoE's release of its first public draft of the Convention on Cyber-Crime, as well as the release of its subsequent versions, opposition groups rallied together and flooded the Council with requests urging the group to put a hold on the treaty. The 22nd draft received over 400 e-mails.[59] The Global Internet Liberty Campaign, an organization consisting of 35 lobby groups ranging from Internet users to civil liberties activists and anti-censorship groups, wrote to the European Council stating that they "believe that the draft treaty is contrary to well-established norms for the protection of the individual (and) that it improperly extends the police authority of national governments."[60] Member organizations represent North America, Asia, Africa, Australia, and Europe, and include the American Civil Liberties Union, Privacy International (United Kingdom), and Human Rights Network (Russia). Other groups opposed to the proposed treaty are the International Chamber of Commerce, all the ISP associations, and data security groups that are concerned with some key areas regarding human rights, privacy, and the stifling of innovation.

Lack of NGO Involvement

The primary concern — and the problem from which all the others stem — is the fact that the PC-CY worked in seclusion without the involvement of important interest groups representing human rights, privacy, and industry. According to opposition sources, the PC-CY is comprised of "police agencies and powerful private interests."[61] A request by the author was made to the CoE for a list of PC-CY members; however, the request was declined, stating that they "are not allowed to distribute such a list."[62] Throughout the entire period during which the PC-CY was drafting the treaty, not a single open meeting was held. Marc Rotenberg of the Electronic Privacy Information Center called the draft a "direct assault on legal protections and constitutional protections that have been established by national governments to protect their citizens."[63] If the three years of work done by the PC-CY were more inclusive and transparent, many if not all of the remaining issues could have already

been addressed. Unfortunately, although opposition has been expressed, little has been done to address the issues raised; and the Council of Europe passed the Convention regardless.

Overextending Police Powers and Self-Incrimination

A chief concern of many opposition groups is that the Convention extends the power of law enforcement beyond reasonable means and does not provide adequate requirements to ensure that individual rights are preserved. The Global Internet Liberty Campaign points out that an independent judicial review is not required before a search is undertaken. Under Article 19 of the Convention, law enforcement is empowered to search and seize any computer system within its territory that it believes has data that is lawfully accessible or available to the initial system. With today's operating systems and their advanced networking capabilities, it is difficult to find a computer system without a network connection that would make it accessible to any other system. The only question remaining is whether that access is "lawful." If law enforcement draws the same conclusion, where might they stop their search? Such a broad definition of authority can implicate nearly any personal computer attached to the Internet. Furthermore, Article 19 gives law enforcement the power to order any person who has knowledge about the functioning of the computer system, or measures applied to protect the computer data therein, to provide any information necessary to grant access. This would easily include encryption keys or passwords used to encrypt information. To date, only Singapore and Malaysia are believed to have introduced such a requirement into law. The required disclosure of such information to some people might seem to be contrary to U.S. law and the Fifth Amendment, which does not require people to incriminate themselves.

Privacy

The Convention requires that ISPs retain records regarding the activities of their customers and to make that information available to law enforcement when requested. The Global Internet Liberty Campaign letter to the CoE stated, "these provisions pose a significant risk to the privacy and human rights of Internet users and are at odds with well-established principles of data protection such as the Data Protection Directive of the European Union." They argue that such a pool of information could be used "to identify dissidents and persecute minorities." Furthermore, for ISPs to be able to provide such information, the use of anonymous e-mailers and Web surfing tools such as SafeWeb would need to be outlawed because they mask much of the information that ISPs would be expected to provide.

ISP organizations have also taken exception to the proposed requirements, which would place a heavy responsibility on them to manage burdensome record-keeping tasks as well as capture and maintain the information. In addition, they would be required to perform the tasks necessary to provide the requested information.

Mutual Assistance

Under the Convention's requirements, countries are not obligated to consider dual criminality to provide mutual assistance. That is, if one country believes that a law under the new Convention's guidelines is broken and the perpetrator is in foreign territory, that foreign country, as the "requested nation," is required to assist the "requesting nation," regardless of whether a crime was broken in the requested nation's territory. The "requested nation" is allowed to refuse only if they believe the request is political in nature. What will happen if there is a disagreement in definition? In November of 2001, Yahoo! was brought to trial in France because it was accused of allowing the sale of Nazi memorabilia on its auction site — an act perfectly legal in the United States, Yahoo!'s home country. Barry Steinhardt, associate director for the American Civil Liberties Union, asked, "Is what Yahoo! did political? Or a 'crime against humanity,' as the French call it?" Germany recently announced that anyone, anywhere in the world, who promotes Holocaust denial is liable under German law; and the Malaysian government announced that online insults to Islam will be punished.[64] How will this impact national sovereignty over any country's citizens when that country legally permits freedom of speech?

Stifling of Innovation and Safety

Article 6 of the Convention, titled "Misuse of Devices," specifically outlaws the "production, sale, procurement for use, import, distribution or otherwise making available of, a device, including a computer program,

designed or adapted primarily for the purpose of committing any of the offences established (under Title 1)." The devices outlawed here are many of the same devices that are used by security professionals to test their own systems for vulnerabilities. The law explains that the use of such devices is acceptable for security purposes provided the device will not be used for committing an offense established under Title 1 of the Convention. The problem with the regulation is that it may prohibit some individuals or groups from uncovering serious security threats if they are not recognized as authorities or professionals. The world may find itself in a position whereby it must rely on only established providers of security software. They, however, are not the only ones responsible for discovering system vulnerabilities. Quite often, these companies also rely on hobbyists and lawful hacker organizations for relevant and up-to-date information. Dan Farmer, the creator of the free security program "SATAN," caused a tremendous uproar with his creation. Many people saw his program solely as a hacking device with a purpose of discovering system weaknesses so that hackers could exploit them. Today, many professionals use that tool and others like it in concert with commercially available devices to secure systems. Under the proposed treaty, Dan Farmer could have been labeled a criminal and possession of his program would be a crime.

Council of Europe Response

Despite the attention that the draft Convention on Cyber-Crime has received, CoE representatives appear relatively unconcerned; and the treaty has undergone minimal change. Peter Csonka, the CoE deputy head, told Reuters, "We have learned that we have to explain what we mean in plain language because legal terms are sometimes not clear."[65] It is interesting to note that members of the Global Internet Liberty Campaign — and many other lobby groups that have opposed elements of the Convention — represent and include in their staff and membership attorneys, privacy experts, technical experts, data protection officials, and human rights experts from all over the world. The chance that they all may have misinterpreted or misread the convention is unlikely.

Part 3: Approaches for Internet Rule

The effects of globalization have increasingly challenged national governments. Little by little, countries have had to surrender their sovereignty in order to take advantage of gains available by global economic and political factors. The Council of Europe's Convention on Cyber-Crime is a prime example. The advent of the Internet and global communications networks have been responsible for tearing down national borders and permitting the free flow of ideas, music, news, and possibly a common culture we can call cyber-culture. Saudi Arabia is feeling its sovereignty threatened and is attempting to restrict access to Web sites that it finds offensive. France and Germany are having a difficult time restricting access to sites related to Nazism. And all countries that are taking full advantage of the digital age and its tools are threatened by cyber-criminals, whether they are a neighborhood away or oceans away. Sovereign nations are choosing to control the threat through the CoE's cyber-crime treaty. Is this the only option for governing the Internet? No, not necessarily. The following is a selection of possible alternatives.

Anarchic Space

The Internet has remained relatively unregulated. Despite government attempts, Saudis can still access defamatory information about the Saudi royal family; and U.S. citizens are still able to download copyrighted music regardless of restrictions placed on Napster. It is possible that the Internet could be treated as anarchical space beyond any control of nations. This, however, does not solve the cyber-crime problem and could instead lead to an increase in crime.

Supranational Space

On the opposite end of the spectrum, a theoretical possibility is that of the Internet as supranational space. Under this model, a world governing body would set legislation and controls. Because no world government actually exists, this not a realistic option.

National Space

A more probable approach is the treatment of the Internet as national space, wherein individual nations would be responsible for applying their own territorial laws to the Internet. This, unfortunately, has been an approach that seems to be favored by the more powerful nations such as the United States, but it has little effect without coordination and cooperation from other nations and nongovernmental organizations (NGOs).

Epistemic Communities

Another option for Internet rule could be to establish an epistemic community — a "knowledge-based transnational community of experts with shared understandings of an issue or problem or preferred policy responses."[66] This has been a successful approach leading up to the Outer Space Treaty and the Antarctica Treaty. The Outer Space Treaty claims outer space as the "province of mankind"[67] and the Antarctica Treaty "opens the area to exploration and scientific research, to use the region for peaceful purposes only, and to permit access on an equal, nondiscriminatory basis to all states."[68] Scientists specializing in space and ocean sciences have driven much of the decision making that has taken place. A similar approach was used in the computing environment when decisions were made on how to make the Internet handicap accessible. Experts gathered with an understanding of the issue and implemented systems to manage the problem. However, as has been discussed, national governments have an interest in controlling particular aspects of the Internet; and an epistemic community does not provide them the control they desire. Therefore, the success of an epistemic solution in resolving the cyber-crime threat is unlikely.

International Regimes

The most obvious choice for Internet rule — bearing in mind its borderless nature and the interest of states to implement controls and safeguards — is an international regime. According to the noted regime theory expert Stephen Krasner, a regime is defined as "sets of implicit or explicit principles, norms, rules, and decision-making procedures around which actors' expectations converge in a given area of international relations."[69] In fact, it can be argued that a regime is already in the making concerning Internet rule and cyber-crime, and that the Council of Europe's Convention on Cyber-Crime represents the regime's set of explicit "rules." Regrettably, the rules outlined by the Convention do not represent the principles of all the actors. The actors concerning Internet rule extend beyond national governments and include all of the actors that have been described previously, including individual users, privacy and human rights advocates, corporations, ISPs, and, yes, national governments. The Convention was created solely by government representatives and therefore has ignored these other important actors. If a cyber-crime regime did exist that included all interested parties or actors, the principles, norms, rules, and decision-making procedures would be different than what is currently represented in the CoE cyber-crime treaty.

The principles — "beliefs of fact, causation, and rectitude"[70] — for a government-based regime as witnessed in the Convention are primarily concerned with preservation of sovereignty. The focus of the Convention is based on the needs of government-based law enforcement for pursuing and capturing the agent responsible for limiting state sovereignty — the cyber-criminal. A treaty drafted by a fully represented regime would include recommendations and regulations that consider the need for unhindered innovation and the preservation of privacy and basic human rights. Such a regime would also foster discussions that could take place concerning the detrimental effects of criminalizing hacking tools and maintaining communications records for all Internet users.

The norms — "standards of behavior defined in terms of rights and obligations"[71] — for the government-based regime once again center on the need to pursue and deter cyber-criminals. The articles addressing mutual assistance explicitly define the obligations and rights of states concerning jurisdiction, extradition, and extra-territoriality, while paying little respect to the rights of individuals under their own territorial laws. A fully represented regime could table issues concerning the need for dual criminality.

The rules — "specific prescriptions or proscriptions for action" — that would be included in a government-based regime are now painfully evident. Although most of the convention rules are necessary for addressing the cyber-crime problem, their lack of sensitivity to nongovernmental interests is clear.

Finally, the decision-making procedures — prevailing practices for making and implementing collective choice — are obviously absent of any representation outside of government interests. If it were possible to roll back time by three years — and instead of having closed-door sessions with minimal representation, have open

meetings that practiced transparency in all of its dealings and invited representation of all actors involved in Internet activity — the Convention would most likely be a treaty that truly represented the opinions of the collective Internet community.

Part 4: Formula for Success

It is surprising that the CoE, an organization that proclaims one of its primary aims to be "to protect human rights,"[72] would ignore the basic principles of regime theory and the success factors of thriving international regimes, instead prescribing rules that primarily cater to the needs of law enforcement.

One of the more obvious examples of a successful regime is based on the Montreal Protocol on Substances that Deplete the Ozone Layer signed in 1987. As a result of the Montreal Protocol, industries have developed safer, cleaner methods for handling ozone-depleting chemicals and pollution-prevention strategies.[73] The success of this regime can be directly attributed to the cooperation and coordination among all relevant actors, including government, industry, and environmental sciences.

The Convention on Cyber-Crime is open for signatures, the opposition has spoken, and it appears that the only thing standing in the way of the treaty becoming law is the final ratification and introduction of national laws by individual countries. It is now too late for the cyber-crime treaty to truly represent the opinions of all the primary actors, but it is still possible for individual nations to protect the interests of its citizenry. Pressure on the more powerful nations may be enough to make sure that what is adopted will include appropriate measures and safeguards. Unfortunately, many countries do not have a very good history of keeping the best interests of its citizens in mind when they create their laws. Regardless of the ultimate outcome of the treaty, a broadly represented regime is vital to future success in fighting the cyber-crime threat. Although the Convention may not be an ideal solution, it is possible that the introduction of the Convention on Cyber-Crime and the worldwide attention that it has brought to cyber-crime will be the catalyst for finally establishing an effective cyber-crime regime — one that truly represents all actors.

Notes

1. Minihan, K.A., "Defending the Nation against Cyberattack: Information Assurance in the Global Environment," USIA, U.S. Foreign Policy Agenda, Nov. 1998, p. 1. <http://usinfo. state.gov/journals/itps/1198/ijpe/pj48min.htm> Feb. 27, 2001.
2. Excerpt from the source file posted by the computer hacking group "Hacking 4 Girliez." The text was displayed on the defaced *New York Times* Web site, September 13, 1998.
3. "Hacking Around, A NewsHour Report on Hacking." *The NewsHour with Jim Lehrer.* May 8, 1998. PBS Online. Apr. 16, 2001.
4. The term "cyber space" was first used by author William Gibson in his 1984 science fiction novel, *Neuromancer.*
5. Steiner, P. "A Dog, Sitting at a Computer Terminal, Talking to Another Dog." Cartoon. *The New Yorker,* Jul. 5, 1993
6. Schiller, J. "Profile of a Hacker." *The NewsHour with Jim Lehrer.* PBS Online. May 8, 1998. Transcript.<http://www.pbs.org/newshour/bb/cyberspace/jan-june98/hacker_profile.html> Mar. 14, 2001, p. 1.
7. The annual "CSI/FBI Computer Crime and Security Survey" for 2000 is based on the responses from 643 computer security practitioners in U.S. corporations and government agencies.
8. Power, R. "2000 CSI/FBI Computer Crime and Security Survey." *Computer Security Journal,* XVI(2), 45, Spring 2000.
9. "Russia's Hackers: Notorious or Desperate?" CNN.com. Nov. 20, 2000. <http://www.cnn. com/2000/TECH/computing/11/20/russia.hackers.ap/index.html>. Jan. 25, 2001, p. 1.
10. "Russia's Hackers: Notorious or Desperate?" CNN.com. Nov. 20, 2000. <http://www.cnn. com/2000/TECH/computing/11/20/russia.hackers.ap/index.html> .Jan. 25, 2001, p. 1.
11. "10 Foreign Hot Spots for Credit Card Fraud." *Internet World.* Feb. 1, 1999. Infotrac. Mar. 24, 2001, p. 1.
12. The London School of Economics and Political Science. "Cybercrime: The Challenge to Leviathan?" Feb. 27, 2001, p. 1.
13. The London School of Economics and Political Science. "Cybercrime: The Challenge to Leviathan?" Feb. 27, 2001, p. 1.

14. The London School of Economics and Political Science. "Cybercrime: The Challenge to Leviathan?" Feb. 27, 2001, p. 1.
15. Freeh, L.J. "Statement for the Record of Louis J. Freeh, Director, Federal Bureau of Investigation on Cybercrime before the Senate Committee on Judiciary Subcommittee for the Technology, Terrorism, and Government Information." Department of Justice, Mar. 28, 2000. <http://www. usdoj.gov/criminal/cybercrime/freeh328.htm> Jan. 26, 2002.
16. IMRG Interactive Media in Retail Group. "Napster Offers $1 Billion to Record Companies." Feb. 21, 2001. <http://www.imrg. org/imrg/imrgreports.ns> April 1, 2001, p. 1.
17. Computer Crime and Intellectual Property Section (CCIPS) of the Criminal Division of the U.S. Department of Justice. Computer Intrusion Cases. Mar. 31, 2001. <http://www.cybercrime.gov/cccases.html>, p. 1.
18. The Affidavit for Robert Hanssen's arrest is available online at <http://www.fas.org/irp/ops/ci/hanssen_affidavit.html>.
19. Godoy, J. "Computers and International Criminal Law: High Tech Crimes and Criminals." *Lexis Nexis,* 2000. New England International and Comparative Law Annual. Mar. 24, 2001. <http://Web.lexis-nexis.com/universe/document?_ansset>.
20. Minihan, K.A. "Defending the Nation against Cyberattack: Information Assurance in the Global Environment." USIA, U.S. Foreign Policy Agenda. Nov. 1998, p. 1. <http://usinfo. state.gov/journals/itps/1198/ijpe/pj48min.htm> Feb. 27, 2001.
21. Vise, D.A. "FBI Sees Rising Threat from Computer Crime." *Lexis Nexis,* Mar. 21, 2001. *International Herald Tribune,* Mar. 24, 2001, p. 1.
22. Vise, D.A. "FBI Sees Rising Threat from Computer Crime." *Lexis Nexis,* Mar. 21, 2001. *International Herald Tribune,* Mar. 24, 2001, p. 1.
23. Charney, S. "The Internet, Law Enforcement and Security." Internet Policy Institute. Feb. 27, 2001, p. 1. <http://www.internetpolicy.org/briefing/charney.html>.
24. Denning, D. "Reflections on Cyberweapons Controls." *Computer Security Journal.* XVI(4), 1, Fall 2000.
25. Denning, D. "Reflections on Cyberweapons Controls." *Computer Security Journal.* XVI(4), 1, Fall 2000.
26. Denning, D. "Reflections on Cyberweapons Controls." *Computer Security Journal.* XVI(4), 1, Fall 2000.
27. U.S. Department of Justice, "Juvenile Computer Hacker Cuts Off FAA Tower at Regional Airport." Press Release. Mar. 18, 1998, p. 1. <http:/www.cybercrime.gov/juvenilepld.htm> Jan. 4, 2001.
28. Information Technology Association of America, "Industry Partnerships to Combat Cyber Crime Take on Bold Agendas." *InfoSec Outlook*. Feb. 27, 2001, p. 1. <http://www.itaa.org/infosec/pubs/ISArticle.cfm?ID=73>.
29. Attrition.Org maintains defacement counts and percentages, by domain suffix for worldwide Internet Web site defacement <www.attrition.org>. Attrition.Org. *Defacement Counts and Percentages*, by Domain Suffix. Mar. 31, 2001. <http://www.attrition.org/mirror/attrition/country.html>.
30. Denning, D. "Reflections on Cyberweapons Controls." *Computer Security Journal.* XVI(4), 43, Fall 2000.
31. Ticehurst, J. "Cybercrime Soars in the UK." Vnunet.com. Nov. 6, 2000, p. 1. <http://www. vnunet.com/News/1113497> Jan. 25, 2001.
32. Vise, D.A. "FBI Sees Rising Threat from Computer Crime." *Lexis Nexis,* Mar. 21, 2001, p. 1. *International Herald Tribune,* Mar. 24, 2001.
33. Kelsey, D. "GartneróHalf of All Small Firms Will Be Hacked." Newsbytes. Oct. 11, 2000, p. 1. <http://www.newsbytes.com/pubNews/00/156531.html> Mar. 27, 2001.
34. Konrad, R. "Hack Attacks a Global Concern." CNET New.com. Oct. 29, 2000, p. 1. <http://news.cnet.com/news/0-1003-200-3314544.html?tag+rltdnws> Feb. 27, 2001.
35. Konrad, R. "Hack Attacks a Global Concern." CNET New.com. Oct. 29, 2000, p. 1. <http://news.cnet.com/news/0-1003-200-3314544.html?tag+rltdnws> Feb. 27, 2001.
36. "Reno Urges Crackdown on Cybercrime in The Americas." <www.FreeRepublic.com> Nov. 27, 1998, p. 1. Fox News Network. <http://www.freerepublic.com/forum/a365e8c3e6753.htm> Feb. 27, 2001.
37. "Many Countries Said to Lack Computer Crime Laws." CNN.com. Jul. 26, 2000, p. 1. <http://www.cnn.com/2000/TECH/computing/07/26/crime.internet.reut/> Jan. 25, 2001.

38. Schjolberg, S. "Penal Legislation in 37 Countries." Moss Bryett, Moss City Court Web site. Feb. 22, 2001, p. 1. <http://www.mossbyrett.
of.no/info/legal.html> April 14, 2001.
39. McConnell International with Support from WITSA. *Cyber Crime … and Punishment? Archaic Laws Threaten Global Information.* McConnell International LLC. Dec. 2000, p. 5.
40. McConnell International with Support from WITSA. *Cyber Crime … and Punishment? Archaic Laws Threaten Global Information.* McConnell International LLC. Dec. 2000, p. 6.
41. Black, H., Campbell, M.A., Nolan, J.R., and Connolly, M.J. *Black's Law Dictionary,* fifth edition. St. Paul: West Publishing Co.,1979, p. 489.
42. *Law.Com Legal Dictionary.* Apr. 25, 2001, p. 1. <http://www.law.com>.
43. Black, H., Campbell, M.A., Nolan, J.R., and Connolly, M.J. *Black's Law Dictionary,* fifth edition. St. Paul: West Publishing Co.,1979, p. 528.
44. Godoy, J. "Computers and International Criminal Law: High Tech Crimes and Criminals." *Lexis Nexis,* 2000. New England International and Comparative Law Annual. Mar. 24, 2001, p. 1. <http://Web.lexis-nexis.com/universe/document?_ansset>.
45. Lee, J. "Punching Holes in Internet Walls." *New York Times,* Apr. 26, 2001, p. G1.
46. Lee, J. "Punching Holes in Internet Walls." *New York Times,* Apr. 26, 2001, p. G1.

150

Honeypot Essentials

Anton Chuvakin, Ph.D., GCIA, GCIH

This chapter discusses honeypot (and honeynet) basics and definitions, and then outlines important implementation and setup guidelines. It also describes some of the security lessons a company can derive from running a research honeypot, based on this author's experience running a research honeypot. This chapter also provides insight into the techniques of attackers and concludes with considerations useful for answering the question, "Should your organization deploy a honeynet?"

Introduction and Background

Although known to security processionals for a long time, honeypots recently became a hot topic in information security. However, the amount of technical information available on their setup, configuration, and maintenance is still sparse, as are qualified people able to run them. In addition, higher-level guidelines (such as need and business-case determination) are similarly absent.

What is a honeypot? Lance Spitzner, a founder of the Honeynet Project (http://project.honeynet.org/) defines a honeypot as "a security resource whose value lies in being probed, attacked, or compromised." The Project differentiates between *research* and *production* honeypots. The former are focused on gaining intelligence information about attackers and their technologies and methods, and the latter are aimed at decreasing the risk to company IT resources and providing advance warning about the incoming attacks on the network infrastructure. Honeypots of any kind are difficult to classify using the "prevention–detection–response" metaphor, but it is hoped that after reading this chapter their value will become clearer to readers.

This chapter focuses on operating a research honeypot or a "honeynet." The term "honeynet," used in this chapter originated in the Honeynet Project and means a network of systems with fairly standard configurations connected to the Internet. The only difference between such a network and a regular production network is that all communication is recorded and analyzed, and no attacks targeted at third parties can escape the network. Sometimes, the system software is slightly modified to help deal with encrypted communication, often used by attackers. The systems are never "weakened" for easier hacking but are often deployed in default configurations with a minimum of security patches. They might or might not have known security holes. The Honeynet Project defines such honeypots as "high-interaction honeypots," meaning that attackers interact with a deception system exactly as they would with a real victim machine. On the other hand, various honeypot and deception daemons are "low-interaction," as they provide only an illusion to an attacker, and one that can hold their attention for a short time only. Such honeypots have value as an early attack indicator but do not yield in-depth information about the attackers.

Research honeypots are set up with no extra effort to lure attackers — blackhats locate and exploit systems on their own. It happens due to the widespread use of automatic hacking tools, such as fast, multiple vulnerability scanners and automatic penetration scripts. For example, an attacker from our honeynet has attempted to scan 200,000 systems for a single FTP vulnerability in one night using such tools. Research honeypots are also unlikely to be used for prosecuting intruders; however, researchers are known to track hacker activities using various covert techniques for a long time after the intruder broke into their honeypot. In addition, prosecution based on honeypot evidence has never been tested in a court of law. It is still wise to involve the company's legal team before setting up such a hacker study project.

Overall, the honeypot is the best tool for looking into malicious hacker activity. The reason is simple: all communication to and from the honeynet is malicious by definition. No data filtering, no false-positives, and no false-negatives (the latter only if the data analysis is adequate) are obscuring the picture. Watching the honeypot provides insight into intruders' personalities and can be used to profile attackers. For example, during the summer of 2002, the majority of penetrated Linux honeypots were hacked by Romanian attackers.

Setting Up a Honeypot

What are some of the common-sense prerequisites for running a honeynet? First, a honeypot is a sophisticated security project, and it makes sense to take care of security basics first. If your firewall crashes or your intrusion detection system (IDS) misses attacks, you are clearly not yet ready for a honeypot deployment. Running a honeypot also requires advanced knowledge in computer security. After running a honeynet for netForensics (http://www.netForensics.com) and as a member of the Honeynet Research Alliance, I can state that operating a honeynet presents the ultimate challenge a security professional can face. The reason is simple: no "lock it down and maintain secure state" model is possible for such a deception network. It requires in-depth expertise in many security technologies and beyond.

Additionally, a honeypot system should not be allowed to attack other systems or, at least, such ability should be minimized. This requirement often conflicts with a desire to create a more realistic environment for malicious hackers to "feel at home" so that they manifest a full spectrum of their behavior. Related to the above is the need for proper separation of a research honey network from company production machines. In addition to protecting innocent third parties, similar measures should be utilized to prevent attacks against your own systems from your honeypot. Honeypot systems should also have reliable out-of-band management. The main reason for having this capability is to be able to quickly cut off the network access to and from the honeypot in cases of emergency (and they do happen!), even if the main network connection is saturated by an attack. That sounds contradictory to the above statement about preventing outgoing attacks but Murphy's law might play a trick or two and "human errors" can never be totally excluded.

The Honeynet Research Alliance (http:// project.honeynet.org/alliance/) has guidelines on data control and data capture for the deployed honeynet. They distill the above ideas and guidelines into a well-written document "Honeynet Definitions, Requirements, and Standards" (http:// project.honeynet.org/alliance/requirements.html). This document establishes some "rules of the game," which have a direct influence on honeynet firewall rule sets and IDS policies.

Data control is a capability required to control the network traffic flow in and out of the honeynet in order to contain the blackhat actions within the defined policy. For example, rules such as "no outgoing connections," "limited number of outgoing connections per time unit," "only specific protocols and/or locations for outgoing connections," "limited bandwidth of outgoing connections," "attack string filtering in outgoing connections" or their combination can be used on a honeynet. Data control functionality should be multi-layered, allow for manual and automatic intervention (such as remote disabling of the honeypot), and make every effort to protect innocent third parties from becoming victims of attacks launched from the honeynet.

Data capture defines the information that should be captured on the honeypot system for future analysis, data retention policies, and standardized data formats that facilitate information sharing between the honeynets and cross-honeynet data processing. Cross-honeypot correlation is an extremely promising area of future research because it allows for the creation of an early warning system about new exploits and attacks. Data capture also covers the proper separation of honeypots from production networks to protect the attack data from being contaminated by the regular network traffic. Another important aspect of data capture is timely documentation of attacks and other incidents occurring in the honeypot. It is crucial for research to have a well-written log of malicious activities and configuration changes performed on the honeypot system.

Running a Honeynet

Consider some of the practical aspects of running a honeynet. Our example setup, a netForensics honeynet, consists of three hosts (see Exhibit 150.1): a victim host, a firewall, and an IDS. This is the simplest configuration to maintain. However, a workable honeynet can even be set up on a single machine if a virtual environment (such as VMWare or UML-Linux) is used. Combining IDS and firewall functionality using a gateway IDS (such as Hogwash) allows one to reduce the requirement to just two machines. A gateway IDS is a host with two

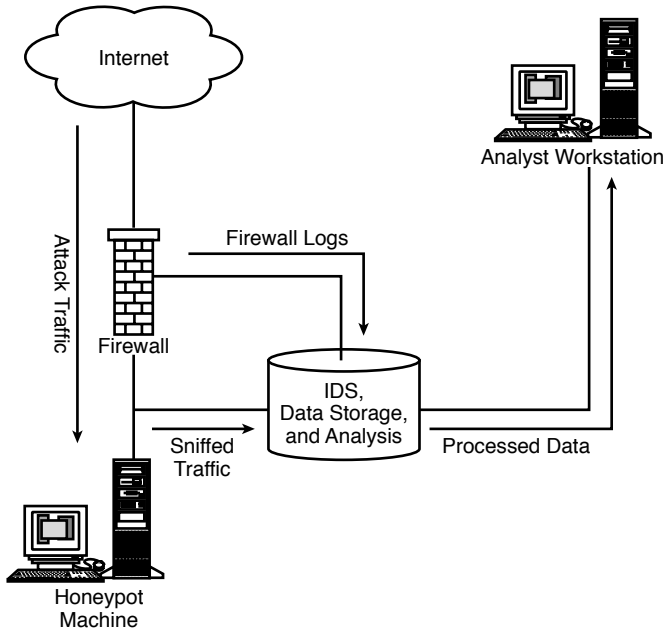


EXHIBIT 150.1 Example setup.

network cards that analyzes the traffic passing through it and can make packet-forwarding decisions (like a firewall) and send alerts based on network packet contents (like an IDS). Currently, the honeynet uses Linux on all systems but various other UNIX flavors will be deployed as "victim" servers by the time this chapter is published. Linux machines in default configurations are hacked often enough to provide a steady stream of data on blackhat activity. "Root"-level system penetration within hours of being deployed is not. UNIX also provides a safe choice for a victim system OS due to its higher transparency and ease of reproducing a given configuration.

The honeypot is run on a separate network connection — always a good idea because the deception systems should not be seen as owned by your organization. The firewall (hardened Linux "iptables" stateful firewall) allows and logs all the inbound connections to the honeypot machines and limits the outgoing traffic, depending on the protocol (with full logging as well). It also blocks all IP spoofing attempts and fragmented packets, which are often used to conceal the source of a connection or launch a denial-of-service attack. The firewall also protects the analysis network from attacks originating from the honeypot. In fact, in the above setup, an attacker has to pierce two firewalls to get to the analysis network. The IDS machine is also firewalled, hardened, and runs no services accessible from the untrusted network. The part of the rule set relevant to protecting the analysis network is very simple: no connections are allowed from the untrusted LAN to an analysis network. The IDS (Snort from www.snort.org) records all network traffic to a database and a binary traffic file via a stealth IP-less interface and also sends alerts on all known attacks detected by its wide signature base (approximately 1650 signatures as of July 2002). In addition, specially designed software is used to monitor the intruder's keystrokes and covertly send them to a monitoring station.

All data capture and data control functionality is duplicated as per Honeynet Project requirements. The 'tcpdump' tool is used as the secondary data capture facility, bandwidth-limiting device serves as the second layer of data control and the stealth kernel-level key logger backs up the keystroke recording. Numerous automated monitoring tools, some custom-designed for the environment, monitor the honeypot network for alerts and suspicious traffic patterns.

Data analysis is crucial for the honeypot environment. The evidence — in the form of system, firewall and IDS log files, IDS alerts, keystroke captures, and full traffic captures — is generated in overwhelming amounts. Events are correlated and suspicious ones are analyzed using the full packet dumps. It is highly recommended to synchronize the time via Network Time Protocol on all the honeypot servers for more reliable data correlation. netForensics software can be used to enable advanced data correlation and analysis. Unlike in the

production environment, having traffic data available in the honeypot is extremely helpful. It also allows for reliable recognition of new attacks. For example, a Solaris attack on the "dtspcd" daemon (TCP port 6112) was first captured in one of the Project's honeypots and then reported to CERT.

The above setup has gone through six system compromises, several massive outbound denial-of-service attacks (all blocked by the firewall!), major system vulnerability scanning, serving as an Internet Relay Chat server for Romanian hackers, and other exciting stuff. It passed with flying colors through all the above "adventures" and can be recommended for deployment.

Lessons Learned

What insight have we gained about the attacking side from running the honeynet? It is true that most of the attackers "caught" in such honeynets are "script kiddies"; that is, the less enlightened part of the hacker community. Although famous early honeypot stories (such as those described in Bill Cheswick's "An Evening with Berferd" and Cliff Stolls' "Cuckoo's Nest") dealt with advanced attackers, most honeypot experiences will probably be related to script kiddies. Opposite to common wisdom, companies do have something to fear from the script kiddies. The number of scans and attacks aimed by the attackers at Internet-facing networks ensures that any minor mistake in network security configuration will be discovered fairly soon. Every unsecured server running a popular operating system (such as Solaris, Linux, or Windows) will be taken over fairly soon. Default configurations and bugs in services (UNIX/Linux ssh, bind, ftpd, and now even Apache Web server and Windows IIS are primary examples) are the reason. We have captured and analyzed multiple attack tools using the above flaws. For example, a fully automated scanner that looks for 25 common UNIX vulnerabilities, runs hundreds of attack threads simultaneously, and deploys a rootkit on the system is one such tool. The software can be set to choose a random A class (16 million hosts) and first scan it for a particular network service (currently, FTP is the favorite, see http://www.dshield. org site for some global scan and attack statistics). Then on the second pass, the program collects FTP banners (such as "ftp. example.com FTP server (Version wu-2.6.1-16) ready") for target selection. On the third pass, the servers that had the misfortune of running a particularly vulnerable version of the FTP daemon are attacked, exploited, and back-doored for convenience. The owner of such a tool can return in the morning to pick up a list of IP addresses that he now "owns" (meaning, has privileged access to).

In addition, malicious attackers are known to compile Internet-wide databases of available network services, complete with their versions so that the hosts can be compromised quickly after the new software flaw is discovered. In fact, there is always a race between various groups to take over more systems. This advantage can come in handy in case of a local denial-of-service (DoS) war. While "our" attackers have not tried to draft the honeypot in their army of "zombie" bots, they did use it to launch old-fashioned, point-to-point DoS attacks (such as UDP, ping floods, and even the ancient modem hang-up ATH DoS).

Attacker behavior seems to indicate that attackers are accustomed to operating with no resistance. One attacker's first action was changing the "root" password on the system — clearly an action that will be noticed the next time the system admin tries to log in. Not a single attacker bothered to check for the presence of the Tripwire integrity checking system, which is included by default in many Linux distributions. On the next Tripwire run, all the "hidden" files are easily discovered. One more attacker had created a directory for himself as "/his-hacker-handle," something that every system admin worth his or her salt will see at once. The rootkits (i.e., hacker toolkits to maintain access to a system that include backdoors, Trojans, and common attack tools) now reach megabyte sizes and feature graphical installation interfaces suitable for novice blackhats. Research indicates that some of the script kiddies "own" networks consisting of hundreds of machines that can be used for DoS or other malicious purposes.

The exposed UNIX system is most often scanned for ports 111 (RPC services) and 21 (FTP). Recent (2000–2002) remote "root" bugs in those services account for this phenomenon. The system with a vulnerable FTP daemon is compromised within two to five days via the WU-FTPD hole described in CERT advisory CA-2001-33.

Another benefit of running a honeypot is a better handle on the Internet noise. Clearly, security professionals who run Internet-exposed networks are well aware of the common Internet noise (such as CodeRed and now SQL worms, warez site FTP scans, etc.). A honeypot allows one to observe the minor oscillations of such noise. Sometimes, such changes are meaningful. In the recent case of the MS SQL worm, we detected a sharp increase

in TCP port 1433 access attempts just before the news of the worm became public. The number of hits was similar to a well-researched CodeRed growth pattern. Thus, we concluded that a new worm was out.

An additional value of the honeypot is in its use as a security training platform. Using the honeypot, a company can bring up the level of incident response skills of its security team. Honeypot incidents can be investigated and then the answers verified by the honeypot's enhanced data collection capabilities. What tool was used to attack? Here it is, on the captured hard drive or extracted from network traffic. What did they want? Look at their shell command history and know. One can quickly and effectively develop network and disk forensics skills, attacker tracking, log analysis, IDS tuning, and many other critical security skills in the controlled but realistic environment of the honeypot.

More advanced research uses of the honeypot include hacker profiling and tracking, statistical and anomaly analysis of incoming probes, the capture of worms, and analysis of malicious code development. By adding some valuable resources (such as E-commerce systems and billing databases) and using the covert intelligence techniques to lure attackers in, more sophisticated attackers can be attracted and studied.

Note that these advanced techniques will increase the operating risks.

Conclusion

Trying to answer the question "Should you do it?" concludes the discussion. The precise answer depends on your organization's mission and available security expertise. Again, the emphasis here is on research honeypots and not on "shield" or protection honeypots. If your organization has taken care of most routine security concerns, has a developed in-house security program (calling an outside consultant to investigate your honeypot incident does not qualify as a wise investment), and requires first-hand knowledge of attacker techniques and last-minute Internet threats, the answer tends toward a tentative "yes." Major security vendors and consultancies or universities with advanced computer security programs might fall into this category. If you are not happy with your existing security infrastructure and want to replace or supplement it with the new, cutting-edge "honeypot technology," the answer is a resounding "no." Research honeypots will not *directly* impact the safety of your organization. Moreover, honeypots have their inherent dangers. They are analyzed in chapters posted on the Honeynet Project site. The dangers include uncertain liability status, possible hacker retaliation, and others.

151

CIRT: Responding to Attack

Chris Hare, CISSP, CISA

This chapter presents a number of topics and issues for today's organization when considering the requirements and impact of establishing a computer incident response team (CIRT). This chapter makes no assumptions as to where a CIRT should be positioned from an organizational perspective within an organization, but focuses on why establishing a CIRT is important and what is involved in setting one up.

The term Computer Emergency Response Team, or CERT, is used to identify the government-funded team located at Carnegie Mellon University. The university has trademarked the name CERT (http://www.cert.org). Consequently, incident response teams are known by one of several other names. These include:

- Computer Incident Response Team (CIRT)
- Computer Security Incident Response Team (CSIRT)
- Systems Security Incident Response Team (SSIRT)

Regardless of the nomenclature, the CIRT is typically responsible for the initial evaluation of a computer security incident and providing corrective action recommendations to management. This chapter explores in detail the prerequisites, roles and responsibilities, and supportive processes necessary for a successful CIRT capability.

History

Prior to the Morris Internet worm of 1988, computer security incidents did not really get a lot of attention, as the problem was not well understood. At that time, there was only a fraction of the total network hosts connected today.

The Morris worm demonstrated to the Internet community, and to the computing world in general, that any determined attacker could cause damage, wreak havoc, and paralyze communication systems by using several commonly known vulnerabilities in UNIX system applications.

The nature of the problem is quite severe. An Internet mailing list known as BUGTRAQ discussed security issues and vulnerabilities in applications and operating systems. This mailing list currently has a volume of more than 1000 messages per quarter, most of which are exploits, bugs, or concerns about commercial applications.

Consider that IBM's mature MVS operating system has 17 million lines of assembly language instructions. Microsoft's Windows NT 5 (Windows 2000) has more than 48 million lines of C and assembly language code. The recognized "bug" factor is one bug for each 1000 lines of code. Windows NT 4 had more than 100,000 validated bugs. This means that there is potential for 48,000 bugs in Windows NT 5.

These bugs provide the perfect opportunity for the attacker to gain access to a system, and either steal, modify, or destroy information or resources from the system owner.

Who Is Attacking Who?

The nature of the attacker is changing dramatically. Considering the movies of a few years ago, *The Net* and *Sneakers*, computer hackers were portrayed as well-educated adults who knew their way around computer systems. They understood what information they needed, how to get it, and what they had to do once they gained access to a system.

Attacker profiles vary considerably:

- *The Naïve*: These attackers have little real knowledge or experience. They are out to do it for fun, with no understanding of the potential consequences.
- *Brutish* (*script kiddies*): These attackers also lack little real knowledge, and make heavy use of the various attack tools that exist. This means that they become obvious and visible on attacked systems due to the heavy probing and scanning used.
- *Clueful*: These are more experienced attackers, who use a variety of techniques to gain access to the system. The attacks are generally more subtle and less obvious.
- *Truly Subtle*: These are the computer criminals of the twenty-first century. They know what they want, who will pay for it, how to get access, and how to move around the system once they enter it. These attackers leave few or no traces on a system that they were in fact there.

The Teenage Attacker

The development of more sophisticated tools has lowered the required sophistication level of the attacker. There are reports of attackers who successfully used the tools to gain access to a system, but then did not know what to do once they got in.

Many teenage attackers also make use of the techniques demonstrated by actor Matthew Broderick in the 1980s movie *War Games*. Broderick used a program known as a "war dialer" to locate the modem tones for computer systems. Today's tools provide the naïve or clueful and brutish attackers with the necessary tools to gain access to almost any system. These tools are meant to be stealthy by nature; and although frequently used by the people outside the organization, they are also used from within. Information on common exploits and attacked sites are available at http://www.rootshell.com, among others.

Although these tools do provide an easier method to compromise a system and gain access, attackers must still know what to do on the system once they have gained access. Recent attempts, as reported in *Systems Administration and Network Security (SANS)* (http://www.sans.org) bulletins and briefings, show that some successful attacks result in little damage or information loss because the attacker did not know how to interact with the system.

The Insider

Insiders may or may not have malicious intent. Their authorized presence on the network allows them virtually unrestricted access to anything, and may allow them to access information that they would normally not have the authority to access. This makes the distinction between the fact that employees are authorized to access the network and specific information and applications available. It does not imply that an employee has any implicit or explicit authorization to access all of the information available on the network.

Malicious insiders are insidious individuals whose goal is to steal or manipulate information so that the company does not have access to complete and accurate data. They may simply destroy it, provide it to the competition, or attempt to embarrass the company by leaking it to the media. These people have authorized access to the network, and therefore are difficult to trace and monitor effectively.

Insiders who are experiencing personal difficulties (e.g., as financial problems), are targets for recruitment by competitive intelligence agencies.

Even more important, insiders can make copies of the information and leave the original intact, thereby making it more difficult to detect that a theft took place. Those insiders that do cause damage lead to detection of the event, but those that undertake some planning make detection much more difficult — if not impossible

The Industrial Spy

Probably the most feared are the industrial spies. These attackers specifically target a particular company as a place from which to obtain information that they have been hired to collect, or that they believe will be considered valuable to others who would buy it. This is known as industrial or economic espionage. The difference between the two is that industrial espionage is conducted by organizations on behalf of companies, and economic espionage is data collection that is authorized and driven by governments.

These criminals are likely well-trained and will use any means at their disposal to discover and steal or destroy information, including social engineering, dumpster diving, coordinated network attacks, even getting a job as a contractor. The FBI (http://www.fbi.org) states that a typical organization can expect that one in every 700 employees is actively working against the company.

Nature of the Attack

The attackers have a variety of tools and an increasing number of vulnerabilities in today's software from which to choose. The nature of the attack and the tools used will vary for each of the attacker types and their intent.

Attack Tools

A very extensive — and for the most part easily obtained — set of attack tools is available to today's attacker. They range from C language files that must be compiled and run against a system, to complex scanning and analysis tools such as nmap. A sample nmap run against several different hosts is illustrated in Exhibit 151.1.

The output of the various attack tools can provide the attacker with a wealth of information regarding the system platform, and as such is used by many attackers and system administrators alike. For example, the output illustrated in Exhibit 151.1 identifies the network services that are configured and additional information regarding how easy it would be to launch a particular types of attack against the system. Take special note that it was able to correctly guess the operating system.

Viruses and Mobile Code

A virus is program code that is intended to replicate from system to system and execute a set of instructions that would not normally be executed by the user. The impact of a virus can range from simple replication, to destruction of the information stored on the system, even to destruction of the computer itself.

EXHIBIT 151.1 Sample Output of nmap of a Linux System.
Log of: ./nmap -O -v -v -o /tmp/log2 192.168.0.4
Interesting ports on linux (192.168.0.4)

Port	State	Protocol	Service
21	open	tcp	ftp
23	open	tcp	telnet
25	open	tcp	smtp
37	open	tcp	time
79	open	tcp	finger
80	open	tcp	http
110	open	tcp	pop-3
111	open	tcp	sunrpc
113	open	tcp	auth
139	open	tcp	netbios-ssn

TCP Sequence Prediction: Class = random positive increments
Difficulty = 4686058 (Good luck!)
Remote operating system guess: Linux 2.2.0-pre6 - 2.2.2-ac5

Viruses are quite common on the Windows platform due to the architecture of the processor and the operating system. It is likely that most computer users today have been "hit" by one virus or another. The attacker no longer has to be able to write the World Wide Web (WWW).

Use of the WWW introduces additional threats through "active code" such as Microsoft's ActiveX and Sun Microsystems' Java languages. These active code sources can be used to collect information from a system, or to introduce code to defeat the security of a system, inject a virus, or modify or destroy information.

The First CERT

The first incident response team was established by the Defense Applied Research Projects Agency (DARPA) (http://www.darpa.mil) in 1988 after the Morris worm disabled approximately 10 percent of the computer systems connected to the Internet. This team is called the Computer Emergency Response Team (CERT) and is located at the Software Engineering Institute at Carnegie Mellon University.

Learning from the Morris Worm

The Morris worm of 1988 was written by Robert Morris, Jr. to demonstrate the vulnerabilities that exist in today's software. Although Morris had contended since his arrest that his intent was not to cause the resulting damage, experts who have analyzed the program have reported that the Morris Worm operated as expected.

There were a large number of reports written in the aftermath of the incident. The General Accounting Office (GAO) issued a thorough report of the Morris worm, its impact and the issues surrounding security on the Internet, and the prosecution of this and similar cases in the future.

The GAO report echoes observations made in other reports on the Morris worm. These observations include:

- The lack of a focal point in addressing Internet-wide security issues contributed to problems in coordination and communication during security emergencies.
- Security weaknesses exist in some sites.
- Not all system managers have the skills and knowledge to properly secure their systems.
- The success of the Morris Worm was through its method of attack, where it made use of known bugs, trusted hosts, and password guessing.
- Problems exist in vendor patch and fix development and distribution.

While these issues were discussed after the Morris worm incident, they are, in fact, issues that exist within many organizations today.

Legal Issues

There are many and inconsistent legal issues to be considered in investigating computer crime. It is worth noting, however, that an incident response team (or corporate investigations unit) typically has considerably more leeway in its operations than law enforcement.

As the property being investigated belongs to the company, the company is free to take any action that it deems appropriate. Once law enforcement is notified of the crime, then the situation becomes a law enforcement issue, and the organization's ability to act is significantly curtailed. This is because once law enforcement is informed, the company's investigators become agents for law enforcement and are then bound by the same constraints.

Among the legal issues that must be addressed are the rules of evidence. These vary from country to country due to differences in legal systems. These rules address how evidence must be collected and handled in order for it to be considered evidence by law enforcement agencies and in a court of law.

The exact actions that the CIRT can perform are governed by the appropriate legislation. The team will be advised by Corporate Counsel, at which point appropriate action will be taken with the intent of not jeopardizing the value of collected evidence or interviews.

Threat Analysis

Threat — and risk analysis in general — is a major proactive role of the CIRT. The CIRT must evaluate every vulnerability report and, based on an analysis of the situation, recommend the appropriate actions to management and who is responsible for completing these actions.

Most often, risk analysis focuses on new exploits or attack methods to determine if there are associated risks within the organizational environment and how such risks can best be mitigated. This is part of the CIRTs ongoing activity, and can include a variety of methods, including research and penetration testing. From this collected information, the CIRT can make recommendations on how to mitigate these risks by making changes to our computing or security infrastructures.

There is, however, the notion of "acceptable" risk. Acceptable risk is that risk which the company is knowingly prepared to accept. For example, if the company can earn $1 million but in the process has an exposure that could cause the loss of $10,000, the company may choose to accept such risk.

These decisions, however, cannot be made by just anyone in the organization. The exact nature of the vulnerability, the threat, and the resulting impact must be clearly evaluated and understood.

- *Threat* is defined as the potential to cause harm to the corporation — intentional or otherwise. Threats include hackers, industrial espionage, and at times, internal employees.
- *Vulnerability* is a weakness or threat to the asset. If there are no vulnerabilities, then a threat cannot put the organization at risk.
- *Impact* reflects degree of harm and is concerned with how significant the problem is, or how much effect it will have on the company.

The threat graph in Exhibit 151.2 illustrates threat, impact, and vulnerability. The risk is lowest when threat and impact are both low. Low impact, low threat, and low vulnerability imply that the *risk* is also low.

If the threat is low, the impact is high, and the vulnerability is low, the company may accept the risk of information loss. The same is true if the impact is low, the vulnerability low, and the threat high. This may still be an acceptable risk to the organization.

Finally, as the impact, vulnerability, and threat all increase, the issue becomes one of high risk. This is typically the area that most companies choose to address and place their emphasis. This is where the greatest risk is and, consequently, where the greatest return on security investment is found.

CIRT: Roles and Responses

Most people think of "Incident Response Teams" as the emergency response unit for computers. The confusing term is "computer." A security incident that involves a computer is only different from a physical security incident in how the event took place. When an unauthorized person gains tactical access to a system or specific information, it should have equivalent importance to unauthorized physical access.

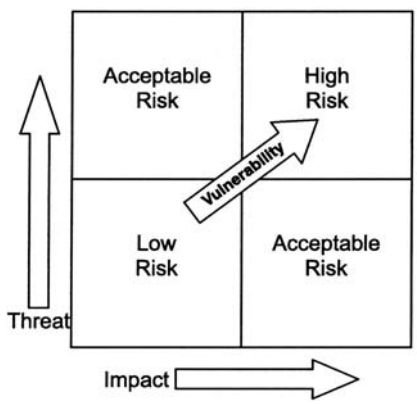


EXHIBIT 151.2 Threat graph: threat, impact, and vulnerability.

The CIRT must be able to handle a crisis and prevent it from becoming worse than it already is. The CIRT, however, has much more to offer, including a proactive role of vulnerability testing, vulnerability analysis, and awareness.

Obviously, the exact nature of responsibilities that one assigns to a CIRT will depend on the size and nature of the organization, the number of incidents recorded, and how many systems and networks exist. Consequently, some of the suggested activities may not be possible for a CIRT to integrate into its day-to-day tasks.

Incident Response

As mentioned, incident response is the prime reason behind establishing a CIRT. This incident response team puts highly trained people at the forefront of any incident, and allows for a consistently applied approach to resolving the incident. The team handles the investigation from start to finish and makes recommendations to management regarding its findings.

Vulnerability Testing

There are two elements to vulnerability testing. The first is to use automated tools with preconfigured tests to determine if there are vulnerabilities that could be exploited by an attacker. The second element test security implementation is to try it out. A penetration or protection test simulates the various types of attacks — internal and external, blind and informed — against the countermeasures of the network. Essentially, a penetration test attempts to gain access through available vulnerabilities by taking on the mindset of the perpetrator.

As the CIRT is responsible for investigating incidents, over time it will develop a set of skills that can be used to offer penetration or protection testing services to the organization's product developers or IS organization. Vulnerability testing is considered the cornerstone of the effort to improve a security program as it attempts to use vulnerabilities as an attacker would. Protection testing is conducted in a similar manner, but the goal is different.

Types of Penetration Tests

There are essentially three major types of penetration testing, each with its own tools and techniques:

> *Level 1. Zero-Knowledge*: This attempts to penetrate the network from an external source without knowledge of its architecture. However, information that is obtained through publicly accessible information is not excluded.
> *Level 2. Full-Knowledge*: This attempts to penetrate the network from an external source with full knowledge of the network architecture and software levels.
> *Level 3. Internal*: This attempts to compromise network security and hosts from inside one's network.

Penetration testing is interval based, meaning that it is done from time to time and against different target points. Penetration testing is not a real-time activity.

The process consists of collecting information about the network and executing the test. In a level 1 test, the only information available is what is published through open source information. This includes network broadcasts, upstream Internet service providers, domain name servers, and public registration records. This helps simulate an attack from an unsophisticated intruder who may try various standard approaches. This approach primarily tests one's ability to detect and respond to an attack.

A Level 2 penetration test assumes full knowledge of the hardware and software used on the network. Such information may be available to meticulous and determined intruders using whatever means, including social engineering, to increase their understanding of one's networks. This stage of the test assumes the worst-possible scenario, and calls to light the maximum number of vulnerabilities.

A Level 3 penetration test (or acid test) is an attack from within the network. This is the best judge of the quality of the implementation of the company's security policy. A real attack from within a network can come from various sources, including disgruntled employees, accidental attacks, and brazen intruders who can socially engineer their way physically into a company.

Penetration testing should be considered very carefully in the implementation of an overall detection program, but it can lead to the negative side effects that one is trying to prevent. Therefore, it should be used

cautiously, but still be used to attempt to locate vulnerabilities and to assess the overall operation of the protection program.

Studying Security Vulnerabilities

When an incident occurs, it is essential to understand what allowed it to happen. Examining the vulnerability used during the incident allows the organization to improve its Security Infrastructure Program to prevent further exploitation.

In addition, security vulnerabilities that are released to the security community need to be assessed for their impact within the organization, and a course of action recommended. The CIRT, with its enhanced skills and knowledge, is capable of reviewing those vulnerabilities and offering the operating system and product groups a method of addressing them.

Publishing Security Alerts

When new issues are found that impact the organization, the CIRT is responsible for the publication of those bulletins and warnings, along with a set of instructions or recommendations regarding how users and systems administrators should react.

Publishing security alerts within the corporation, or new vulnerabilities found, does not include publishing the details of security incidents. The reporting of security incidents is a role for Corporate Security.

Security and Survivability in Wide Area Network-Based Computing

Working from the analysis of incident data, the CIRT is able to make specific recommendations to the systems administrators or applications owners on how to better configure their systems to increase the level of security.

Survivability comes from the application of good administration and consistently applied security techniques to reduce the threat of loss of data from an incident, or the loss of the system. Having to completely rebuild a system is an onerous task that is costly to the business, and one that few people want to repeat frequently.

Defining Incidents

An obvious question is, "What is an incident?" Incidents cannot be easily identified without the team. However, an incident can be defined as any unexpected action that has an immediate or potential effect on the organization.

Example incidents include:

- Viruses
- Unauthorized access, regardless of source
- Information theft or loss of confidentiality
- Attacks against systems
- Denial of service
- Information corruption

However, incidents can be further classified based on the extent to which the incident affects the organization.

The classification of CIRT responses is often based on several factors, including geography, business impact, and the apparent nature of the problem. Business impact includes how many people are affected; how many sites are affected, and will the issue affect stock prices, investor confidence, or damage the organization's reputation.

These classifications are meant to be a guide for discussion purposes — the CIRT may choose to broaden or identify improved characteristics for each.

> *Class 1: Global.* These incidents have the greatest impact on an organization. They have the potential of affecting the entire organization, and they are serious. The uncorrected distribution of a virus can have very significant effects on the organization's ability to function. Other examples include a firewall breach, potential financial loss, customer services, compromise of the corporation's credibility, or the

compromise of the organization's external Web site. In these situations, the CIRT is activated immediately, due to the threat to the company.

Class 2: Regional. Regional incidents affect specific areas of the company. They do, however, have the capability of becoming global. Regional threats include logic bombs, and attacks against specific systems in that region. Although these can become global in nature, the information systems and security organizations in that region may be able to handle the issue without involvement from the CIRT. In this situation, the CIRT is activated at the request of the region IS or Security Directors.

Class 3: Local. Local incidents are isolated to a specific department and are of low impact. Examples include a virus on a single system, and the building cleaning crew playing solitaire on improperly configured desktop systems. In this situation, the CIRT is not activated unless requested by the department manager.

When Does the CIRT Respond?

The CIRT responds in one of several situations:

- At the request of a manager when an event is noticed or reported to them
- When the incident requires it, based on sufficient evidence, probability, or due to a pattern of occurrence
- As the result of issues found during vulnerability testing
- On the advice of the help desk personnel who receive problem reports
- On the advice of an external security agency

CIRT response is based on the severity of an issue, as outlined previously. Managers can request CIRT involvement when they suspect unauthorized activity, regardless of whether there has been an incident reported to the CIRT.

If an incident is believed to have occurred based on evidence (e.g., missing or altered information in a database) or due to alerts from an intrusion detection system, the CIRT is involved to determine the significance, scope, and method of the attack.

It is important to note that help desks can assist in reporting incidents to the CIRT. As employees call their help desks with issues, the help desk may see a pattern emerge that will initiate contacting the CIRT. Consequently, additional training is required for the help-desk staff to inform them of what they should be looking for.

The CIRT then provides a recommendation on how to address the attack and proceed with the investigation of the incident. In some situations, external agencies such as security departments of other organizations may advise of a potential incident and this must be investigated.

Relationship to External Agencies

The CIRT operates within the organizational framework and reviews incidents and provides other services as discussed. It is important, however, that the CIRT establish a relationship with external Computer Emergency Response Teams, such as CERT, CANCERT, etc. These teams provide similar services, but focus on incident reporting and advisory capabilities.

In addition, contact with law enforcement and other external teams that may be required must be established early on, so that if an issue arises, the CIRT is not spending valuable time looking for the correct external resource and then contacting them.

CIRT: The CIRT Process

There is a defined process for creating and establishing the CIRT function. This process is presented in this section. The process consists of six steps. These steps are explained here, but more information on some of the process steps is discussed in other sections.

CIRT is a global process. The team must be available 24 hours a day, 365 days per year. As such, mechanisms to contact the CIRT regardless of where the incident is, must be put into place to allow quick response.

Establishing the Process Owner

The process owner is responsible for supporting the team, and is the individual to whom the team itself reports. The process owner provides the interface to executive management and ensures that the CIRT is fulfilling its responsibilities effectively.

The process owner is assigned by senior management — not by the reputation or position of a single individual. Many organizations choose the Chief Information Officer (CIO) as the process owner, due to the technical nature of the team. Although this is not necessarily incorrect, it is now considered more appropriate to choose either the CFO or the Internal Audit Director to avoid any possibility of conflict of interest. The two alternate positions have legally defined fiduciary responsibilities to protect the corporation's assets and their departments often include staff with fraud investigation backgrounds.

Establishing the Team

The development of a CIRT is a process that requires full acceptance from the corporation's executives, and the groups involved in forming the core team. Specific resources, funding, and authority must be granted for the initiative to be successful and have benefit to the corporation. This section discusses the structure of the CIRT and how it interacts with other internal organizations.

Many organizations consider computer security incidents as an IS problem, although in fact they are a business problem although because any security incident, regardless of how it is caused, has the potential to affect the corporation in many ways, including financial loss, legal or financial liabilities, or customer service.

The very nature of computer involvement means that what is deemed to be an incident may not be when investigated. For example, consider the user who forgets his password and disables it. This may appear like a denial-of-service attack, when in fact it is not. This strains the internal investigative resources, and impacts the company by redirecting resources where they are not needed.

The investigation of an event is a complex process that involves a precise sequence of events and processes to ensure that, should the corporation choose to, it could involve law enforcement and not lose access to the valuable information, or evidence, already collected.

To do this, and for the response to any incident to be effective, people with a wide range of backgrounds and experiences are required. The CIRT ideally would have people from the following areas:

- Technical specialists: An understanding of the production aspects of the technology that are relevant to the investigation
- Information security specialists: Data and systems protection
- Auditors and fraud examiners: Compliance and fraud
- Corporate security: Investigations
- Human resources: Personnel and labor issues
- Business continuity specialists: System and data recovery
- Legal specialists: Protecting the organization's intellectual property
- Corporate public relations: Press and media interaction
- Executive management: The decision makers
- Any other organization- or industry-specific personnel, such as business unit or geographically relevant personnel

The Core Team

For most organizations, it is difficult to rationalize the dedication of such a group of people to the CIRT role and, consequently, it is seen within the industry that the CIRT has two major components: a core team and a support team. The core team is composed of five disciplines, preferably staffed by a single individual from each discipline. These disciplines are:

- Corporate security
- Internal audit
- Information protection
- Legal specialists.
- Technical specialists, as required

The CIRT core team must:

- Determine if the incident is a violation
- Determine the cause and advise management on the action required
- If required, establish the appropriately skilled support team
- Manage the investigation and report
- All in external agencies as necessary

It is essential that the core team be made up of individuals who have the experience required to determine the nature of the incident and involve the appropriate assistance when required.

Many larger organizations have a corporate security group that provides the investigators who are generally prime for the incident. Smaller organizations may have a need to address their investigative needs with a security generalist. This is because the ultimate recommendation for the CIRT may be to turn the incident over to the corporate security organization for further investigation or to contact law enforcement. Obviously, the correct course of action depends on organization structure, and whether or not to contact law enforcement. In that event, specific rules must have been followed. These rules, although important, are not germane to the discussion here.

Internal Audit Services provide the compliance component. Every organization is required to demonstrate compliance with its policies and general business practices. The internal audit organization brings the compliance component to the team; moreover, it will be able to recommend specific actions that are to be taken to prevent further incidents.

The Information Protection or Information Services security specialist is required because the incident involved the use of a computer. The skills that this person holds will enable rapid determination of the path of the attack from one place to another, or gain rapid access to the information contained on a system.

Legal Specialists are essential to make sure that any actions taken by the CIRT are not in violation of any existing corporate procedures, ofany rights of any individuals within the company or country. This is especially important, as there are different laws and regulations governing the corporation and the rights of the individual in many countries.

Although team members have these backgrounds in their respective areas, the core team operates in one of two ways:

- Dedicated full-time to the role of the CIRT and its additional responsibilities identified previously
- Called as needed to examine the incident

In large, geographically dispersed organizations, the CIRT must be capable of deploying quickly and getting the information such as logs, files, buffers, etc., while it is still "fresh," There is no "smoking gun" — only the remnants left behind. Quick action on the part of the CIRT may enable collection of incident-related information that would otherwise be rendered useless as evidence minutes or hours or later.

Selecting the Core Team Members

The selection of the core team members is done based on experience within their knowledge area, their ability to work both individually and as part of a team, and their knowledge of the company as a whole. The process owner, who will select a team leader and then work together to choose the other members of the core team, would conduct the selection process. It is recommended that the team leader be a cooperating member of the team, and that the team leader operate as the point of contact for any requests for assistance.

The Support Team

The support team is used to provide additional resources once the core team has determined what the incident really is, and what other experts need to be called in to assess the situation.

The support team is vital to the operational support of the core team. This is because it is impossible for the core team to have all of the knowledge and expertise to handle every possible scenario and situation. For the core team to be effective, it must identify who the support team members are and maintain contact with and backup information for them over time.

The Support team consists of:

- Human resources (HR)
- Corporate communications
- Platform and technology specialists

- Fraud specialist
- Others as required, such as business unit specialists or those geographically close to the incident

Human resources (HR) is a requirement because any issue that is caused by an employee will require HR's involvement up front to assist in the collection of relevant information, and discussion of the situation with the employee's manager and the employee, and recommendations of appropriate sanctions.

If the incident is a major one that might gain public attention, it is recommended that the corporate public relations function issue a press release earlier, rather than take "knocks" from the public press. Although any bad news can affect a company, by releasing such information on its own, the company can retain control of the incident and report on planned actions. However, it is essential that any press announcements must be cleared through the appropriate departments within the company, including the legal department and senior management. However, there have been sufficient examples with companies (like Microsoft) that would argue this point both ways.

Additionally, the team must designate an individual who is not actively participating to provide information and feedback to management and employees, as deemed appropriate. By choosing a person who does not have an active part in that particular investigation, that person can focus on the communications aspect and let the rest of the team get the job done.

The platform and technology specialists are used to provide support to the team, as no single individual can be aware of and handle all of the technology-related issues in the company. It is also likely that multiple technical specialists will be required, depending on the nature of the incident.

Fraud specialists provide guidance on the direction and investigation of fraud. In some cases, fraud will be hidden behind other issues to cloud the fraud and throw confusion on the issue.

The core team does the selection of the support team members. The core team must evaluate what types of skills it must have access to and then engage the various units within the organization to locate those skills.

It is essential that the core team conduct this activity to allow establishment of a network of contacts should the identified support team member and his or her backup be unavailable. Support team members are selected based on experience within their knowledge area, their ability to work both individually and as part of a team, and their knowledge of the company as a whole.

A major responsibility of the core team is to maintain this database of support team members to allow for quick response by the team when its involvement is required.

Creating the CIRT Operation Process

With the structure of the actual team in mind, it becomes necessary to focus on how the CIRT will operate. This is something that cannot be easily established in advance of core team selection. The process defines the exact steps that are followed each time the team is activated, either by request or due to the nature of the incident.

Aside from some steps that are required to create, establish, and authorize the team, the remaining steps in the process are to be handled by the core team. In addition to training and various other roles, the team must also:

- Document its own practices and procedures
- Establish and maintain databases of contact names and information
- Maintain software and hardware tools required and used during an incident

Several matrices must be developed by the newly formed CIRT. These include an incident matrix and a response matrix. In the incident matrix, the team attempts to discover every possible scenario, and establish the:

- Incident type
- Personnel required
- Financial resources required
- Source of resources

With this, the CIRT can establish the broad budget it will need to investigate incidents. The response matrix identifies the incident type, what the team feels is an appropriate response to the incident, what resources it anticipates will be needed, and how it will escalate the incident should that become necessary. Neither of these matrices can be developed without the core team, and even some initial members of the support teams.

With the matrices completed, it is necessary to establish the training and funding requirements for the team.

Training Requirements

With the CIRT formed, it is necessary that the training requirements be determined. At a minimum, all members of the core team will need to be trained in intrusion management techniques, investigations, interviewing, and some level of computer forensics. (There are organizations that can conduct training specifically in these areas.)

Funding Requirements

The CIRT must now establish its requirements for a budget to purchase the needed equipment that will be used on a frequent or daily basis. A contingency budget is also needed to establish spending limits on equipment that is needed in the middle of an incident.

Given the nature and size of the core team, it is easy to establish that personnel budgets within a large organization will include a minimum of $500K for salaries and other employee costs. Training will approximate $50K per year, with an initial training expense of approximately $100K.

Policy and Procedures

The operation of the CIRT must be supported through policy. The policy establishes the reasons for establishing a CIRT, its authority, and the limits on its actions. Aside from the issues regarding policy in general, policies that support a CIRT must:

- Not violate the law: Doing so results in problems should the need for law enforcement result, or if the employee challenges the actions taken by the company as a result
- Address privacy: Employees must be informed in advance that they have no reasonable expectation of privacy (management has the right to search e-mail, stored files and their on-site workstations during an investigation)
- Have corporate counsel review and approve the policy and procedures as being legal and sustainable in the given local areas

The policy itself leaves out the specifics surrounding the CIRT and how it operates. These are written in standards and procedures and describe how the team will react in specific situations, who the team members are, what the organization structure is, etc.

As mentioned, the employee must not have any expectation of privacy. This can only be accomplished effectively by understanding the privacy laws in the different regions, and stating specifically in policy, that this is the case.

CIRT members should operate within a code of ethics specifically designed for them, as they will be in contact and learn information about employees or situations that they would otherwise not know.

Funding

Funding is essential to the operation of the CIRT. Although it is impossible to know what every investigation will cost, the team will have established a series of matrices identifying possible incidents and the equipment and resources required to handle them. This information is required to establish an operating budget, but contingency funds must be available should an incident cause the team to run over budget, or need a resource that was not planned.

Obviously, not having this information up front affects senior management's decisions to allocate base funding. This means, however, that senior management must believe in the role of the CIRT and the value that it brings to the overall security posture. The CIRT process owner in consultation with the identified CIRT members and external CIRTs, should be able to establish a broad level of required funding and modify it once the matrices are completed.

Authority

The CIRT must be granted the authority to act by senior management. This means that during an investigation of an incident, employees — regardless of level in the company — must be directed to cooperate with the CIRT. They must operate with extreme attention to confidentiality of the information they collect. The CIRT's responsibility is to collect evidence and make recommendations — not to determine guilt.

The role of the CIRT, as previously mentioned, is to investigate incidents and recommend appropriate actions to be taken by management to deal appropriately with the issues. The authority for the creation of the CIRT and its ability to get the job done is conveyed through policy.

Summary

This author has previously discussed intrusion detection.[1] Intrusion detection, regardless of the complexity and accuracy of the system, is not effective without an incident response capability. Consequently, any organization — regardless of size — must bear this in mind when deciding to go ahead with intrusion detection.

But incident response goes well beyond. Incident response is a proactive response to an incident. However, the CIRT can assist in the prevention and detection phases of the security cycle, and thereby create a much stronger, more resilient, and more responsive security infrastructure for today's organization.

References

1. Farrow, Rik, *Intrusion Techniques and Countermeasures*, Computer Security Institute: San Francisco, 1999
2. Icove, David, Seger, Karl, and VonStorch, William, *Computer Crime: A Crime Fighter's Handbook*, O'Reilly & Associates: Sebastopol, CA, 1995.
3. Stephenson, Peter, *How to Form a Skilled Computer Incident Response Team*, Computer Security Institute: San Francisco, 1999.
4. CERT, *Responding to Intrusions*, Carnegie Mellon Software Engineering Institute, 1998.
5. Winkler, Ira, *Corporate Espionage*, Prima Publishing: Rocklin, California, 1997.
6. Sterneckert, Alan B., *Critical Incident Management*, Auerbach Publications, Boca Raton, FL, 2004.

153

Managing the Response to a Computer Security Incident

Michael Vangelos, CISSP

Organizations typically devote substantial information security resources to the prevention of attacks on computer systems. Strong authentication is used, with passphrases that change regularly, tokens, digital certificates, and biometrics. Information owners spend time assessing risk. Network components are kept in access-controlled areas. The least privilege model is used as a basis for access control. There are layers of software protecting against malicious code. Operating systems are hardened, unneeded services are disabled, and privileged accounts are kept to a minimum. Some systems undergo regular audits, vulnerability assessments, and penetration testing. Add it all up, and these activities represent a significant investment of time and money.

Management makes this investment despite full awareness that, in the real world, it is impossible to prevent the success of all attacks on computer systems. At some point in time, nearly every organization must respond to a serious computer security incident. Consequently, a well-written computer incident response plan is an extremely important piece of the information security management toolbox. Much like disaster recovery, an incident response plan is something to be fully developed and practiced — although one hopes that it will never be put into action.

Management might believe that recovering from a security incident is a straightforward exercise that is part of an experienced system administrator's job. From a system administrator's perspective, that may be true in many instances. However, any incident may require expertise in a number of different areas and may require decisions to be made quickly based on factors unique to that incident. This chapter discusses the nature of security incidents, describes how to assemble an incident response team (IRT), and explains the six phases of a comprehensive response to a serious computer security incident.

Getting Started

Why Have an Incident Response Plan?

All computer systems are vulnerable to attack. Attacks by internal users, attacks by outsiders, low-level probes, direct attacks on high-privilege accounts, and virus attacks are only some of the possibilities. Some attacks are merely annoying. Some can be automatically rejected by defenses built into a system. Others are more serious and require immediate attention. In this chapter, incident response refers to handling of the latter group of attacks and is the vehicle for dealing with a situation that is a direct threat to an information system.

Some of the benefits of developing an incident response plan are:

- *Following a predefined plan of action can minimize damage to a network.* Discovery that a system has been compromised can easily result in a state of confusion, where people do not know what to do.

Technical staff may scurry around gathering evidence, unsure of whether they should disable services or disconnect servers from the network. Another potential scenario is that system administrators become aggressive, believing their job is to "get the hacker," regardless of the effect their actions may have on the network's users. Neither of these scenarios is desirable. Better results can be attained through the use of a plan that guides the actions of management as well as technicians during the life of an incident. Without a plan, system administrators may spend precious time figuring out what logs are available, how to identify the device associated with a specific IP address, or perform other basic tasks. With a plan, indecision can be minimized and staff can act confidently as they respond to the incident.

- *Policy decisions can be made in advance.* An organization can make important policy decisions before they are needed, rather than in the heat of the moment during an actual incident. For example, how will decisions be made on whether gateways or servers will be taken down or users disconnected from the network? Will technicians be empowered to act on their own, or must management make those decisions? If management makes those decisions, what level of management? Who decides whether and when law enforcement is notified? If a system administrator finds an intruder with administrative access on a key server, should all user sessions be shut down immediately and log-ins prohibited? If major services are disrupted by an incident, how are they prioritized so that technicians understand the order in which they should be recovered? Invariably, these and other policy issues are best resolved well in advance of when they are needed.
- *Details likely to be overlooked can be documented in the plan.* Often, a seemingly unimportant event turns into a serious incident. A security administrator might notice something unusual and make a note of it. Over the next few days, other events might be observed. At some point, it might become clear that these events were related and constitute a potential intrusion. Unless the organization has an incident response plan, it would be easy for technical staff to treat the situation as simply another investigation into unusual activity. Some things may be overlooked, such as notifying internal audit, starting an official log of events pertaining to the incident, and ensuring that normal cleanup or routine activities do not destroy potential evidence. An incident response plan will provide a blueprint for action during an incident, minimizing the chance that important activities will fall through the cracks.
- *Nontechnical business areas must also prepare for an incident.* Creation of an incident response plan and the act of performing walk-throughs or simulation exercises can prepare business functions for incident response situations. Business functions are typically not accustomed to dealing with computer issues and may be uncomfortable providing input or making decisions if "thrown into the fire" during an actual incident. For example, attorneys can be much better prepared to make legal decisions if they have some familiarity with the incident response process. Human resources and public relations may also be key players in an incident and will be better able to protect the organization after gaining an understanding of how they fit into the overall incident response plan.
- *A plan can communicate the potential consequences of an incident to senior management.* It is no secret that, over time, companies are becoming increasingly dependent on their networks for all aspects of business. The movement toward the ability to access all information from any place at any time is continuing. Senior executives may not have an appreciation for the extent to which automation systems are interconnected and the potential impact of a security breach on information assets. Information security management can use periodic exercises in which potential dollar losses and disruption of services in real-life situations are documented to articulate the gravity of a serious computer security incident.

Requirements for Successful Response to an Incident

There are some key characteristics of effective response to a computer security incident. They follow from effective preparation and the development of a plan that fits into an organization's structure and environment. Key elements of a good incident response plan are:

- *Senior management support.* Without it, every other project and task will drain resources necessary to develop and maintain a good plan.
- *A clear protocol for invoking the plan.* Everyone involved should understand where the authority lies to distinguish between a problem (e.g., a handful of workstations have been infected with a virus because users disabled anti-virus software) and an incident (e.g., a worm is being propagated to hundreds of

workstations and an anti-virus signature does not exist for it). A threshold should be established as a guide for deciding when to mobilize the resources called for by the incident response plan.

- *Participation of all the right players.* Legal, audit, information security, information technology, human resources, protection (physical security), public relations, and internal communications should all be part of the plan. Legal, HR, and protection may play an important role, depending on the type of incident. For some organizations, public relations may be the most important function of all, ensuring that consistent messages are communicated to the outside world.
- *Clear establishment of one person to be the leader.* All activity related to the incident must be coordinated by one individual, typically from IT or information security. This person must have a thorough knowledge of the incident response plan, be technical enough to understand the nature of the incident and its impact, and have the ability to communicate to senior management as well as technical staff.
- *Attention to communication in all phases.* Depending on the nature of the incident, messages to users, customers, shareholders, senior management, law enforcement, and the press may be necessary. Bad incidents can easily become worse because employees are not kept informed and cautioned to refer all outside inquiries concerning the incident to public relations.
- *Periodic testing and updates.* The incident response plan should be revisited regularly. Many organizations test disaster recovery plans annually or more frequently. These tests identify existing weaknesses in the plan and uncover changes in the automation environment that require corresponding adjustments for disaster recovery. They also help participants become familiar with the plan. The same benefits will be derived from simulation exercises or structured walk-throughs of an incident response plan.

Defining an Incident

There is no single, universally accepted definition of incident. The Computer Emergency Response Team Coordination Center (CERT/CC) at Carnegie Mellon University defines incident as "the act of violating an explicit or implied security policy."[1] That may be a great way to describe all events that are bad for computer systems, but it is too broad to use as a basis for the implementation of an incident response plan. The installation of a packet sniffer without management authorization, for instance, may be a violation of policy but probably would not warrant the formality of invoking an incident response plan. However, the use of that sniffer to capture sensitive data such as passwords may be an incident for which the plan should be invoked. The Department of Energy's Computer Incident Advisory Capability (CIAC) uses this definition for *incident*:

> Any adverse event that threatens the security of information resources. Adverse events may include compromises of integrity, denial-of-service attacks, compromise of confidentiality, loss of accountability, or damage to any part of the system. Examples include the insertion of malicious code (e.g., viruses, Trojan horses, or backdoors), unauthorized scans or probes, successful and unsuccessful intrusions, and insider attacks.[2]

This, too, is a good definition and one that is better aligned with the goal of identifying events that should trigger implementation of an incident response plan. To make this definition more useful in the plan, it should be complemented by guidelines for assessing the potential severity of an incident and a threshold describing the level of severity that should trigger invocation of the plan. Responding to an incident, as described in this chapter, involves focused, intense activity by multiple people in order to address a serious condition that may materially affect the health of an organization's information assets. Therefore, as the incident response plan is developed, an organization should establish criteria for deciding whether to invoke the plan.

Developing an Incident Response Team

There is no singularly correct makeup of an incident response team (IRT). However, it is generally agreed that if the following functional units exist in an organization, they should be represented: information security, information technology, audit, legal, public relations, protection (physical security), and human resources. In an ideal situation, specific individuals (preferably a primary and secondary contact) from each of these areas are assigned to the IRT. They will be generally familiar with the incident response plan and have an understanding of what kinds of assistance they may be called upon to provide for any incident. Exhibit 153.1 lists the participants and their respective roles.

EXHIBIT 153.1 Incident Response Team roles

Function	Probable Role
Information security	Often has responsibility for the plan and leads the response; probably leads the effort to put preventive controls in place during preparation phase; staff may also be involved in the technical response (reviewing logs, cleaning virus-infected workstations, reviewing user definitions and access rights, etc.)
Information technology	Performs most eradication and recovery activities; probably involved during detection phase; should be active during preparation phase
Audit	Independent observer who reports to highest level of the organization; can provide valuable input for improving incident response capability
Legal	May be a key participant if the incident was originated by an employee or agency hired by the victim organization; can also advise in situations where downstream liability may exist (e.g., there is evidence that a system was compromised and subsequently used to attack another company's network); may want to be involved any time a decision is made to contact law enforcement agencies; should have input to decisions on whether to prosecute criminal activity; would advise on any privacy issues
Public relations	Should coordinate all communication with the outside world; probably creates the messages that are used
Protection	May be necessary if the incident originated from within the organization and the response may involve confronting a potentially hostile employee or contractor; might also be the best entity to take custody of physical evidence
Human resources	Provides input on how to deal with a situation in which an employee caused the incident or is actively hacking the system

Some organizations successfully manage incidents by effectively splitting an IRT into two distinct units. A technical team is made up of staff with responsibility for checking logs and other evidence, determining what damage if any has been done, taking steps to minimize damage if the incident is ongoing, and restoring systems to an appropriate state. A management team consists of representatives of the functional areas listed above and would act as a steering committee and decision-making body for the life of the incident. An individual leading the response to an incident would appoint leaders of each team or serve as chair of the management team. The two teams, of course, should be in frequent communication with each other, generally with the management team making decisions based on input from the technical team.

Six Phases of Incident Response

It is generally accepted that there are six phases to the discipline of incident response, and the cycle begins well before an incident ever occurs. In any one incident, some of these phases will overlap. In particular, eradication and recovery often occur concurrently. The phases are:

- Preparation
- Detection
- Containment
- Eradication
- Recovery
- Follow-up

Exhibit 153.2 briefly describes the goal of each phase.

Preparation Phase

If any one phase is more important than the others, it is the preparation phase. Before an incident occurs is the best time to secure the commitment of management at all levels to the development of an effective incident

EXHIBIT 153.2 Goal of Each Incident Response Phase

Phase	Goal
Preparation	Adopt policies and procedures that enable effective incident response
Detection	Detect that an incident has occurred and make a preliminary assessment of its magnitude
Containment	Keep the incident from spreading
Eradication	Eliminate all effects of the incident
Recovery	Return the network to a production-ready status
Follow-up	Review the incident and improve incident-handling capabilities

response capability. This is the time when a solid foundation for incident response is built. During this phase, an organization deploys preventive and detective controls and develops an incident response capability.

Management responsible for incident response should do the following:

- Name specific individuals (and alternates) as members of the IRT. Each functional area described in the preceding section of this chapter (audit, legal, human resources, public relations, information security, information technology) should be represented by people with appropriate decision-making and problem-solving skills and authority.
- Ensure that there is an effective mechanism in place for contacting team members. Organizations have a similar need for contacting specific people in a disaster recovery scenario. It may be possible to use the same process for incident response.
- Include guidelines for deciding when the incident response plan is invoked. One of the key areas of policy to be considered prior to an incident is answering the question, "What are the criteria for declaring an incident?"
- Specify the relative priority of goals during an incident. For example,
  - Protect human life and safety (this should always be first).
  - Protect classified systems and data.
  - Ensure the integrity of key operating systems and network components.
  - Protect critical data.
- Commit to conducting sessions to exercise the plan, simulating different types of incidents. Exercises should be as realistic as possible without actually staging an incident. An exercise may, for example, prompt legal, human resources, and protection to walk through their roles in a situation where an employee and contractor have conspired to compromise a network and are actively hacking the system while on company premises. Exercises should challenge IT and information security staff to identify the logs and other forensic data or tools that would be used to investigate specific types of incidents.
- Decide on the philosophy to be used in response to an intrusion. Should an attacker successfully hack in, does the victim organization want to get rid of the intruder as quickly as possible and get back to business (protect and proceed)? Or does the organization want to observe the intruder's movements and potentially gather data for prosecution (pursue and prosecute)?
- Ensure that there is a reasonable expectation that the skills necessary to perform the technical tasks of the incident response plan are present in the organization. Enough staff should understand the applicable network components, forensic tools, and the overall plan so that when an incident occurs, it can be investigated in a full and competent manner.
- Make adjustments to the plan based on test scenario exercises and reviews of the organization's response to actual incidents.
- Review the organization's security practices to ensure that intrusion detection systems are functional, logs are activated, sufficient backups are taken, and a program is in place for regularly identifying system vulnerabilities and addressing those vulnerabilities.

Detection Phase

The goal of the detection phase is to determine whether an incident has occurred. There are many symptoms of a security incident. Some common symptoms are:

- New user accounts not created by authorized administrators
- Unusual activity by an account, such as an unexpected log-in while the user is known to be on vacation or use of the account during odd hours
- Unexpected changes in the lengths of timestamps of operating system files
- Unusually high network or server activity or poor system performance
- Probing activity such as port scans
- For Windows operating systems, unexplained changes in registry settings
- Multiple attempts to log in as root or administrator

Various tools are available to help detect activity that could indicate a security incident. First, there are system logs. Systems should be configured so that logs capture events such as successful and failed log-ins of administrator-level accounts. In addition, failed log-ins of all accounts should be logged. Because log data is relatively worthless unless someone analyzes it, logs should be reviewed on a regular basis. For many systems, the amount of data captured in logs is so great that it is impossible to review it without a utility that searches for and reports those records that might be of interest.

Data integrity checkers exist for UNIX and Windows platforms. These utilities typically keep a database of hash values for specified files, directories, and registry entries. Any time an integrity check is performed, the hash value for each object is computed and compared to its corresponding value in the database. Any discrepancy indicates that the object has changed since the previous integrity check. Integrity checkers can be good indications of an intrusion, but it can take a great deal of effort to configure the software to check only those objects that do not change due to normal system activity.

Intrusion detection systems (IDSs) claim to identify attacks on a network or host in real-time. IDSs basically come in two flavors — network based and host based. A network-based IDS examines traffic as it passes through the IDS sensor, comparing sequences of packets to a database of attack signatures. If it finds a match, the IDS reports an event, usually to a console. The IDS may also be able to send an e-mail or dial a pager as it detects specific events. In contrast, a host-based IDS examines log data from a specific host. As the system runs, the IDS looks at information written to logs in real-time and reports events based on policies set within the IDS.

Organizations become aware of security incidents in many ways. In one scenario, technical staff probably notices or is made aware of an unusual event and begins to investigate. After some initial analysis, it is determined that the event is a threat to the network, so the incident response plan is invoked. If so, the IRT is brought together and formal logging of all activity related to this incident begins. It should be noted that early detection of an incident could mean a huge difference in the amount of damage and cost to the organization. In particular, this is true of malicious code attacks as well as intrusions.

In this phase, the IRT is formally called into action. It is important that certain things occur at this time. Perhaps most importantly, one person should take charge of the process. A log of all applicable events should be initiated at this time and updated throughout the incident. Everyone involved in responding to the incident must be aware of the process. They should all be reminded that the incident will be handled in accordance with guidance provided by the plan, that technical staff should communicate all new developments as quickly as possible to the rest of the team, that everyone must remember to observe evidence chain-of-custody guidelines, and that all communication to employees as well as the outside world should flow through official channels. Some organizations will specify certain individuals who should always be notified when the incident response plan is invoked, even if they are not members of the IRT. For example, the highest internal audit official, COO, the highest information security official, or, in the case where each division of an organization has its own incident response capability, corporate information security may be notified.

Containment Phase

The goal of the containment phase is to keep the incident from spreading. At this time, actions are taken to limit the damage. If it is a malicious code incident, infected servers and workstations may be disconnected from the network. If there is an intruder on the network, the attacker may be limited to one network segment

and most privileged accounts may be temporarily disabled. If the incident is a denial-of-service attack, the sources may be able to be identified and denied access to the target network. If one host has been compromised, communication to other hosts may be disabled.

There is much that can be done prior to an incident to make the job of containment easier. Putting critical servers on a separate subnet, for example, allows an administrator to quickly deny traffic to those servers from any other subnet or network known to be under attack.

It is prudent to consider certain situations in advance and determine how much risk to take if faced with those situations. Consider a situation where information security staff suspects that a rogue NT/2000 administrator with privileges at the top of the tree is logged in to the company's Active Directory (AD). In effect, the intruder is logged in to every Windows server defined to the AD. If staff cannot identify the workstation used by the intruder, it may be best to immediately disconnect all workstations from the network. On the other hand, such drastic action may not be warranted if the intrusion occurs on a less sensitive or less critical network segment. In another example, consider a devastating e-mail-borne worm spreading through an enterprise. At what point is the e-mail service disabled? The incident response plan should contain guidance for making this decision.

The containment phase is also the time when a message to users may be appropriate. Communication experts should craft the message, especially if it goes outside the organization.

Eradication Phase

Conceptually, eradication is simple — this is the phase in which the problem is eliminated. The methods and tools used will depend on the exact nature of the problem. For a virus incident, anti-virus signatures may have to be developed and applied; and hard drives or e-mail systems may need to be scanned before access to infected systems is allowed to resume. For an intrusion, systems into which the intruder was logged must be identified and the intruder's active sessions must be disconnected. It may be possible to identify the device used by the intruder and either logically or physically separate it from the network. If the attack originated from outside the network, connections to the outside world can be disabled.

In addition to the immediate effects of the incident, such as an active intruder or virus, other unauthorized changes may have been made to systems as a result of the incident. Eradication includes the examination of network components that may have been compromised for changes to configuration files or registry settings, the appearance of Trojan horses or backdoors designed to facilitate a subsequent security breach, or new accounts that have been added to a system.

Recovery Phase

During the recovery phase, systems are returned to a normal state. In this phase, system administrators determine (as well as possible) the extent of the damage caused by the incident and use appropriate tools to recover. This is primarily a technical task, with the nature of the incident determining the specific steps taken to recover. For malicious code, anti-virus software is the most common recovery mechanism. For denial-of-service attacks, there may not even be a recovery phase. An incident involving unauthorized use of an administrative-level account calls for a review of (at least) configuration files, registry settings, user definitions, and file permissions on any server or domain into which the intruder was logged. In addition, the integrity of critical user databases and files should be verified.

This is a phase where tough decisions may have to be made. Suppose, for example, the incident is an intrusion and an administrative account was compromised for a period of two days. The account has authority over many servers, such as in a Windows NT domain. Unless one can account for every action taken by the intruder (maybe an impossible task in the real world), one can never be sure whether the intruder altered operating system files, updated data files, planted Trojan horses, defined accounts that do not show up in directory listings, or left time bombs. The only ways to be absolutely certain that a server has been recovered back to its preincident state is to restore from backup using backup tapes known to be taken before the intrusion started, or rebuild the server by installing the operating system from scratch. Such a process could consume a significant amount of time, especially if there are hundreds of servers that could have been compromised. So if a decision is made not to restore from tape or rebuild servers, an organization takes on more risk that the problem will not be fully eradicated and systems fully restored. The conditions under which an organization is willing to live with the added risk is a matter deserving of some attention during the preparation phase.

Follow-Up Phase

It should come as no surprise that after an incident has been detected, contained, eradicated, and all recovery activities have been completed, there is still work to do. In the follow-up phase, closure is brought to the matter with a thorough review of the entire incident.

Specific activities at this time include:

- Consolidate all documentation gathered during the incident.
- Calculate the cost.
- Examine the entire incident, analyzing the effectiveness of preparation, detection, containment, eradication, and recovery activities.
- Make appropriate adjustments to the incident response plan.

Documentation should be consolidated at this time. There may have been dozens of people involved during the incident, particularly in large, geographically dispersed organizations. If legal proceedings begin years later, it is highly unlikely that the documentation kept by each participant will still exist and be accessible when needed. Therefore, all documentation must be collected and archived immediately. There should be no question about the location of all information concerning this incident. Another potential benefit to consolidating all of the documentation is that a similar incident may occur in the future, and individuals handling the new incident should be able to review material from the earlier incident.

The cost of the incident should be calculated, including direct costs due to data loss, loss of income due to the unavailability of any part of the network, legal costs, cost of recreating or restoring operating systems and data files, employee time spent reacting to the incident, and lost time of employees who could not access the network or specific services.

All aspects of the incident should be examined. Each phase of the plan should be reviewed, beginning with preparation. How did the incident occur — was there a preventable breakdown in controls, did the attacker take advantage of an old, unpatched vulnerability, was there a serious virus infection that may have been prevented with more security awareness? Exhibit 153.3 shows questions that could apply at each phase of the incident.

Appropriate adjustments should be made to the incident response plan and to information security practices. No incident response plan is perfect. An organization may be able to avoid future incidents, reduce the damage of future incidents, and get in a position to respond more effectively by applying knowledge gained from a postincident review. The review might indicate that changes should be made in any number of places, including the incident response plan, existing controls, the level of system monitoring, forensic skills of the technical staff, or the level of involvement of non-IT functions.

Other Considerations

Common Obstacles to Establishing an Effective Incident Response Plan

It may seem that any organization committed to establishing an incident response plan would be able to put one in place without much difficulty. However, there are many opportunities for failure as you address the issue of incident response. This section describes some of the obstacles that may arise during the effort.

- There is a tendency to think of serious computer security incidents primarily as IT issues to be handled on a technical level. They are not. Security incidents are primarily business issues that often have a technical component that needs prompt attention. Organizations that consider security incidents to be IT issues are more likely to make the mistake of including only IT and information security staff on the IRT.
- Technical staff with the skills to create and maintain an effective incident response plan may already be overworked simply trying to maintain and improve the existing infrastructure. There can be a tendency to have system administrators put together a plan in their spare time. Typically, these efforts lead to a lot of scurrying to get a plan thrown together in the last few days before a management-imposed deadline for its completion.

EXHIBIT 153.3 Sample Questions for Postincident Review

Preparation

- Were controls applicable to the specific incident working properly?
- What conditions allowed the incident to occur?
- Could more education of users or administrators have prevented the incident?
- Were all of the people necessary to respond to the incident familiar with the incident response plan?
- Were any actions that required management approval clear to participants throughout the incident?

Detection

- How soon after the incident started did the organization detect it?
- Could different or better logging have enabled the organization to detect the incident sooner?
- Does the organization even know exactly when the incident started?
- How smooth was the process of invoking the incident response plan?
- Were appropriate individuals outside of the incident response team notified?
- How well did the organization follow the plan?
- Were the appropriate people available when the response team was called?
- Should there have been communication to inside and outside parties at this time; and if so, was it done?
- Did all communication flow from the appropriate source?

Containment

- How well was the incident contained?
- Did the available staff have sufficient skills to do an effective job of containment?
- If there were decisions on whether to disrupt service to internal or external customers, were they made by the appropriate people?
- Are there changes that could be made to the environment that would have made containment easier or faster?
- Did technical staff document all of their activities?

Eradication and Recovery

- Was the recovery complete — was any data permanently lost?
- If the recovery involved multiple servers, users, networks, etc., how were decisions made on the relative priorities, and did the decision process follow the incident response plan?
- Were the technical processes used during these phases smooth?
- Was staff available with the necessary background and skills?
- Did technical staff document all of their activities?

- It can be difficult to get senior management's attention unless a damaging incident has already occurred. Here is where it may help to draw parallels between business continuity/disaster recovery and incident response. By and large, executives recognize the benefits of investment in a good business continuity strategy. Pointing out the similarities, especially noting that both are vehicles for managing risk, can help overcome this obstacle.
- One can think of a hundred reasons *not* to conduct exercises of the plan. Too many people are involved; it is difficult to stage a realistic incident to test the plan; everybody is too busy; it will only scare people; etc. Lack of testing can very quickly render an incident response plan less than adequate. Good plans evolve over time and are constantly updated as the business and technical environments change. Without periodic testing and review, even a well-constructed incident response plan will become much less valuable over time.

The Importance of Training

It is crucial that an organization conduct training exercises. No matter how good an incident response plan is, periodic simulations or walk-throughs will identify flaws in the plan and reveal where the plan has not kept pace with changes in the automation infrastructure. More importantly, it will keep IRT members aware of the general flow as an incident is reported and the organization responds. It will give technical staff an opportunity to utilize tools that may not be used normally. Each exercise is an opportunity to ensure that all of the tools that might be needed during an incident are still functioning as intended. Finally, it will serve to make key participants more comfortable and more confident during a real incident.

Benefits of a Structured Incident Response Methodology

As this chapter describes, there is nothing trivial about preparing to respond to a serious computer security incident. Development and implementation of an incident response plan require significant resources and specialized skills. It is, however, well worth the effort for the following reasons.

- *An incident response plan provides structure to a response.* In the event of an incident, an organization would be extremely lucky if its technicians, managers, and users all do what they think best and those actions make for an effective response. On the other hand, the organization will almost always be better served if those people acted against the backdrop of a set of guidelines and procedures designed to take them through each step of the way.
- *Development of a plan allows an organization to identify actions and practices that should always be followed during an incident.* Examples are maintaining a log of activities, maintaining an evidentiary chain of custody, notifying specific entities of the incident, and referring all media inquiries to the public relations staff.
- *It is more likely that the organization will communicate effectively to employees if an incident response plan is in place.* If not, messages to management and staff will tend to be haphazard and may make the situation worse.
- *Handling unexpected events is easier if there is a framework that is familiar to all the participants.* Having critical people comfortable with the framework can make it easier to react to the twists and turns that sometimes occur during an incident.

Years ago, security practitioners and IT managers realized that a good business continuity plan was a sound investment. Like business continuity, a computer incident response plan has become an essential part of a good security program.

Notes

1. CERT/CC *Incident Reporting Guidelines*, available at http://www.cert.org/tech_tips/ incident_ reporting.html.
2. *CIAC Incident Reporting Procedures*, available at http://doe-is.llnl.gov/.

154

Cyber-Crime: Response, Investigation, and Prosecution

Thomas Akin, CISSP

> Any sufficiently advanced form of technology is indistinguishable from magic.
>
> **— Arthur C. Clark**

As technology grows more complex, the gap between those who understand technology and those who view it as magic is getting wider. The few who understand the magic of technology can be separated into two sides — those who work to protect technology and those who try to exploit it. The first are information security professionals, the latter hackers. To many, a hacker's ability to invade systems does seem magic. For security professionals — who understand the magic — it is a frustrating battle where the numbers are in the hackers' favor. Security professionals must simultaneously protect every single possible access point, but a hacker only needs a single weakness to successfully attack a system. The lifecycle in this struggle is:

- Protection
- Detection
- Response
- Investigation
- Prosecution

First, organizations work on protecting their technology. Because 100 percent protection is not possible, organizations realized that if they could not completely protect their systems, they needed to be able to detect when an attack occurred. This led to the development of intrusion detection systems (IDSs). As organizations developed and deployed IDSs, the inevitable occurred: "According to our IDS, we've been hacked! Now what?" This quickly led to the formalization of incident response. In the beginning, most organizations' response plans centered on getting operational again as quickly as possible. Finding out the identity of the attacker was often a low priority. But as computers became a primary storage and transfer medium for money and proprietary information, even minor hacks quickly became expensive. In attempts to recoup their losses, organizations are increasingly moving into the investigation and prosecution stages of the life cycle. Today, although protection and detection are invaluable, organizations must be prepared to effectively handle the response, investigation, and prosecution of computer incidents.

Response

Recovering from an incident starts with how an organization responds to that incident. It is rarely enough to have the system administrator simply restore from backup and patch the system. Effective response will greatly affect the ability to move to the investigation phase, and can, if improperly handled, ruin any chances of prosecuting the case. The high-level goals of incident response are to preserve all evidence, remove the

vulnerability that was exploited, quickly get operational again, and effectively handle PR surrounding the incident. The single biggest requirement for meeting all of these goals is preplanning. Organizations must have an incident response plan in place before an incident ever occurs. Incidents invariably cause significant stress. System administrators will have customers and managers yelling at them, insisting on time estimates. Executives will insist that they "just get the damn thing working!" Even the customer support group will have customers yelling at them about how they need everything operational now. First-time decisions about incident response under this type of stress always lead to mistakes. It can also lead to embarrassments such as bringing the system back online only to have it hacked again, deleting or corrupting the evidence so that investigation and prosecution are impossible, or ending up on the evening news as the latest casualty in the war against hackers.

To be effective, incident response requires a team of people to help recover from the incident. Technological recovery is only one part of the response process. In addition to having both IT and information security staff on the response team, there are several nontechnical people who should be involved. Every response should include a senior executive, general counsel, and someone from public relations. Additionally, depending on the incident, expanding the response team to include personnel from HR, the physical security group, the manager of the affected area, and even law enforcement may be appropriate.

Once the team is put together, take the time to plan response priorities for each system. In a Web server defacement, the top priorities are often getting the normal page operational and handling PR and the media. If an online transaction server is compromised and hundreds of thousands of dollars are stolen, the top priority will be tracking the intruder and recovering the money. Finally, realize that these plans provide a baseline only. No incident will ever fall perfectly into them. If a CEO is embezzling money to pay for online sex from his work computer, no matter what the standard response plan calls for, the team should probably discreetly contact the organization's president, board of directors, and general counsel to help with planning the response. Each incident's "big picture" may require changes to some of the preplanned details, but the guidelines provide a framework within which to work.

Finally, it is imperative to make sure the members of the response team have the skills needed to successfully respond to the incident. Are IT and InfoSec staff members trained on how to preserve digital evidence? Can they quickly discover an intruder's point of entry and disable it? How quickly can they get the organization functional again? Can they communicate well enough to clearly testify about technology to a jury with an average education level of sixth grade? Very few system or network administrators have these skills — organizations need to make sure they are developed. Additionally, how prepared is the PR department to handle media inquiries about computer attacks? How will they put a positive spin on a hacker stealing 80,000 credit card numbers from the customer database? Next, general counsel — how up to date are they on the ever-changing computer crime case law? What do they know about the liability an organization faces if a hacker uses its system to attack others?

Without effective response, it is impossible to move forward into the investigation of the incident. Response is more than "just get the damn thing working!" With widespread hacking tools, a volatile economy, and immature legal precedence, it is not enough to know how to handle the hacker. Organizations must also know how to handle customers, investors, vendors, competitors, and the media to effectively respond to computer crime.

Investigation

When responding to an incident, the decision of whether to formally investigate will have to be made. This decision will be based on factors such as the severity of the incident and the effect an investigation will have on the organization. The organization will also have to decide whether to conduct an internal investigation or contact law enforcement. A normal investigation will consist of:

- Interviewing initial personnel
- A review of the log files
- An intrusion analysis
- Forensic duplication and analysis
- Interviewing or interrogating witnesses and suspects

Experienced investigators first determine that there actually was an intrusion by interviewing the administrators who discovered the incident, the managers to whom the incident was reported, and even users to determine if they noticed deviations in normal system usage. Next, they will typically review system and

network log files to verify the organization's findings about the intrusion. Once it is obvious that an intrusion has occurred, the investigator will move to a combination of intrusion analysis and forensics analysis. Although they often overlap, intrusion analysis is most often performed on running systems, and forensic analysis is done offline on a copy of the system's hard drive. Next, investigators will use the information discovered to locate other evidence, systems to analyze, and suspects to interview. If the attacker came from the outside, then locating the intruder will require collecting information from any third parties that the attacker passed through. Almost all outside organizations, especially ISPs, will require either a search warrant or subpoena before they will release logs or subscriber information. When working with law enforcement, they can provide the search warrant. Nonlaw enforcement investigators will have to get the organization to open a "John Doe" civil lawsuit to subpoena the necessary information. Finally, while the search warrant or subpoena is being prepared, investigators should contact the third party and request that they preserve the evidence that investigators need. Many ISPs delete their logs after 30 days, so it is important to contact them quickly.

Due to the volatility of digital evidence, the difficulty in proving who was behind the keyboard, and constantly changing technology, computer investigations are very different from traditional ones. Significant jurisdictional issues can come up that rarely arise in normal investigations. If an intruder resides in Canada, but hacks into the system by going first through a system in France and then a system in China, where and under which country's laws are search warrants issued, subpoenas drafted, or the case prosecuted? Because of these difficulties, international investigations usually require the involvement of law enforcement — typically the FBI. Few organizations have the resources to handle an international investigation. Corporate investigators can often handle national and internal investigations, contacting law enforcement only if criminal charges are desired.

Computer investigations always involve digital evidence. Such evidence is rarely the smoking gun that makes or breaks an investigation; instead, it often provides leads for further investigation or corroborates other evidence. For digital evidence to be successfully used in court, it needs to be backed up by either physical evidence or other independent digital evidence such as ISP logs, phone company records, or an analysis of the intruder's personal computer. Even when the evidence points to a specific computer, it can be difficult to prove who was behind the keyboard at the time the incident took place. The investigator must locate additional proof, often through nontechnical means such as interviewing witnesses, to determine who used the computer for the attack.

Much of technology can be learned through trial and error. Computer investigation is not one of them. Lead investigators must be experienced. No one wants a million-dollar suit thrown out because the investigator did not know how to keep a proper chain of custody. There are numerous opinions about what makes a good investigator. Some consider law enforcement officers trained in technology the best. Others consider IT professionals trained in investigation to be better. In reality, it is the person, not the specific job title, that makes the difference. Investigators must have certain qualities. First, they cannot be afraid of technology. Technology is not magic, and investigators need to have the ability to learn any type of technology. Second, they cannot be in love with technology. Technology is a tool, not an end unto itself. Those who are so in love with technology that they always have be on the bleeding edge lack the practicality needed in an investigation. An investigator's nontechnical talents are equally important. In addition to strong investigative skills, he or she must have excellent communications skills, a professional attitude, and good business skills. Without good oral communications skills, an investigator will not be able to successfully interview people or testify successfully in court if required. Without excellent written communications skills, the investigator's reports will be unclear, incomplete, and potentially torn apart by the opposing attorney. A professional attitude is required to maintain a calm, clear head in stressful and emotional situations. Finally, good business skills help make sure the investigator understands that sometimes getting an organization operational again may take precedent over catching the bad guy.

During each investigation, the organization will have to decide whether to pursue the matter internally or to contact law enforcement. Some organizations choose to contact law enforcement for any incident that happens. Other organizations never call them for any computer intrusion. The ideal is somewhere in between. The decision to call law enforcement should be made by the same people who make up the response team — senior executive management, general counsel, PR, and technology professionals. Many organizations do not contact law enforcement because they do not know what to expect. This often comes from an organization keeping its proverbial head in the sand and not preparing incident response plans ahead of time. Other reasons organizations may choose not to contact law enforcement include:

- They are unsure about law enforcement's computer investigation skills.
- They want to avoid publicity regarding the incident.

- They have the internal resources to resolve the investigation successfully.
- The incident is too small to warrant law enforcement attention.
- They do not want to press criminal charges.

The reasons many organization will contact law enforcement are:

- They do not have the internal capabilities to handle the investigation.
- They want to press criminal charges.
- They want to use a criminal prosecution to help in a civil case.
- They are comfortable with the skills of law enforcement in their area.
- The incident is international in scope.

All of these factors must be taken into account when deciding whether to involve law enforcement. When law enforcement is involved, they will take over and use state and federal resources to continue the investigation. They also have legal resources available to them that corporate investigators do not. However, they will still need the help of company personnel because those people are the ones who have an in-depth understanding of policies and technology involved in the incident. It is also important to note that involving law enforcement does not automatically mean the incident will be on the evening news. Over the past few years, the FBI has successfully handled several large-scale investigations for Fortune 500 companies while keeping the investigation secret. This allowed the organizations to publicize the incident only after it had been successfully handled and avoid damaging publicity. Finally, law enforcement is overwhelmed by the number of computer crime cases they receive. This requires them to prioritize their cases. Officially, according to the Computer Fraud and Abuse Act, the FBI will not open an investigation if there is less than $5000 in damages. The actual number is significantly higher. The reality is that a defaced Web site, unless there are quantifiable losses, will not get as much attention from law enforcement as the theft of 80,000 credit card numbers.

Prosecution

After the investigation, organizations have four options — ignore the incident, use internal disciplinary action, pursue civil action, or pursue criminal charges. Ignoring the incident is usually only acceptable for very minor infractions where there is very little loss and little liability from ignoring the incident. Internal disciplinary action can be appropriate if the intruder is an employee. Civil lawsuits can be used to attempt to recoup losses. Criminal charges can be brought against those violating local, state, or federal laws. Civil cases only require a "preponderance of evidence" to show the party guilty; criminal cases require evidence to prove someone guilty "beyond a reasonable doubt."

When going to trial, not all of the evidence collected will be admissible in court. Computer evidence is very different from physical evidence. Computer logs are considered hearsay and therefore generally inadmissible in court. However, computer logs that are regularly used and reviewed during the normal course of business are considered business records and are therefore admissible. There are two points to be aware of regarding computer logs. If the logs are simply collected but never reviewed or used, then they may not be admissible in court. Second, if additional logging is turned on during the course of an investigation, those logs will not be admissible in court. That does not mean additional logging should not be performed but that such logging needs to lead to other evidence that will be admissible.

Computer cases have significant challenges during trial. First, few lawyers understand technology well enough to put together a strong case. Second, fewer judges understand technology well enough to rule effectively on it. Third, the average jury has extremely little or no computer literacy. With these difficulties, correctly handling the response and investigation phases is crucial because any mistakes will confuse the already muddy waters. Success in court requires a skilled attorney and expert witnesses, all of whom can clearly explain complex technology to those who have never used a computer. These challenges are why many cases are currently plea-bargained before ever going to trial.

Another challenge organizations face is the financial insolvency of attackers. With the easy availability of hacking tools, many investigations lead back to teenagers. Teenagers with automatic hacking tools have been able to cause billions of dollars in damage. How can such huge losses be recovered from a 13-year-old adolescent? Even if the attacker were financially successful, there is no way an organization could recoup billions of dollars in losses from a single person.

It is also important to accurately define the losses. Most organizations have great difficulty in placing a value on their information. How much is a customer database worth? How much would it cost if it were given to a competitor? How much would it cost if it were inaccessible for three days? These are the type of questions organizations must answer after an incident. It is easy to calculate hardware and personnel costs, but calculating intangible damages can be difficult. Undervalue the damages, and the organization loses significant money. Overvaluing the damages can hurt the organization's credibility and allow opposing counsel to portray the organization as a money-hungry goliath more interested in profit than the truth.

Any trial requires careful consideration and preparation — those involving technology even more so. Successful civil and criminal trials are necessary to keep computer crime from becoming even more rampant; however, a successful trial requires that organizations understand the challenges inherent to a case involving computer crime.

Summary

For most people, technology has become magic — they know it works, but have no idea how. Those who control this magic fall into two categories — protectors and exploiters. Society uses technology to store and transfer more and more valuable information every day. It has become the core of our daily communications, and no modern business can run without it. This dependency and technology's inherent complexity have created ample opportunity for the unethical to exploit technology to their advantage. It is each organization's responsibility to ensure that its protectors not only understand protection but also how to successfully respond to, investigate, and help prosecute the exploiters as they appear.

Response Summary

- Preplan a response strategy for all key assets.
- Make sure the plan covers move than only technological recovery — it must address how to handle customers, investors, vendors, competitors, and the media to be effective.
- Create an incident response team consisting of personnel from the technology, security, executive, legal, and public relations areas of the organization.
- Be flexible enough to handle incidents that require modifications to the response plan.
- Ensure that response team members have the appropriate skills required to effectively handle incident response.

Investigation Summary

- Organizations must decide if the incident warrants an investigation.
- Who will handle the investigation — corporate investigators or law enforcement?
- Key decisions should be made by a combination of executive management, general counsel, PR, and technology staff members.
- Investigators must have strong skills in technology, communications, business, and evidence handling — skills many typical IT workers lack.
- Digital evidence is rarely a smoking gun and must be corroborated by other types of evidence or independent digital evidence.
- Knowing what computer an attack came from is not enough; investigators must be able to prove the person behind the keyboard during the attack.
- Corporate investigators can usually successfully investigate national and internal incidents. International incidents usually require the help of law enforcement.
- Law enforcement, especially federal, will typically require significant damages before they will dedicate resources to an investigation.

Prosecution Summary

- Organizations can ignore the incident, use internal disciplinary action, pursue civil action, or pursue criminal charges.
- Civil cases require a "preponderance of evidence" to prove someone guilty; criminal cases require evidence "beyond a reasonable doubt."
- Most cases face the difficulties of financially insolvent defendants; computer-illiterate prosecutors, judges, and juries; and a lack of strong case law.
- Computer logs are inadmissible as evidence unless they are used in the "normal course of business."
- Due to the challenges of testifying about complex technology, many cases result in a plea-bargain before they ever go to trial.
- Placing value on information is difficult, and overvaluing the information can be as detrimental as undervaluing it.
- Most computer attackers are financially insolvent and do not have the assets to allow organizations to recoup their losses.
- Successful cases require attorneys and expert witnesses to be skilled at explaining complex technologies to people who are computer illiterate.

155

Incident Response Exercises

Ken M. Shaurette, CISSP, CISA, CISM, IAM and Thomas J. Schleppenbach

It was a quiet clear morning at about 2:26 a.m. I was sleeping soundly when I felt something on my leg. Whatever it was it was smaller than Holly, our cat, but bigger than a bug or a mouse. I quickly rolled over and, taking a swipe with my hand, knocked it off the bed. Without my glasses I could barely see anything, but I saw something run into the master bathroom. I thought to myself, "That sure looked like a small bunny rabbit."

I got up a bit apprehensive, put on my slippers, and found my glasses so I could focus better. Slowly I walked to the bathroom and, sure enough, there it was; something about the size of a softball, all brown and furry, crouching by the toilet. I had not turned the lights on, so it was still dark and hard to see. I stepped back quickly and closed the bathroom door. Gotcha. I had stopped the animal's activity by confining it to the master bathroom. Now what was I going to do? I walked over to the bed and tapped my wife on the shoulder. "What the heck is going on?" she asked, and I said, "I think there is a dangerous animal in our bathroom, it could be a rabbit." She said, "You're just having a bad dream, go back to sleep." I said, "No I can't, I saw it and felt it on my leg. I knocked it off and now I have it confined to the bathroom. I think it's a rabbit!" She said, "You've been stressed out lately, you're just having a dream, go back to bed." I said, "I don't think so; there is a rabbit in our bathroom." She followed with, "Are you sure? What are we going to do? We should call the Department of Natural Resources!" I said, "No, that won't work; the DNR isn't going to do anything at 2:30 in the morning." Suddenly, as quickly as she had doubted me, she asked, "Can we keep it?" I responded, "No, this is a wild rabbit; we need to get it out of here and figure out how it got in."

What was I to do? That thing could make a terrible mess in the bathroom, I thought, remembering that the kids had butterfly nets downstairs and that I had a pair of old leather gloves down on the counter in the kitchen that I had just used that afternoon. I gathered up my makeshift antirabbit tools and went back upstairs to the bathroom. I went in the bathroom and blocked the escape route by shutting the door behind me. I was now prepared to do battle. I could see it crouching motionless behind the toilet. I quickly determined that if I put the butterfly net on one side of the toilet, I could use the small bathroom garbage can to encourage the little beast to go toward the other side. I swiftly put my counter attack in motion. My hasty plan had worked; the critter ran out from under the toilet right into the net. I pounced on it, quickly grabbing the netting to trap a small rabbit, yes a bunny rabbit, in the net. I picked it up and carried it outside. Shortly, the bunny was released back into the wild.

As I put the battle tools away and walked back to the bedroom, I passed my fearless dog, a yellow lab, sleeping peacefully at the top of the stairs. I patted him and said, "Thanks a lot, where were you? That thing must have hopped right past your nose. Man's best friend indeed!" I walked back into the master bedroom and there was Holly, our fearless cat, rolled up in a cozy little ball in the corner. I thought to myself, "What about you? You didn't do your job either; you are supposed to protect me from undesirable events like what just happened." I got nothing from her but a little meow scolding me for the disturbance.

So now I am at work, a little bit tired from the whole experience and thinking, how did that rabbit get into our house? What measures can I take to better manage the risk of losing another night's sleep in the future? Why did the protection measures — the door, the dog, the cat — not stop the event from occurring? Thank goodness I woke up and was able to put my makeshift response plan into effect.

Information Security: Layered Security

By now you have to be wondering what this story about a little bunny has to do with information security. This entire event brings to mind issues about intrusion detection, incident response procedures, testing, and overall security infrastructure as well as protecting evidence. It is really a great analogy covering several of these aspects. So we will evaluate the incident for comparisons to information security.

Start by asking a few basic questions such as: How secure is your perimeter? Has it been tested for vulnerabilities to undesirable entry? Are you prepared for a security incident? Are you prepared to respond?

We will respond to each question by comparing them to a typical environment.

How Secure Is Your Perimeter?

Is your firewall like the door of the house that let the bunny slip though? Was it just a small hole that allowed undesirable access to the internal environment? It did not take a very large hole to let the bunny into the house. I consciously opened the small porthole so I did not have to keep getting up to let the pets in or out. Does that sound familiar? "We just need one port open in order for this application to work."

Has It Been Tested for Vulnerabilities to Undesirable Entry?

I made sure that both my cat and dog were able to come and go using the small hole that I had opened. I did not consider what other, less-desirable creatures might take advantage of this opportunity. If I had only configured the hole a little differently, it could have kept out many of the other undesirables, including the bunny, and still have been functional for my pets to use.

Are You Prepared for a Security Incident? Are You Prepared to Respond?

I was not prepared. Who would have imagined a little bunny wanting to get into the house? There is nothing inside my house wild animals would want, why should I be concerned? Does your corporation have information someone might see as valuable, or perhaps a network that someone, like the bunny, might just be curious to check out? I was totally unprepared and did not expect the events that occurred. How many times do you hear from users that they do not have access to anything that anyone else would want? I was on my last line of defense and just lucky to have the tools available to quarantine as well as to capture and remove the unwanted visitor.

Are you ready with the necessary tools to stop an intruder? Do you have a policy against normal users running hacker-type tools inside your environment? Does the policy allow for administrators to access similar tools to identify vulnerabilities and track incidents? After you stop an intruder, can you capture the necessary evidence to track any damage that may have been done? Is there an incident response plan in place that would have clearly instructed you on who to call and what to do to protect the evidence of an intrusion? Does your organization plan to prosecute for damages? Does a process exist to ensure that the evidence does not get damaged or tampered with and that a proper chain of custody is in place so the evidence retains its forensic quality and will hold up in court?

This chapter will not attempt to answer all of these questions, but it should leave you with some ideas, points to ponder, and actions to consider.

An incident could be something as simple as an attacker (the bunny) spreading (hopping around the bedroom) the latest virus or worm ("rabbit raisins," poop, all over the room), or a more serious incident like using your e-mail server to send spam to other companies (the bunny biting one of your children) or maybe even penetrating through the external security architecture, the firewall (exterior doors), and getting inside the organization to disrupt services or steal intellectual property and confidential information (eating the dog and cat food or chewing on furniture).

EXHIBIT 155.1 Intrusion Detection: Incident Response Questions

- What actions are to be taken to identify that this is in fact an unwanted attacker who has penetrated the organization?
- Is there a call list for specific incidents?
- Can an automated action be taken to react to the alert, such as closing a port, or shutting down a service?
- Is it possible to identify the type of attack being used from the events logged and the intrusion detection information? (Refer to Figure 155.2 for different types of attacks.)
- Would it be possible to determine where the attack is coming from or where it originated?
- Is this an organized attack against your organization and similar organizations in the same industry?
- What might an attacker want that the organization has, or what might be lost: reputation, public confidence, integrity, credibility?
- Is it possible to identify when the incident occurred along with all previous attempts that may or may not have failed?
- If there is real damage, would it be possible to get sufficient evidence to show damage, or evidence that can stand up to a court's scrutiny and meet forensic-level quality?

Information security is all about defense-in-depth, layering protection so that the valuable assets of the organization are properly protected. In the bunny story I was essentially the last line of defense to protect my family and property from this rogue rabbit that had penetrated my exterior defenses.

What is the first thing you would do if you received a page from your Incident Response System or server system log paging software at 2:26 a.m. alerting you to the potential of a breach of external security?

On the other hand, perhaps the intrusion detection system generates so many alerts that system and network administrators have become numb to them and the messages are just ignored until the next day? Proper configuration of an intrusion detection solution so that it only sends message alerts for events that are considered issues is critical. Numerous alerts, such as every time the network is being "pinged," will cause numbness, resulting in a technician ignoring alerts and potentially missing the real thing.

Just having intrusion detection is inadequate protection. Without an incident response plan to react to the intrusion, just logging the event is not very effective. It is very important to identify the steps to take beyond simply preparing for a long night (or day) by brewing another pot of strong coffee or getting a couple more liters of Code Red.

Most operating environments and network devices already produce volumes of logged activity. The availability of this data causes a need for answers to several more questions (refer to Exhibit 155.1). Finding answers to the questions as outlined will help you select and configure effective intrusion detection as well as plan an organization's incident response system.

Preparing for an incident by planning and building the entire security program is essential. The security program becomes that defense-in-depth or layering of protection. Planning for security as well as selecting and testing intrusion detection and incident response is outlined in the remainder of this chapter.

What Is an Information Security Operations Plan?

Every organization should have an Information Security Operations Plan (ISOP) as the starting point for layers of security. The plan establishes the components in an organization's security program. It ensures that an organization does not place too much emphasis on technology and not enough on people and process. It prioritizes the security activities that will be focused on during each year, helps set budget, and provides a status reports to management on the state of all security activities. The plan should include functional areas defined by industry standards such as ISO17799. Before framing an incident response system, consider the components of an Information Security Operations Plan. The components of an effective security plan include:

- *Baseline:* This establishes where the company is at present. It is a high-level position statement of where security is at this point in time. It becomes an annual review to understand the current status of information security efforts in the organization. Each year it establishes what has been completed in the plan, areas of change, and new areas that have been added during the year.
- *Policies, Standards, and Procedures:* Policies, standards, and procedures are continuously changing. Information Security Policy provides the roadmap by which an organization identifies security philosophy and establishes the importance of security in the organization. Policy is the roadmap that defines

appropriate handling of information in the business environment and sets the ground rules for building the information security architecture and technology. It helps determine the requirements for information security by setting expectations and requires management commitment and sign-off at the highest levels.

- *Architecture and Processes:* Designing security into the creation, selection, approval, and roll out of all technologies is vital. Security must be an integral part of building the data processing environment. Including information security early in the process of application and system development and selection will ensure that security issues can be addressed and that alternatives have sufficient lead time to be implemented within business deadlines. Secure architectures and the processes to support them are crucial to a secure environment.
- *Awareness and Training:* Every computer user in the organization must be made aware of company policy. An effective security program requires that everyone understands their personal responsibilities to protect the corporate information assets to help minimize organization liability.
- *Technologies and Products:* Technology is an important component of the security program. Although people and process make up potentially 70 percent of the security structure, technology alone accounts for probably 30 percent of the requirements to protect an organization. Technologies can range from simple system monitoring tools and access controls such as passwords and multiple factor authentication systems to virtual private networks (VPN) and data encryption or public key infrastructure (PKI) systems. Often, third-party vendor products are required to support and monitor the operating environment.
- *Assessment and Monitoring:* To meet the needs and expectations of customers, auditors, and various levels of management, appropriate information must be collected so that reports can be created and distributed. Perimeter connections to the network and host system logs must also be monitored for unauthorized activities.
- *Compliance:* The mission of information security is to minimize security risks while maintaining the least possible impact on cost and schedules. To meet both company and customer expectations for information security, it is necessary to implement a process of continuous feedback so that business units can provide input to the improvement of information security planning.

The ISOP will frame out the organization's security program. The incident response program and procedures would be included as a component or subset of the overall information security program. In the next few paragraphs we focus on incident response and preparedness.

What Are the Components of an Incident Response Program?

An incident response program should include:

- Forming an incident response team
- Identifying a main contact (this must be a decision maker)
- Defining the monitoring or intrusion detection strategy
- Establishing an incident response flow
- Developing a set of basic required actions based on incident
- Preparing for recovery (business continuance)
- Knowing how and when to report an incident

There are several best-practice guidelines that are worth following, many of which can be found in books such as *Critical Incident Management* by Alan B. Sterneckert (Auerbach Publications, 2004).

As organizations develop their incident response program and associated procedures, they must take into consideration the specifics and details of their own unique networking and operating environment that only a person familiar with the inside workings of the organization would have. An important aspect in establishing intrusion detection and incident response is gaining an understanding of some of the typical attacker intrusion approaches.

Attack Approaches

To better understand intrusion activity and the process of identifying undesirable events, it is important to look at hacking approaches. Attacks can be separated into the following categories:[1]

- *Bomb:* This is a general synonym for crash, normally consisting of software or operating system failures.
- *Buffer Overflow:* This happens when more data is put into a buffer or holding area than the buffer can handle. It can be a result when there is a mismatch between processing rates of the producing and consuming processes. This can result in system crashes or the creation of a backdoor leading to system access.
- *Demon Dialer:* One name for a program that repeatedly calls the same telephone number is a demon dialer. This can be benign and legitimate for access to an authorized network or malicious when used as a denial-of-service attack.
- *Derf:* This is the name given to the act of exploiting a terminal which someone else has absentmindedly left logged on.
- *DNS (Domain Name Service) Spoofing:* The process of assuming the DNS name of another system by either corrupting the name service cache of a victim system or by compromising a domain name server to obtain a valid domain is called "spoofing."
- *Ethernet Sniffing:* This refers to the action of listening for packets or datagrams with software on the network looking at the Ethernet interface for packets that interest the user. Because Ethernet sends data by broadcasting all packets to all machines connected to the local network, it is trivial to receive packets that were intended for other machines. Ethernet interfaces support a feature commonly called "promiscuous mode," in which the interface listens to network traffic "promiscuously." That is, instead of dropping all packets that do not have the machine's Ethernet address in them, the interface processes all of the packets that it receives. When the software sees a packet that fits certain criteria, it logs it to a file. The most common criteria for an interesting packet are ones that contain words like log-in or password.
- *Fork Bomb:* Also known as Logic Bomb. Code that can be written in one line of code on any UNIX system; used to recursively spawn copies of itself; "explodes," eventually eating all the process table entries and effectively locks up the system.
- *IP Splicing/Hijacking:* The action caused when an active, established session is intercepted and co-opted by an unauthorized user. IP splicing attacks can occur after an authentication has been made, permitting the attacker to assume the role of an already authorized user. Primary protections against IP splicing rely on encryption at the session or network layer.
- *IP Spoofing:* This type of attack occurs when the attacker causes one system to impersonate another system by using the system's IP network address without proper authorization. Essentially the attacker impersonates a different address than is normally assigned to him.
- *Keystroke Monitoring:* A specialized form of logging software, or a specially designed hardware device usually placed between the keyboard and the CPU. The device or software can record every keystroke a user makes. Properly used and secured, a legitimate use for this functionality is to capture forensic-quality evidence for prosecuting illegal computer incidents. Improper use can result in an intruder capturing passwords and other personal information.
- *Leapfrog Attack:* The leapfrog attack results in the use of an illicitly obtained user ID and password gained from compromise of information on one host to compromise another host; for example, the act of TELNETing through one or more hosts to confuse attempts to trace the activity. This is a very common attacker activity used to make tracking undesirable activity back to the actual source more difficult.
- *Letter Bomb:* A piece of e-mail containing live data intended to do malicious things to the recipient's machine or terminal. In a UNIX environment, a letter bomb could try to get part of its contents interpreted as a shell command to the mailer. The results of this could range from silly to denial of service or complete system compromise.

- *Logic Bomb:* Also known as a Fork Bomb. A resident computer program which, when executed, checks for a particular condition or particular state of the system that, when satisfied, triggers the perpetration of an unauthorized act. These could be planted in the operating system software or coded into application code by an unscrupulous programmer.
- *Mail Bomb:* The mail sent to urge others to send massive amounts of e-mail to a single system or person, with the intent to crash the target recipient's system. Mail bombing is widely regarded as a serious offense. In more minor amounts or when not targeted to only one system, this is commonly known as spam.
- *Malicious Code:* This hardware, software, or firmware can be intentionally included in a system for an unauthorized purpose; e.g., a Trojan horse, virus, or any other code that might demonstrate nasty, undesirable behavior.
- *Mimicking:* This term is synonymous with impersonation, masquerading, or spoofing.
- *NAK Attack:* NAK stands for negative acknowledgment. It is used as a penetration technique that capitalizes on a potential weakness in an operating system that does not handle asynchronous interrupts properly, and thus leaves the system in an unprotected state during such interrupts.
- *Network Weaving:* Another name for leapfrogging.
- *Phreaking:* This describes the art and science of cracking the telephone networks.
- *Replicator:* A program that copies itself is called a replicator program. Examples include a worm, a fork bomb, or virus. It is even claimed by some that UNIX and C are the symbiotic halves of an extremely successful replicator.
- *Retro-Virus:* A retro-virus is a form of malicious code that waits until all possible backup media are infected, so that it is not possible to restore the system to an uninfected state.
- *Rootkit:* The "root kit" is a hacker security tool that provides that ability to capture passwords and message traffic to and from a computer. It is a collection of tools that allow a hacker to create a backdoor into a system, collect information on other systems on the network, mask the fact that the system is compromised, and much more. Rootkit is a classic example of Trojan horse software and is available for a wide range of operating systems. It gets its name from the name of the system administrative account in UNIX operating environments.
- *Smurfing:* Smurfing is an attack of a network by using spoofing of the source address to exploit Internet Protocol (IP) broadcast addressing and certain other aspects of Internet operation. Smurfing uses a program called Smurf and similar programs to cause the attacked part of a network to become inoperable, such as in a denial-of-service attack. The exploit of smurfing, as it has come to be known, takes advantage of certain known characteristics of the Internet Protocol (IP) and the Internet Control Message Protocol (ICMP). The ICMP is used by network nodes and their administrators to exchange information about the state of the network.
- *Spoofing:* Pretending to be someone else; also see mimicking. This is the deliberate inducement of a user or a resource to take an incorrect action. An attempt to gain access to a system by pretending to be an authorized user.
- *Subversion:* This intrusion act occurs when an intruder modifies the operation of the intrusion detector to force false-negatives to occur. The act can cause an intrusion detection system to send traffic that camouflages an attack.
- *SYN Flood:* The SYN flood attack sends TCP connection requests faster than a machine can process them. When the SYN queue is flooded, no new connection can be opened. The attacker creates a random source address for each packet. A SYN flood attack can be used as part of other attacks, such as disabling one side of a connection in TCP hijacking or by preventing authentication or logging between servers.
- *Terminal Hijacking:* This attack method allows an attacker, on a certain machine, to control any terminal session that is in progress. An attacker can send and receive terminal I/O while a user is on the terminal.
- *Trojan Horse:* The Trojan Horse can appear as an apparently useful and innocent program, but actually contains additional hidden code that allows the unauthorized collection, exploitation, falsification, or destruction of data. The actions can be activated by some other event such as on a specific date or when the deletion occurs of a specific account on a system.

- *Virus:* This is the common name assigned to many malicious programs that can "infect" other programs by modifying them to include a possibly evolved copy of itself.
- *Wardialer:* Made popular by the 1980s movie, *War Games,* this consists of a program that can dial a list or range of numbers and record those that answer with handshake tones, which might be entry points to computer or telecommunications systems. Handshake tones are "answer" tones given by a modem set to answer a request for connection.

Understanding the attack methods can be helpful in understanding why some features are important in selection of an intrusion detection system.

Selecting Intrusion Detection

Picking the proper intrusion detection (IDS) technology is an important step not to be taken lightly, and is not an easy task. An IDS should be easy to install and require minimal training, and should deploy in a "passive" or "parallel" mode, and not inline, which creates a potential bottleneck and failure point. To help with selection of an intrusion detection system, a capabilities matrix has been provided in Exhibit 155.2.

Organizations must establish their internal requirements and priorities as they pertain to intrusion detection, to establish the components identified in Figure 155.3 that are most important in a product.

Once the technology is chosen and deployed, just like a disaster recovery plan, it would be wise to periodically test it along with incident response plans.

Incident Response Exercises

Incident response exercises are one method in helping reduce organizational risk and better prepare a company's staff to respond to intrusive behavior. Like war games, incident response exercises are designed to raise awareness to the security posture of an organization through the continuous testing of incident response procedures and network device and system configuration. The testing is followed by regular review with experienced information security personnel and the organization's IS staff.

The goal is to raise security awareness with an organization's IS and management staff, and to verify and enforce proper and continual setup, configuration, and tuning of security and network systems while ensuring they are kept up to current patch levels.

The IS staff gets continuous exposure to real-world infiltration scenarios in a controlled environment, educating them to identify malicious behavior as well as having the organization's incident response procedures properly tested.

Incident response exercises generally work through continual footprinting of an organization's resources, and surprise infiltrations that are followed up with a meeting to discuss the "whats": what went right, what went wrong, and what could be done better?

If an organization is outsourcing the incident exercise service, the organization must be sure to check references and work with a credible company. Here are a few basic rules for selecting a security outsourcing vendor:

- Demand credentials and expertise, consider background checks
- Seek outside certifications
- Ensure vendor affiliations
- Talk to other customers and references
- Comparison shop
- Take small steps if unsure
- Know your escalation procedures
- Require standard inspections
- Know the rules with the vendor

EXHIBIT 155.2 Modern IDS Capability Comparison

Modern IDS Capability Comparison	Product 1	Product 2	Product 3	Product 4
2.0 Detection				
2.1 Protocol anomaly detection				
2.2 DoS attack detection				
2.3 Network infrastructure attack detection				
2.4 Common application protocol detection				
2.5 Stateful signature detection				
2.6 Custom signature support				
2.7 Full protocol decode				
2.8 Evasion detection and resistance to IDS attack				
2.9 Full fragment reassembly				
2.10 Full multi-interface reassembly				
3.0 Analysis				
3.1 Third-party event integration				
3.2 Real-time event aggregation				
3.3 Real-time analysis				
3.4 Automated correlation and prioritization				
3.5 Cross-node event correlation				
3.6 Full packet capture				
3.7 Secure data store				
3.8 Duplicate suppression				
3.9 User tunable controls				
4.0 Response Capabilities				
4.1 Automated policy-based response				
4.2 Alerting (SNMP, e-mail, console log)				
4.3 Session termination				
4.4 User-defined response actions				
4.5 Traffic recording and playback				
4.6 Remote threat tracing				
4.7 Peer network event notification				
4.8 Session blocking suggestions or integration				
5.0 Performance/Scalability				
5.1 Full 100 Mbps throughput (no packet loss)				
5.2 Full 1 Gbps throughput (no packet loss)				
5.3 Multiple 100 Mbps segment throughput (no packet loss)				
5.4 Handle 500,000 simultaneous TCP sessions				
5.5 Scales to hundreds of sensors				
5.6 Robust under edge conditions				

EXHIBIT 155.2 Modern IDS Capability Comparison (continued)

6.0 High Availability				
6.1 Automatic failover and failback				
6.2 High-speed failover				
6.3 "Five nines" (99.999 percent) reliability				
6.4 Cost-effective high-availability deployment configurations				
7.0 Management				
7.1 Secure remote management				
7.2 Broad platform support for management				
7.3 Scalable information presentation				
7.4 Incident drill-down capability				
7.5 Additional reference data provided (CVE, BUGTRAQ, etc.)				
7.6 Cluster administration support				
7.7 Incident annotation/auditing				
8.0 Deployment				
8.1 Multiple interface support (Gigabit and Fast Ethernet)				
8.2 Sensor roaming in switched networks				
8.3 Easy to deploy and install				
8.4 Nonintrusive deployment (noninline)				
8.5 VLAN-aware detection				
8.6 Minimal training requirements				
9.0 Reporting				
9.1 Integrated deep drill-down console reporting				
9.2 Web-based reporting				
9.3 SQL export				
10.0 Hardware Requirements				
10.1 Multiple sensors per unit				
10.2 Multi-processor scalable				

Moral of the Story

We will go back to the story for a moment. The bunny entered a traditional, two-story house even though the doors were locked. It managed to get past the intrusion prevention system: a dog, which was deployed at the foot of the stairs that lead to the bedrooms. This is a dog that normally enjoys chasing small bunnies all over the backyard because they are invading his territory. The bunny climbed a flight of stairs, and found its way down the hall and into the master bedroom. It even managed to get past another line of defense, a very territorial house cat. The defense-in-depth failed in this case.

Doing a post mortem on the event, it would be necessary to consider the experience of a near-complete breach of security: getting past three layers of defense. Fortunately, the last defense — the owner — was able to react on the internal incident response procedure and take the appropriate actions to minimize damage and manage the risk. That action kept this attack from damaging property or causing terror for the inhabitants.

The moral of the story is that several layers of defense are not always adequate regardless of how technically advanced or how cost effective they may be. A plan or procedure defined in advance with proper testing that can be quickly put into motion might be the last defense to protect the organization's assets. Any organization that does business using the Internet or private wide-area communications networks should have a security incident response program set up before an incident occurs. Having just access control, monitoring, and intrusion detection or prevention are not enough.

Note

1. Definitions come from the NSA's *Glossary of Terms Used in Security and Intrusion Detection.*

156

Software Forensics

Robert M. Slade, CISSP

Introduction and Definitions

Software, and particularly malicious software, has traditionally been viewed in terms of a tool for the attacker. The only value that has been apparent in the study of such software is in regard to protection against malicious code. However, experience in the virus research field, and more recent studies in detecting plagiarism, indicates that we can obtain evidence of intention as well as cultural and individual identity from examination of software itself.

Computer forensics is primarily seen in terms of the recovery of data and its preservation for presentation as evidence from computers that may have been used in the commission of some criminal activity. This restriction of the field to data recovery, and occasionally decryption, has been so complete that a new term has been coined to describe the more generic area of evidence from all forms of computer activity: *digital forensics.*

Aside from the data-recovery activity of computer forensics, network forensics is another major and growing field of digital forensics, and involves analysis of data from network logs and activity. A company can use network forensic analysis to detect intrusions or attacks launched against its network, generally from the Internet. This type of evidence can also be used to trace and track attackers or criminals who are using public systems such as the Internet.

Outside of the virus research community, forensic programming is a little-known field. It involves the analysis of program code, generally object or machine language code, to make a determination of, or provide evidence for, the intent or authorship of a program.

In the case of viruses, object code was usually all that was available. Even so, researchers were often able to determine a lot about a given piece of code. The first conclusion to be pursued was whether or not the program was malicious, and whether or not it was a virus. The next obvious question is to ascertain who wrote the piece of malware. Sometimes virus writers made this an easy task, including names, addresses, and, in one case, ham radio license call-sign letters. However, it was also possible to find out whether one virus was modified from another, which came first, whether the modified version was created by the same author as the original, or whether someone else had used the precursor as a template. Researchers were often able to determine whether the programmer of a piece of software was a member of a specific linguistic, national, ethnic, cultural, or age group, as well as the influence of various schools of programming.

Software forensics, a relatively new addition to digital forensics, is the broader extension of this work. Software forensics involves the analysis of evidence from program code itself. Program code can be reviewed for evidence of activity, function, and intention, as well as evidence of authorship. The technology has a number of possible uses. In analyzing software suspected of being malicious it can be used to determine whether a problem is a result of carelessness, or was deliberately introduced as a payload. Information can be obtained about authorship and the culture behind a given programmer, and the sequence in which related programs were written. This can be used to provide evidence about a suspected author of a program, or to determine intellectual property issues. The techniques behind software forensics can sometimes also be used to recover source code that has been lost.

Two different types of code, source and object, are the commodities for software forensic study. There is source code, which is relatively legible to people. Analysis of source code is often referred to as code analysis, and is closely related to literary analysis. Analysis of object, or machine, code is generally referred to as forensic programming.

Literary analysis has contributed much to code analysis, and is an older and more mature field. It is variously referred to as authorship analysis, stylistics, stylometry, forensic linguistics, or forensic stylistics.

Objectives and Objects of Software Forensics

Historically, the virus research community has used forensic programming for a variety of purposes. First and foremost, of course, was to determine the intent of a program. Was this code actually a virus, a Trojan, or other piece of malware, or had someone merely blamed it for some unrelated event? Various methods are used for this type of assessment, ranging from "black-box" execution of the program to disassembly and decompilation.

Commonly in virus research an attempt is made to determine whether a virus exists in other versions, or belongs to an existing family of viruses. Indications can be found in a direct analysis of the object code, analysis of a disassembly, or a review of text strings and messages that may be found in the code. With slight modifications of code, where only text, specific triggers, or minor functions are changed, the program might be considered a variant, identified with the original name, usually with an additional numeric or letter code. If structural changes have been made or new functions added, a virus may be assigned its own name, but noted to be part of a specific family.

In regard to families, an attempt is generally made to sequence the different variants. Even when a single author is involved, an analysis of the code can determine how the program developed. If we have a sequence of programs from a single author, we can potentially glean even more information that might help us identify the programmer.

Identity

This brings us to another objective for forensic programming and software forensics. From the earliest appearances of Trojan horse programs on bulletin boards there has been an interest in finding the authors of malicious software. In some cases, viruses have contained names, addresses, company names, e-mail addresses, Web sites, and even ham radio license identifiers, either in plaintext or in various forms of encryption. Other information can be obtained from hints in the code that indicate that two programs were written by the same author, or that an author is a member of a group. As well, stylistic or stylometric analysis of messages and text may provide information and evidence that can be used for identification or confirmation of identity.

Individual Identification

I recently spoke at a conference where the section in which I was presenting was titled "Electronic Fingerprints." The term *electronic fingerprint* is particularly well chosen in regard to identification of individuals from this type of analysis. Physical fingerprint evidence frequently does not help us identify a perpetrator in terms of finding the person once we have a fingerprint. However, a fingerprint can confirm an identity, or place a person at the scene of a crime, once we have a suspect. In the same way, the evidence we gather from analyzing the text of a message or a body of messages may help to confirm that a given individual or suspect is the person who created the fraudulent postings. Both the content and the syntactical structure of text can provide evidence that relates to an individual.

Programmers have styles in the same way that writers have styles. Code may be sloppy or optimized, and if optimized, may conserve either processor cycles or memory space. Programmers will have preferences for lookup tables or algorithmic methods and for different types of loop structures. There will be other characteristics that programmers use, either consciously or unconsciously. Taken together, these can be used to compare a sample of program code to a body of such code that a programmer is known to have produced.

Group Identification

Some of the evidence that we discover may not relate to an individual. Some information may relate to a group of people who work together, influence each other, or are influenced from a single outside source. This data

can still be of use to us, in that it provides us with clues in regard to a group with which the author may be associated, and may be helpful in building a profile of the writer.

Groups may also use common tools. One area we need to investigate in regard to program code involves programming environments that may generate or partially generate code for the programmer. Compilers also have specific signatures, sometimes in a header area of the program, and sometimes in terms of the translation of source code into object, or optimization provided by the compiler program itself. Other types of tools, such as text editors or databases, may be commonly used by groups and provide similar evidence.

In software analysis, one can find indications of languages, certain compilers, and other development tools. Compilers leave definite traces in programs, and can be specifically identified. Languages leave evidence in the types of functions and structures supported. Other types of software development tools may contribute to the structural architecture of the program or the regularity and reuse of modules.

It is possible to trace indications of cultures and styles in programming. To those unfamiliar with programming, it may seem very strange to talk about cultures in programming. However, you do not have to be around computers for too long before you realize that there are very definite communities, or trails of influence, involved in the development of programs and systems.

This is not as evident, perhaps, as it used to be. It is ironic to note that the availability of different kinds of programs is less nowadays than it was, for example, in the mid-1980s. During the 1980s, I used approximately 40 different word processors in different situations. During the 1990s, I probably used four. Therefore, computer users formerly had much more of a chance to see different programs in operation, and see different types of approaches to essentially the same problem or issue.

A very broad example is the difference between design of programs in the Microsoft Windows environment and the UNIX environment. Windows programs tend to be large and monolithic, with the most complete set of functions possible built into the main program, large central program files, and calls to related application function libraries. UNIX programs tend to be individually small, with calls to a number of single-function utilities.

One source of indicators of cultural styles is the existence or absence of functions in the program itself. For example, a practically universal function in word processors used to be something called boilerplate. This was the ability to have a standard set of text, possibly a variety of frequently used paragraphs, and to import this text into the appropriate place in the document you were creating. The function was not always called boilerplate, but it was always there. Oddly, this function does not seem to exist in Microsoft's flagship word processor, Word. Of course, it is always possible to open another Word window, open the file that you want to get text from, select the text that you want, cut or copy the text, close the second window or switch to the first, move to the position that you want the text to occupy, and then paste the text, but that does seem to be a rather involved process for such a simple function. (There is also the AutoText function, but it is more generally associated with formatting styles.)

The absence of a boilerplate function, therefore, tells us that the original developers of Word were not thoroughly familiar with a variety of standard word processors, or at least were not familiar with actual word processing operations. In addition, if we then find another word processor that does not have a boilerplate function, we know that there is a very strong probability that the developers are primarily or strongly influenced by Word.

There are, of course, numerous examples of cultural influences in programming that are visible in user interfaces. It was fairly obvious to note the bias toward LISP programming that was evident in the Logo programming language, and the predisposition toward the UCSD P-system editor that clearly drove the developers of Wordstar (among others). This was more evident in the past: the dominance of the Microsoft Windows interface has tended to homogenize interface choices. Yet even this can be seen as a cultural artifact: the inclination towards the Windows (originally the IBM Common User Access) interface has become so commanding that developers go to extraordinary lengths to include File, Edit, View, and Tools menus on programs that have no need for those kinds of functions.

Cultures of programming and design are clearly evident in malware. As a simplistic example, the early distributed denial-of-service (DDoS) tools could simply have opened a characteristic port: the author could then identify likely hosts by scanning for that particular port number. Almost all DDoS agent programs, however, were designed to "announce" availability once a machine had been compromised. The announcement could have been made through a variety of channels, and even anonymous ones, but IRC (Internet Relay Chat) was the one most commonly used. Again, DDoS client or agent programs (commonly called "zombies") could

have been commanded by having them "listen" for commands on IRC or Usenet newsgroups, but the authors all preferred to have the attack controller send attack commands directly to the agents.

In some cases, of course, similarity of code or design does not indicate influence as much as direct copying. Virus variants, for example, tend to be related merely because a virus "author" will simply take an existing virus and make minor variations to the code. (In many cases, the code is not changed at all; the new "programmer" will modify text strings, or will throw "no operation" [NOP] codes into the program — making no functional change.) Yet it is also possible to see where ideas and functions have been taken from one or more sources and added to a program. This is especially clear when the function is coded in a slightly different way.

Evidence of cultural influences exists right down to the machine-code level. Those who work with assembler and machine code know that a given function can be coded in a variety of ways, and that there may be a number of algorithms to accomplish the same end. It is possible, for example, to note whether the programming was intended to accomplish the task in a minimum amount of memory space ("tight" code), a minimum number of machine cycles (high performance code) — or a minimal effort on the part of the programmer (sloppy code).

The Blackhat Community: Hackers, Crackers, Phreaks, and Other Doodz

It is not part of the scope of this chapter to describe the blackhat community in general. (I use the term "blackhat" to avoid arguments about the "true" definition of a "hacker.") However, it is instructive to look at the rough ideas we have been able to obtain about the groups of intruders and writers of malicious software. For this information we are all indebted to researchers such as Sarah Gordon, Dorothy Denning, Ray Kaplan, and more recently, the members of the Honeynet Project.

I must also admit, at the outset, that whenever you deal with people there will always be exceptions. There are those who seem to pursue security breaking from motives which are, if not exactly admirable, at least untainted by thoughts of commerce or attention. There are also those who come up with one or two original ideas and experiment with them. Particularly in doing forensic analysis, we need to beware of falling into mental traps occasioned by our own "profiles" of the adversary. However, as with almost any stereotypes, there are reasons for the characterizations presented here.

First, I should point out that the blackhat community is extremely fragmented. Not only are there different groups, often at odds with each other, but the types of activities also differ. There are those who are trying to break into or intrude upon computer systems or networks. Others specialize in gaining unauthorized use of telephone switches and systems, frequently for the purpose of obtaining or even reselling phone service. Some are primarily interested in damaging or corrupting files, particularly in public ways, such as defacing Web sites. A great many of the blackhats in general, and probably the largest majority, really have very little idea of the technology that they are using, having obtained packaged programs or scripts, and operating them without really understanding the functions or situations appropriate for their use. Those who create programs of any type are actually relatively rare. A number do make slight modifications to the creations of others, usually functionally insignificant changes to viruses, which are widely available because of their reproductive function. There are, of course, those who are primarily interested in making illegal copies of commercial software. And, at every level, there are those who "wannabe" more respected in the blackhat community, but lack even those skills.

It may be important to examine the commonly presented justifications for blackhat activity. There are two reasons for this study. First, this examination does demonstrate something of the mindset and philosophy of the members of the community, and such a philosophy can sometimes be evident in programming style. The second reason is that some of these justifications may be presented, quite seriously, as arguments against the activity of software forensics in general.

One of the most frequently attempted justifications of blackhat activity of all kinds is that it is protected under the concept of freedom of speech. Leaving aside the issue of whether free speech is a universal right, and also ignoring for the moment that most blackhat activity does not involve programming, we still have to ask whether programming is or is not speech. Speech generally does not involve other people, and when it does, such as in the case of yelling "Fire!" in a crowded theatre or producing hate propaganda, it often is not protected. In the case where the blackhat individual is not the author of the software, such as where attack scripts are being utilized or preexisting viruses are being released, the protection of free speech is even more tenuous.

A second bid at vindication of security breaking activities is simply "because we can." Although the shallowness of this argument tends to prompt a sarcastic response from security or law enforcement personnel, we should note that the prevalence of this reasoning does make a very strong point about the anarchic nature and mindset of the blackhat community.

Many individuals who practice system violation activity explain themselves on the basis that they are following in the footsteps of the old-time hackers, who explored and discovered the capabilities of early computing devices; this flies in the face of the reality of the current level of blackhat endeavors. The few instances that are not absolutely repetitive are generally slavishly derivative. Even if we ignore the fact that most "cracking" exercises amount to no more than "knocking on doors," we still have to ask what the objective of these explorations is, which usually cannot be clearly articulated, and look at the eventual result, which, to date, has not been anything significant.

Yet another justification for blackhat activities is stated to be educational. As one who has been involved in education and training as well as reviewing for a great many years, I would be very sympathetic to this argument — if it had any basis. Even considering *2600 Magazine,* which can most charitably be described as the best of a bad lot, one is hard pressed to say anything positive about the writing quality, research, originality, or even such basics as sticking to the topic. When one turns to *phrack, 40Hex,* and the myriad others of the "'zine" ilk, the caliber runs steadily downhill. Even articles dealing with simple penetration testing generally state only that systems are weak (we already knew that, thanks), and say nothing about strengthening them.

General Characteristics

Blackhats, particularly writers of malware and viruses, tend to be young and almost invariably male. Despite occasional speculations on the addictive nature of "hacking," they usually "grow out" of the virus-writing game after a few years.

Virus and malware researchers tend to be dismissive of the technical abilities of virus writers. There exist virus writers who write competent code; there are many more who do not. The general public and the media, of course, continue to be fascinated by the image of the mythical boy genius running rings round the authorities. The blackhats like this cliché, too, and many go to some lengths to encourage the stereotype, whether or not they believe in it.

Most of today's malware programmers gain access to a victim system by tricking the victim into executing malicious code.

Malware writers do not understand or prefer not to think about the consequences for other people, or they simply do not care. Recently one researcher has speculated on the characteristics of the blackhat community in comparison to people who fall somewhere in the range between an admittedly ill-defined "normal" and those suffering from full-blown autism. Austistic individuals tend to perceive and interpret the world in an idiosyncratic manner.

Malware authors draw a false distinction between creating malicious software and distributing it. They eschew any responsibility for the damage caused by their creations. In particular, it is the responsibility of the victim to defend himself from encroaching malware, not the responsibility of the creators to keep their handiwork away from systems other than their own. Targets and victims of attacks are typically dehumanized in blackhat writings, described as losers who do not deserve to own a computer. There is also projection and displacement of guilt, frequently expressed in terms justifying security breaking activities because vendor X makes lousy software or large corporations are doing bad things.

In self-reports from blackhats, a number of aspects are reported to be part of the thrill, including the act of vandalism itself, fighting authority, "matching wits" with the security or law enforcement communities, aggression (often arising out of resentment and reinforced by the feeling of safety and power that is engendered by apparent anonymity), the ability to induce fear and panic in the media and the general public, and the "15 minutes of fame" as well as the recognition of peers. Malware writers tend to feel marginalized and unrecognized in normal society, so they feel a very strong sense of identity with the blackhat tribe.

Blackhat Products

Most of the end result of blackhat activity consists of compromised systems, defaced Web pages, and pointlessly consumed bandwidth. Overall, this might be of interest to people investigating network forensics, but is not of much use for us in software forensics. However, attack tools, DDoS kits, Trojans, viruses, worms, remote access Trojans (RATs), and other forms of malware are.

We will, of course, want to find out as much as possible about what the specific piece of malware does. We also want to find about the author, if we possibly can. Knowing about the broad classes of malicious software can help point out, in general outline, the functions to look for. Knowing the class of malware may also help us to identify the author, because blackhats tend to be just as specialized as any other type of programmer.

Malicious Software

It is sometimes hard to make a hard and fast distinction between malware and bugs. For example, if a programmer left a buffer overflow in a system and it creates a loophole that can be used as a backdoor or a maintenance hook, did he do it deliberately? This question cannot be answered technically, although we might be able to guess at it, given the relative ease of use of a given vulnerability. However, there is general agreement that the following types of software do fall into the malware category.

Trojans

Trojans, or Trojan horse programs, may be the largest and most diverse class of malware. A Trojan is a program that pretends to do one thing while performing another, unwanted action. The extent of the pretense may vary greatly. Many of the early Trojans relied merely on the file name and a description on a bulletin board. "Log-in" Trojans, popular among university student mainframe users, mimicked the screen display and the prompts of the normal log-in program and could, in fact, pass the user name and password along to the valid log-in program at the same time as they stole the user data. Some Trojans may contain actual code that does what it is supposed to be doing while performing additional nasty acts that it does not tell you about.

Given the absence of a specific functional requirement for Trojans, and the variety of types of pretense and social engineering that can be used, Trojan programs can be created or modified, sometimes based on widely available utility software, by relatively unskilled people. In addition, Trojans tend to be reused and passed around within the blackhat community. Therefore, the use of software forensic techniques may be of limited use, until methods are refined further.

Logic Bombs

A logic bomb is generally implanted in or coded as part of an application under development or maintenance. Unlike a RAT or Trojan it is difficult to implant a logic bomb after the fact, unless it is during program maintenance.

A Trojan or a virus may contain a logic bomb as part of its payload.

Because of the inclusion of the logic bomb code with the code of the main program, software forensics may be used to determine whether the bomb was introduced by the author of the application or programmed by someone else. (This does not, of course, preclude the possibility that the programmer introduced code written by someone else into his own program.)

A similar situation applies with respect to a backdoor (sometimes called a trap door), a hidden software or hardware mechanism that can be triggered to permit system protection mechanisms to be circumvented. The function will generally provide unusually high or even full access to the system either without an account or from a normally restricted account. It is activated in some innocent-appearing manner; for example, a key sequence at a terminal. Software developers often introduce backdoors in their code to enable them to reenter the system and perform certain functions; this is known as a "maintenance hook." The backdoor is sometimes left in a fully developed system either by design or accident.

Backdoors can also be introduced into software by poor programming practices, such as the infamous buffer overflow error.

DDoS Agents

DDoS (distributed denial of service) is a modified denial-of-service attack, which does not attempt to destroy or corrupt data, but attempts to use up a computing resource to the point where normal work cannot proceed. The structure of a DDoS attack requires a master computer to control the attack, a target of the attack, and a number of computers in the middle that the master computer uses to generate the attack. These computers

between the master and the target are variously called "agents" or "clients," but are usually referred to as running zombie programs. Although zombie is the most widely used term, it is used somewhat indiscriminately, and it is probably most proper to refer to DDoS agent software.

Unfortunately, DDoS kits are widely available, in ready-to-use form, and in some cases the authors are known. Software forensics has relatively little to contribute in terms of those who actually set up such DDoS networks and attacks, and the developers of DDoS agent software are not reticent about admitting authorship. Network forensics is probably more use in determining who launched an attack.

RATs (Remote Access Trojans)

To convey a sense of legitimacy, the authors of remote access Trojans (RATs) would generally like to see them referred to as remote administration tools.

All networking software can, in a sense, be considered remote access tools: we have file transfer sites and clients, World Wide Web servers and browsers, and terminal emulation software that allows a microcomputer user to log on to a distant computer and use it as if he was on-site. The RATs considered to be in the malware camp tend to fall somewhere in the middle of the spectrum. Once a client, such as Back Orifice, Netbus, Bionet, or SubSeven, is installed on the target computer, the controlling computer is able to obtain information about the target computer. The master computer will be able to download files from, and upload files to, the target. The control computer will also be able to submit commands to the victim, which basically allows the distant operator to do pretty much anything to the prey. One other function is quite important: all of this activity goes on without any alert being given to the owner or operator of the targeted computer.

When a RAT program has been run on a computer, it will install itself in such a way as to be active every time the computer is turned on after that. Information is sent back to the controlling computer (sometimes via an anonymous channel such as IRC) noting that the system is active. The user of the command computer is now able to explore the target, escalate access to other resources, and install other software, such as DDoS zombies, if so desired.

Rootkits, containing software that can subvert or replace normal operating system software, have been around for some time. RATs differ from rootkits in that a working account must be either subverted or created on the target computer to use a rootkit. RATs, once installed by a virus or Trojan, do not require access to an account.

As with DDoS agents, both RATs and rootkits tend to be available and are not commonly rewritten for an attack. However, the authors of this software have not always been identified, so software forensics may be used to identify authors, although these may not be the actual attackers.

Other Objects of Study

In the case of malware, we are primarily concerned with finding out what the program does and who wrote it. Frequently this might be a concern with ordinary application software. Generally speaking, with regular software we will know one, but need to find out the other.

One situation that may arise requiring a forensic determination is in the case of intellectual property. There is the possibility of multiple claims of authorship of a particular piece of software. It should be relatively easy to compare a specific program against two or more known bodies of work, and ascertain which of a number of authors has written the disputed application. In the case of multiple authorship of a single program or module this would be more difficult, but forensic linguistics has, in some cases, been able to distinguish between multiple authors, even down to the level of an individual sentence. In any case, we should be able to conclude whether multiple authors were involved fairly easily. Plagiarism detection is already well established as a technology, and there are a number of automated tools that can help us in this regard.

The technologies used in software forensics have uses in software development itself, and, indeed, some of them originated there. For years, reverse engineering has been a common practice in system development: software forensics performs much the same function, albeit sometimes at different levels of detail. Disassembly and decompilation tools may be able to assist in application development, for example, recovering source code for legacy systems where such has been lost over the years.

As noted, the tools used in software forensics are generally utilities employed in programming. At this point it may be worthwhile to list the instruments that can be helpful in the forensic endeavor.

Software Forensics Tools

Before listing the tools themselves, some brief background on the programming process might be in order. (It is commonly said, and attributed to Edgser W. Djkstra, that if debugging is the process of removing bugs, then programming must be the process of putting them in.)

The Programming Process

In the beginning, of course, programmers created object (or machine, or binary) files directly. The operating instructions (opcodes) for the computer and any necessary arguments or data were presented to the machine in the form that was needed to get it to process properly. Assembly language was produced to help with this process; although there is a fairly direct correspondence between the assembly mnemonics and specific opcodes, at least the assembly files are formatted in a way that is relatively easy to read, rather than being strings of hexadecimal or binary numbers.

With the advent of high- or at least higher-level languages, programming language systems split into two types. High-level languages are those where the source code is somewhat more comprehensible to people. Those who work with C or APL may dispute this assertion, of course. The much-maligned COBOL is possibly the best example: the general structure of a COBOL program should be evident from the source code, even for those not trained in the language.

Compiled languages involve two separate processes before a program is ready for execution. The application must be programmed in the source (the text or human readable) code, and then the source must be compiled into object code that the computer can understand. Those who actually do programming will know that I am radically simplifying a process that generally involves linkers and a number of other utilities, but the point is that the source code for languages like Fortran and Modula cannot be run directly; it must be compiled first. (It is, of course, dangerous to make such statements; undoubtedly some completist computer language historian will be able to identify Fortran and Modula interpreters of which I am totally unaware.)

Interpreted languages shorten the process. Once the program has been written, it can be run, with the help of the interpreter. The interpreter translates the source code into object code "on the fly," rendering it into a form that the computer can use. There is a cost in performance and speed for this convenience: compiled programs are "native" or natural for the CPU to use directly (with some mediation from the operating system), and so run considerably faster. In addition, compilers tend to perform some level of optimization on the programs, choosing the best set of functions for a given situation.

However, interpreted languages have an additional advantage: because the language is translated on the machine where the program is being run, a given interpreted program can be run on a variety of different computers, as long as an interpreter for that language is available. Scripting languages, used on a variety of platforms, are of this type. JavaScript applets, for example, may be embedded in Web pages, and then run in browsers that support the language regardless of the underlying computer architecture or operating system. (JavaScript is probably a bad example to use when talking about cross-platform operation, because a given JavaScript program may not even run on a new version of the same software company's browser, let alone one from another vendor or for another platform. But it is supposed to work across platforms.)

As with most other technologies where two options are present, there are hybrid systems that attempt to provide the best of both worlds. Java, for example, "compiles" source code into a sort of pseudo-object code called bytecode. The bytecode is then processed by the interpreter (called the Java Virtual Machine, or JVM) for the CPU to run. Because the bytecode is already fairly close to object code, the interpretation process is much faster than for other interpreted languages. Because bytecode is still undergoing an interpretation, a given Java program will run on any machine that has a JVM. (Java does have a provision for direct compilation into object code, as do a number of implementations for interpreted languages such as BASIC.

The Products

Of course, what we get for analysis depends on how the program was developed. If it was machine language programming, assembler, or a compiled language, we get an object code file for analysis. In the case of assembler or compilation we may also have a copy of the assembler or high-level language source code. If we have an interpreted language used for development, we have a copy of the source code of the program. (For the purposes of software forensics analysis, partially compiled objects such as Java bytecode can be considered to be subject

to the same type of analysis as object code. Also, source code, where available, can be assessed in the same manner regardless of whether the language used was a compiler or an interpreter.)

However, the development system still has some ways to make our analytical task more difficult.

Complicating Factors

When a program is compiled or assembled, all comments (unless they are handled in special ways) are eliminated. Comments often constitute the programmer's "notes to self" during the development process, and therefore this information is lost.

When a program is assembled or compiled, the assembly or compilation program can introduce strings and signatures into the code. Obviously, these sections of code must be identified and eliminated from consideration when we are trying to determine authorship of the program. (Occasionally in virus research, compiler-introduced strings were mistakenly taken as unique and therefore used as signature strings for scanning programs. The anti-virus scanners that used such strings would generate large numbers of false-positive alarms as the strings were found in any programs that had been compiled from those languages.)

An additional concern is that compilers frequently optimize the code in some way, and this process may eliminate or confuse some parts of the characteristic signature of a given author.

As previously noted, other utilities besides compilers may be part of the program generation process. These utilities may also introduce signatures into the code, and these signatures must be taken into account. In addition, CASE (Computer Aided Software Engineering) tools and even programming environments (such as specialized editors directly associated with compilers) can influence the design and structure of programs. On the other hand, these various characteristics and signatures, if properly identified, can help identify a programmer or group, given a record of the use of specific sets of tools.

The source code that we receive with interpreted language programs generally does contain the comments (if the author made any) and did not eliminate them before releasing the program. We usually are not faced with compiler-introduced signatures, although a number of programming environments for interpreted languages may introduce comments or bias the use of certain types of programming styles or structures. However, the major concern with interpreted source code is that, particularly in regard to viruses and other widely distributed programs, the availability of the source means that a number of people have the opportunity to make minor variations to the program. This is easy to do when you have the source code, and interpreted languages tend to be simple to use, and are therefore within the programming skill level of a much wider group.

Finally, Already, The Tools

The first tool used in forensic research is obvious enough that most ignore it: a computer. I am not merely being sarcastic at this point. A great deal of information can be obtained by noting the behavior and operation of the program under study when it is running. First of all, we can directly observe what the program does, in gross terms, as it runs. Then we can perform more detailed or low-level studies: are attempts made to access specific areas of memory? Are calls being made to specific resources? Are attempts being made to contact other computers via a network, particularly the Internet? Then again, we can attempt to treat the program like a black box, and see what happens when we prod at it in various ways. (Of course, when dealing with malware, it is important to take precautions: if the first thing the program tries to do is to overwrite the hard disk, the information obtained can be limited.)

The next tool is the good old-fashioned hex editor. Used for displaying the content of binary files (in hexadecimal format, and usually also with those bytes that could be displayed in ASCII running parallel down the side), hex editors can help us find a number of interesting items that might be in the code.

The first items to look for are any strings of actual text. There tends to be a lot of text in programs. Some strings may be text that might appear as messages on the screen. Obviously, any program that contains a string stating "ha ha luzer i just blue up yer d!sc" probably warrants further study. When dealing with malware, as strange as it may seem, the authors of malware are often very proud of their creations, and may also include copyright notices, instructions for use, and even personally identifiable information about themselves.

Another set of strings that may appear as text in programs are application programming interface (API) calls. Particularly in Windows-based software, API calls can be very common. Even if you are not familiar with the libraries being used, APIs generally have very explanatory names. If, for example, you view the code for something that is supposed to be a game, APIs that indicate calls to close, open, or monitor network ports would be somewhat suspicious. An additional class of identifiable information might be available here: if calls

are made to contact entities on the Internet, we may find URLs (Uniform Resource Locators) or even e-mail addresses.

As well as API calls, we may be able to recognize some function calls, although this takes a bit more practice. Programs use some printable characters (in fact, for Intel CPUs it is quite possible to write programs using only printable characters), and some functions can be recognized by a particular string of ASCII characters. For example, in the old days of MS-DOS viruses, the string "PSQR" was one to watch for. It was related to a call by the program to "terminate and stay resident." Because few programs in those days needed to "go resident," such a call was an indication to look deeper.

Text strings may not appear in the program. In some cases, there may be no need for any. In other situations, malware authors may use simple forms of encryption to obfuscate messages. Generally the encryption takes the form of a simple byte-by-byte XOR with a given byte value: for some reason 2Fh seems to be quite popular. Cryptanalysis appropriate for simple substitution ciphers should be able to recover these text passages.

As there are assemblers and compilers for turning assembly and high-level languages into object code, so there are disassemblers and decompilers that do the reverse. Disassembly is easier than decompilation. However, note that disassemblers do not deal well with sections of text or data: they try and interpret the material as program code, with rather random results. In addition, malware authors also frequently encrypt sections of the code, specifically to frustrate attempts at disassembly. In this case one must find the decryption routine, which must come prior to the encrypted section in linear programming, and then use that to decrypt the material before disassembly takes place.

Decompilers fare rather worse, and are a less-mature technology in any case. Decompliers generally require assembly rather than object code as input, and usually do better if the language and even version of the original compiler can be determined. Decompilation is seldom fully successful, and most likely will produce some source code interspersed with sections of assembly code.

Another tool to use is a debugger, although the ones used in forensic programming differ from those used in high-level programming. Debuggers used in software forensics need to be able to control the execution of another program. Therefore, they need to act as a kind of software in-circuit emulator, allowing one operation at a time to proceed. The debugger should also have the ability to determine and display changes in memory and the CPU registers. The venerable DEBUG, from MS-DOS systems, is able to perform a number of these functions, albeit in a very limited way with large programs, as well as functioning as a hex and sector editor and a disassembler.

Software Forensic Technologies and Practices

There are a variety of ways of looking at code to obtain information and evidence. The most obvious, of course, is to look for text, functions, and other items in the content of the software. However, there are ways, not initially apparent, of looking for patterns that are independent of the content of the program.

Content Analysis

Recently my wife drew my attention to very similar embroidery charts, one in a book and another in a handout given away by a floss company. The composition, proportions, and detailed parts of the images were identical. The obvious conclusion is that either one copied the other, or that both were copied from a third source. If this situation were to be examined in terms of intellectual property, in the absence of evidence of a third source, we could note that the window framing in the free pattern is more complex, and that the free pattern has additional shading around the window. This would tend to indicate that the free pattern had been copied or modified from the book, because copiers tend to embellish rather than simplify.

This is an example of content analysis. The charts provide a specific representation of an idea, in this case presented in graphical form. In the same way, an idea presented in text or program code will have a certain representation, composition, and structure. The way that authors and programmers present ideas tends to be characteristic, both in terms of overall composition and in terms of details such as vocabulary, phrasing, function, and structure.

In addition, we can use analysis of text and code to find sequences of messages, and trace influences. In material that is copied from an original, the overall structure and composition tends to be unchanged, but details and embellishments tend to be added.

The syntax of text tends to be characteristic. Does the author always use simple sentences? Always use compound sentences? Have a specific preference when a mix of forms is used? Syntactical patterns have been used in programs that detect plagiarism in written papers. The same kind of analysis can be applied to source code for programs, finding identity in the overall structure even when functional units are not considered. A number of such plagiarism detection programs are available, and the methods that they use can assist with this type of forensic study.

Of course, when considering the content of the text, most people consider characteristic use of vocabulary and phrases. This does tend to be effective, but it usually relies on having a large set of samples to analyze. We also generally have to ensure that the texts cover the same or similar subjects to avoid problems with disparate vocabularies in differing fields. Similar analysis can be applied to programs, using functional structures that provide analogues of vocabulary, and assessing modules in the same way we read paragraphs.

It may seem strange to those who do not regularly work with object code to find that there can be characteristic "phrases" in these strings of 1s and 0s. As one example, virus researchers would frequently find strings or patterns that would indicate attempts to perform certain functions. In the days of MS-DOS (or PC-DOS or DR-DOS), a program that was making a call to remain in memory while other programs were running (or "go resident") was unusual, and frequently a sign that the program was viral. When viewed with a text or hex editor, most such programs would contain the string of letters "PQSR." The letters did not mean anything as text, they were simply the common pattern of the operating codes making the "terminate and stay resident" function call. However, this pattern was a characteristic indicator, a kind of vocabulary of a certain type of activity.

Error Analysis

Errors in the material can be extremely helpful in our analysis, and should be identified for further study. In some of my early work published on the history of computer viruses, I made a mistake in the spelling of the name of one person involved in the creation of a specific program. Shortly thereafter, another person also published such a history. The histories were very similar, but that could be expected if two people both had access to the same sources. However, the second history also contained the error that I had made. The author of the second history, had he followed original reference materials, would not have made that error, thus indicating that the later text was a copy of my original.

In another example, two students cheated and copied answers on a test I gave at a college. In my report on the incident, I included a statistical analysis on the results. The likelihood of two students getting exactly the same questions correct was extremely small. But the chance of two students making exactly the same errors was five times smaller, making a much stronger case for the cheating presentation.

The existence of errors in program code is problematic, because certain types of mistakes will ensure that the program either does not compile or does not run. However, we may find that certain types of nonfatal errors; such as a failure to optimize various types of operations, may be characteristic of an individual or group.

Noncontent Analysis

At one point my wife worked as a secretary in a government office, typing reports for a variety of officers. She noted that different writers had different styles. In those days of typewriters and monospaced fonts, one of the factors she discovered was that different people required different line lengths. If the wrong length was used, the report turned out to have a ragged edge down the right side of the page. If the right line length was used, the margin was neater. The line length was characteristic of the writer: Tom needed a 65 space line, Dick used 72, and Harriet required 68 spaces. This characteristic is consistent over time: Harriet will always need 68 spaces. This may seem to be a trivial characteristic, but it does indicate that a number of identifying attributes are available in order to build an electronic fingerprint of text. The same is true of program code.

A specific method of finding such characteristics in text is Cusum, explained in the book, *Analysing for Authorship,* by Jill M. Farringdon. Literary critics are quite used to talking about how an author like Henry James would write enormously long sentences that, in more modern writings, would be split into smaller, more digestible chunks, but which were, in the days when it was considered acceptable for someone like Marcel Proust to write an entire book that was one long sentence, the norm that was to be emulated and adopted. Others wrote differently. Hemingway, for example. Short sentences. Sentence fragments, really. Therefore, critics are quite used to making decisions about authorship based on numeric metrics.

Cusum (or QSUM, the two terms seem to be used interchangeably in the book) is such a technique. Instead of looking at meanings or characteristic turns of phrase, the method looks at combinations of statistical patterns in writing, patterns that the writer is probably unaware of using.

It may seem strange to use meaningless features as evidence. However, Richard Forsyth reported on studies and experiments that found short substrings of letter sequences can be effective in identifying authors of textual material. Even a relative count of the use of single letters can be characteristic of authors. Similar measures can probably be applied to program code, both source and object.

Additional Noncontent Indicators

Certain message formats may provide us with additional information. A number of Microsoft e-mail systems include a data block with every message that is sent. To most readers, this block contains meaningless garbage. However, it may include a variety of information, such as part of the structure of the file system on the sender's machine, the sender's registered identity, programs in use, etc. In the case of material distributed by e-mail, this information may be available.

Other programs may add information that can be used. Microsoft Word, for example, is frequently used to create documents sent by e-mail. Word documents include information about file system structure, the author's name (and possibly company), and a "global user ID." This ID was analyzed as evidence in the case of the Melissa virus. MS Word can provide us with even more data: comments and "deleted" sections of text may be retained in Word files, and simply marked as hidden to prevent them from being displayed. Basic utility tools can recover this information from the file itself. Some compilers create similar tables on data within the executable body of a program.

Legal Considerations

First of all, because this section deals with legal issues, I imagine that I need to make legal disclaimer-type noises. Therefore, be it known to all men by these presents that I am not a lawyer, I have never even played one on TV, this is not to be considered legal advice, for legal advice please see qualified legal counsel, void where prohibited by law, no warranty express or implied is made on the fitness of this information for any purpose including the purpose for which it was intended, no added salt, your mileage may vary, this product contains not less than 70 percent recycled opinions, please do not read while operating heavy machinery, have I missed anything?

There are going to be differences in the permissibility of software forensics evidence depending on the legal system that has jurisdiction over the crime. Admissibility of computer records may vary from system to system: some legal systems will consider it hearsay and require higher standards to accept it. Jurisdiction, as with any situation that deals with possible network involvement, may be a problem as well.

With respect to jurisdiction, of course, what may be considered a crime in one location may not be in another. Canadian law, for example, notes that anyone who, without authorization, modifies data or "causes" it to be modified, is guilty of an offense. Therefore, if we can demonstrate through software forensics that the intent of the program was to create data modification (possibly among other things) and to gain access to systems without active user involvement, then we have a case with regard to computer viruses. In addition, if we can reveal a link to a specific individual as the author of the program, we can make a case against that person. Other jurisdictions may not have the same wording in law, and so we may not be able to prosecute certain types of activity. In regard to the use of software forensics with respect to intellectual property cases, note that a number of countries do not have intellectual property laws.

In dealing with legal issues, we have an immediate problem in that not only do different countries have different laws, but possibly even different legal systems. Those from Britain, the Commonwealth countries, and the United States will be most familiar with the "Common Law" system, based on the presumption of laws that uphold the common good, from an originating charter document and case law precedents laid down over the years. (Common Law is also the system under which a suspected criminal is presumed to be "innocent until proven guilty.") In some of those countries there are specific laws that would make, for example, malicious software illegal. However, most Common Law systems also have provisions against mischief or vandalism, so malicious software could probably be prosecuted even in the absence of a specific law. (Successful prosecution is quite another matter, the requirements for which we will be dealing with at length.)

Some countries, such as France, have Code Law or Civil Law systems. Under these systems, an activity is not illegal unless there is a specific law against the activity. (In access control terms, everything is permitted

unless it is forbidden.) Therefore, under such systems, it may be perfectly legal to write and distribute malicious software (or break into computer systems, or sell pirated copies of copyright protected software) simply because the law against it does not exist: the lawmakers have not caught up with the times.

As this chapter was being written (early 2003), the legal situation with regard to software forensics, particularly in the United States, was very confused. Certain laws intended for the protection of intellectual property could be used to prevent the examination of software. There was the case of a programmer from Russia who came to the United States to speak to a conference about a weakness in a security mechanism in a commercial software product. He was, in fact, arrested and held in custody. It may be possible that authors of malicious software may challenge software forensics evidence on the basis that they hold copyright on their software, and did not grant permission for the software to be examined.

Presentation in Court

Presentation of this kind of technical evidence in court can be problematic. Debates over DNA evidence as identification, and the acceptability of such evidence to nonspecialists, which describes most lawyers, judges, and juries, are directly relevant to this issue. The field of forensic linguistics is still developing, and experts may have to be judged individually. Findings and opinions may be dismissed by the court on the basis that the expert cannot prove sufficient knowledge, skill, experience, training, or education.

There is an additional point to be made about the difference between civil and criminal cases under Common Law systems, and it is directly relevant to forensic studies. The test of evidence and proof is not the same in the two types of cases. A criminal case must be proven "beyond a reasonable doubt." Civil cases require only that a decision be made on the balance of the probabilities. Thus evidence for a criminal trial must be presented much more carefully.

A factor in the admissibility of evidence is the concept of hearsay. As a witness in court, you may be asked to say what you did or directly witnessed. Except in very unusual circumstances, you will not be asked, and will not be allowed to say, what someone else told you that they did or saw. This "second-hand" testimony is called hearsay, and is pretty much automatically suspect. If the court is to accept evidence other than directly from the source, there has to be corroborating testimony.

Business documents are all, in fact, considered to be hearsay in some sense. This is because they are all, in a way, information around a transaction rather than direct evidence of a transaction. This is particularly true in relation to electronic data. When presenting printouts or other representations of digital information, there must be testimony about how the information was stored and handled, whether there are regular procedures, whether there was any kind of departure from regular procedures, what protections are in place to ensure the integrity of the data, who has access and the ability to change the data, etc. This has particular relevance to software forensics, because the evidence gathered may result from very minor differences, and, in addition to proving to the court that the differences are significant, we must be able to prove that the material presented in court is identical, to the bit, with the original data or code.

In choosing evidence for court, content-based analysis may seem to be a more reasonable choice for presentation, but its use may backfire. Content analysis may be "morally" convincing, but still lack specific proof and be dismissed as mere opinion. In the embroidery chart example I gave earlier, it is instantly apparent that the patterns are from the same source, but it takes time to determine specific features and reasons, and the sequence of the patterns. It is, of course, just those specific features and reasons that are important for presentation in court.

Future Work

At this point, it would be premature to draw conclusions or even suggest implications for software forensics work. Certainly there is a potential for promise in the field, but a good deal of work remains to be done. I shall, therefore, suggest only a limited number of additional studies that present themselves as needed research.

A good deal of work needs to be done in regard to building a library of resources and references for software forensics analysis. Virus research has concentrated on signatures for individual pieces of malware as well as signatures for various malware kits, encryption engines, and heuristic signatures for "dangerous" functions. These are helpful. However, we also need to collect signatures of compilers and other software creation tools.

Signatures of good, bad, or mature coding practices should also be ascertained. A fairly quick scan of a piece of code to determine the skill level of a programmer would be a good thing. Unfortunately, it is unlikely that

such can be easily codified or automated. In addition, authors of malicious software, in their quest for "3133t" status, would likely take to embedding such signatures in their code, purely as an attempt to be identified as skilled programmers.

Additional work needs to be done in terms of decompilation software. In particular, it would seem obvious that a combination of the two existing types of decompilers, those that recover function and those that recover structure, should be attempted.

The various projects aimed at detecting plagiarism would seem to be producing very useful tools. For broader application, however, it would be important to attempt to make identification out of larger populations, and to validate, statistically, the assurance we can derive from such identifications.

Resources

"Computer Forensics and Privacy," Michael A. Caloyannides, 2001 — A good resource for both data recovery and protection.

"Computer Forensics," Warren G. Kruse II and Jay G. Heiser, 2001 — Concentrates on data recovery and chain of evidence.

"Hackers: Crime in the Digital Sublime," Paul A. Taylor, 1999 — Best coverage of the phenomenon to date, though still with holes.

http://www.cerias.purdue.edu/coast/coast-library.html — Library of papers, many relating to software forensics.

http://citeseer.nj.nec.com/krsul96authorship.html — Authorship Analysis: Identifying The Author of a Program — Krsul, Spafford.

http://www.dfrws.org/, Digital Forensic Research Workshop.

http://hometown.aol.com/qsums, "Analysing for Authorship: A Guide to the Cusum Technique," Jill M. Farringdon, 1996.

http://plg.uwaterloo.ca/~migod/746/papers/bern-cloning.pdf — Detecting duplicated code.

http://www2.informatik.uni-erlangen.de/~phlipp/mypapers/jplag_jucs2001.pdf — JPLAG plagiarism detection.

http://citeseer.nj.nec.com/wise96yap.html — YAP plagiarism detection.

http://www.fbi.gov/hq/lab/fsc/backissu/april2000/swgde.htm — Scientific Working Group on Digital Evidence (SWGDE), Digital Evidence: Standards and Principles.

http://www.forensic-evidence.com/site/ID/linquistics.html — Forensic linguistics/stylistics in court: *United States v. Van Wyk*.

http://www.qucis.queensu.ca/achallc97/papers/p025.html — Short substrings in document discrimination.

http://www.badguys.org/papers.htm — Some papers on cracker/vandal culture and characteristics.

Reporting Security Breaches

James S. Tiller, CISSP

If you are involved with information systems within an organization — whether at the highest levels of technical management or the end user in a remote office — you will ultimately be faced with a security incident. Managing a security breach life cycle encompasses many managerial, technical, communication, and legal disciplines. To survive an event you need to completely understand the event and the impacts of properly measuring and investigating. When reporting an incident, the information provided will be scrutinized as it rolls up the ranks of the organization. Ultimately, as the report gains more attention and it nears the possibility of publication, the structure of the incident report and supporting information will be critical.

This chapter touches upon the definition of an incident and response concepts, but its focus is on reporting the incident. It is assumed that incident response processes, policy, mitigation, and continuity are all existing characteristics — allowing us to focus on the reporting process and escalation.

SCHROEDINGER'S CAT

A quick discussion on the value of information in the world of incidents is in order.

Quantum mechanics is an interesting code of thought that finds its way into the world of security more often than not. Erwin Schroedinger produced a paper in 1935, "Die gegenwartige Situation in der Quantenmechanik," that introduced the "Cat" and the theory of measurement. In general, a variable *has* no definite value before it is measured; then measuring it does *not* mean ascertaining the value that it *has* but rather the value it has been measured against. Using Schroedinger's example, let us assume there is a cat in a box, a black box. You open the box and the cat is dead. How do you know the cat was dead before you actually made the observation by opening the box? Opening the box could have killed it for all you

know. In the most basic terms, the interaction of variables with measurement requirements will raise the question of how much of the value obtained was associated with the act and process of measurement. Of course, Schroedinger's Cat is a theory that impacts quantum mechanics more so than measuring your waistline, but establishing control sets and clear measurement policy related to the technology is critical in the space between the ordinary and the extraordinary. This simple paradox lends itself to interesting similarities in the world of security incidents — albeit loosely.

Your actions when determining an event, or how you have set the environment for detecting an event, can have ramifications on the interpretation of the event as it is escalated and reported. How does the "cat" apply? It is necessary to measure from multiple points in various ways to properly ascertain the event when reporting as an incident.

For example, if you have an intrusion detection system (IDS) at your perimeter and another on your DMZ with an identical configuration and an anomaly is detected, you have proven an anomaly on both sides of your firewall. With information from the logs of the alleged target server and the firewall, you now have more disparate information sources to state your case and clearly ascertain the scope of the incident. Additionally, this will demonstrate the attention to clarity and comprehensiveness of the detection and documentation process, furthering the credibility of the report.

Another application of the analogy is incident response process and the actual collection of information. Although we are focusing on reporting incidents, it is important for the reader to understand the importance of the information to be shared. Collecting information in support of detailing the incident can be a sensitive process, depending on two fundamental directions decided upon at the initial onset of incident response: *proceed and protect* or *pursue and prosecute*. Care should always be practiced when collecting evidence from impacted systems, but this is most true when the decision to pursue and prosecute has been made. It is here, gathering data for future analysis, reporting, or evidence, that Schroedinger's Cat can become a lesson in forensics. Simply stated, the act of extracting data — no matter the perceived simplicity or interaction — can affect the value as well as the integrity of the information collected. Was that log entry there because you created it? Understandably, an oversimplified example, but the point is clear — every interaction with a system can inherently impact your ability to measure the incident in its purest state. Based on Schroedinger's theory, simply the act of quantifying will inevitably and unavoidably influence the measured outcome.

Understanding the consequences of data collection during and after an incident will help you to clearly detail and report an event, ultimately building efficiencies into the mitigation process.

SECURITY REQUIREMENTS

At the risk of communicating an oversimplification, it is necessary to state that proper configuration and management of security is critical. Through the use of technology and defined processes, you can accurately and confidently identify incidents within the network and quickly determine what happened and the vulnerability that was exploited.

Security Policy

Every discussion on security has a section on security policies and their importance. Security policies define the desired security posture through communicating what is expected of employees and systems as well as the processes used to maintain those systems. Security policies are inarguably the core point of any successful security program within an organization. However, with regard to incident management, the criticality of security policies cannot be understated.

Security policies provide an opportunity to understand the detailed view of security within an organization. In many cases, security policies reflect common activities practiced within the organization regularly and can be used as a training resource as well as a communication tool. However, incident response policies could be considered the most important section of any security policy, based on the criticality and uniqueness of the process combined with the simple fact that incidents are not typical occurrences (usually). In the event of a rare occurrence, no one will know exactly what to do — step by step — and in all cases a referenceable document defining what should be done in accordance with the desired security posture can be your lifeblood.

In the day-to-day activities of a nuclear plant, there is always the underlying threat of a failure or event; but it does not permeate the daily tasks — they are preparing and avoiding those events through regular management of the systems. In the rare times there is a significant occurrence, the proprietors will always reference a process checklist to assist in troubleshooting. Another example is a pilot's checklist — a systematic process that could be memorized; but if one portion is exercised out of order or missed, the result could end in disaster.

Therefore, a security policy that clearly defines the identification and classification of an event should also state the process for handling and reporting the incident. Without this significant portion of a security policy, it is almost assured the unguided response procedures will be painful and intermittent in context.

Security Technology

In the realm of digital information, security is realized and measured through technology. The configuration of that technology and the defined

interaction with other forms of technology will directly impact the ability to recognize an incident and its eventual investigation.

Security-related technology comes in many forms, ranging from firewalls and IDSs to authentication systems. Additionally, security characteristics can emerge from other technologies that are traditionally not directly associated with security and provide services beyond the envelope of information security. However, these become the tools to identify events in addition to becoming collection points for gaining information about the incident.

As briefly mentioned above, more points within a network that have the ability to detect or log events will increase the quantity of information available that can be correlated to amplify the quality and accuracy of the incident description. In addition to the number of points in the network, the type and layer with which it interacts may become the defining factor in isolating the event.

For example, a firewall may log traffic flow by collecting information about source and destination IP addresses and port numbers. Along with time stamps and various other data, the information can be used to identify certain characteristics of the incident. To obtain even more of the picture, the target operating system, located by the destination IP address from the firewall's logs, may have logs detailing certain actions on the system that are suspicious in nature and fall within the time of attack window established by the firewall's logs. The last piece of the puzzle is provided by a system-monitoring package, such as Tripwire — an application that essentially detects changes in files. Based on the information from Tripwire, it may appear that several files were changed during the time of the attack. A short search on the Internet may reveal that a Trojan version of the file is in the wild that can provide temporary administrative access using port 54321, which you have verified from the firewall and system logs. Additionally, the report continues to detail known implantation techniques to install the Trojan — replacing the valid file — by leveraging a weakness in the TCP/IP stack by sending overlapping packets that result in distorted IP headers. It was the "notification" log on the firewall that allowed you to initially determine the time frame of the attack; but without the other information, you would be hard-pressed to come to the same detailed conclusion.

The purpose of the example is to communicate the importance of disparate information points and types within the network. The firewall passed the packet because it was not denied by the rules, and the header structure fell within limits; but the vulnerability exploited in the operating system could not survive those changes. The file implantation would normally go undetected without the added information from Tripwire.

It is clear that ample information is helpful, but the variety of data can be the defining factor. Therefore, how technology is configured in your environment today can dramatically affect the ability to detect and survive an incident in the future.

Additionally, the example further demonstrates the need for incident response policies and procedures. Without a well-documented guide to follow, it is doubtful that anyone would be able to traverse the complicated landscape of technology to quickly ascertain an incident's cause, scope, and remedy.

REPORT REASONING

There are many attributes of incident management that must be considered within the subject of reporting. This section discusses:

- *Philosophy.* Simply stated, why report an incident at all? This question insinuates notifying the public, but it can be applicable for internal as well as partnership communications. What are the benefits and pitfalls of reporting an incident?
- *Audience.* When reporting anything, there must be an audience or scope of the people who will be receiving or wanting the information. It is necessary to know your constituents and the people who may have a vested interest in your technical situation.
- *Content.* As information is collected about an incident, there will certainly exist data that an organization would not want to share with some communities that make up the audience. It is necessary to determine the minimal information required to convey the message.
- *Timing.* The point in time when an incident is reported can have dramatic impacts within and beyond an organization. This is especially true when the incident investigation reveals a vulnerability that affects many people, departments, or companies.

Philosophy

Reporting an incident will undoubtedly have ramifications internally; and based on the type, scope, and impact of the event, there could be residual effects globally. So, given the exposure and responsibility — why bother? What are the benefits of reporting that you have a weakness or that you were successfully attacked because you were simply negligent in providing even the basic security? In this light, it seems ridiculous to breathe a word that you were a victim. To add to the malaise, if you report an incident prior to assuring the vulnerability used for the attack is not rectified, you may be in for many more opportunities to refine your incidence response process. Finally, once attackers know you do not have a strong security program or do not perform sound security practices, they may attempt to attack you in hope of finding another vulnerability or simply slip

under the radar of confusion that runs rampant in most companies after an incident.

The answer, as one may expect, is not simple.

There are several factors that are used to determine if an incident should be reported, and ultimately, to whom, when, and what should be shared. The following are some of the factors that may need to be considered. Ultimately, it is a lesson in marketing.

Impact Crater. Essentially, how bad was the impact and who — or what — was affected by the debris? With certain events that stretch the imagination and had catastrophic results, it is usually best to be a reporter and provide your perspective, position, and mitigation prior to CNN dropping the bomb on you publicly.

It is usually best to report your situation first rather than be put in the position of defending your actions. This is a reality for public reports in addition to internal reporting. For example, if the IS department makes an enormous security oversight and money is lost due to the exploited vulnerability, accepting responsibility prior to having an investigation uncover the real issue may be best.

Who's on First? Somewhat related to the impact crater, many organizations will be attacked and attempt to deal with it internally — or within the group. Unfortunately for these organizations, the attackers are usually trying to prove their capability in the hacking community. After some chest thumping on news groups, your demise will soon be public. Again, when faced with public interpretation of the event, it is typically better to be first.

Customer Facing. If the attack affected customer systems or data, you may have no choice. You may not have to reveal the incident publicly; but in the event a customer or partner was affected, you must report the situation, history, plan for mitigation, recovery options, and future protection. If you do not, you run an extreme liability risk and might never recover from the loss of reputation.

The previous factors can be presented in many ways, but all cast a dark shadow on the concept of exposure and do not present any positive reason for reporting an incident. No one wants to be perceived as weak publicly or internally — to customers or partners. However, there are factors that, when properly characterized within the scope of the incident and business objectives, it is essential that a report evolve from an event. Following are some points of interest regarding reporting.

Well Done. There are many occasions where a vulnerability was exploited but there was little or no loss associated with the attack. Moreover, the vulnerability may have proved to be extreme in terms of industry

exposure; it just so happened that you experienced the attack on a system you practically forgot was still in the wire closet. Or better yet, your security awareness and vigilance allowed you to identify the incident in real-time, mitigate the attack, and determine the structure and target. This, of course, is how it is supposed to work. Detect, identify, eradicate, and learn — all without suffering from the attack. If this is the case, you could substantially benefit from letting people know how good you are at security.

Fix First. In some situations, it may be beneficial to report an incident to convey to your constituents that there is a new threat afoot and demonstrate your agility and accuracy in handling the incident.

Good Samaritan. In some cases, you may simply be ethically drawn to report the details of an incident for the betterment of the security community and vendors who can learn and improve based on the information. Of course, all previous points may apply — mitigate the exposure and clearly identify the incident.

Truly, at the end of the day, if an event is detected — regardless of impact — there should be a report created and forwarded to a mediator to work within the organization's policy and the dynamics of the attack to properly determine the next step. If the vulnerability is like the recent SNMP vulnerability, it is generally accepted that working with the vendors first is the best plan of global mitigation. How you identify and react to an attack will relate to whom, what, and when you report.

Audience

For better or for worse, the decision has been made to report the incident; and now the appropriate audience must be determined. You can report to one group or several, but assume the obvious leakage when dealing with people and sensitive information. For example, if you do not feel the employees need to know, it would be unwise to tell the partners, customers, or the public. Keeping this in mind, it is also necessary to understand the audience (for the purposes of this discussion); this is your primary audience and others may be indirect recipients — purposely. For example, the managers should know that there was an incident that could impact operations temporarily. This should not be kept from the employees, but the managers could be advised to convey the announcement to their respective groups within a certain time frame.

To add to the complexity, the audience type is proportional to the impact of the incident and the philosophy, or mindset, of performing the report. Essentially, a three-dimensional matrix should be constructed, with one axis being the impact or the criticality of the event, another the response structure (speed, ethics-based or self-preservation, etc.), and the

last a timeline of events. The matrix would then help determine who should know the details of the incident and when.

Nevertheless, it is feasible to segment the different audience types with associated descriptions to help you assess the appropriate target based on the incident characteristics.

Customers. Customers are people, groups, or companies to whom you provide a service or product. Depending on the incident type and scope, it may be necessary to notify them of the event. Customers are entities that invest in your organization through their utilization of your product or service. The greater the investment, the greater the expectation for a supportive and long relationship. If a customer's investment in your organization is affected, reporting may be critical.

As stated above in the section "Well Done," properly responding to an attack and formulating a mitigation process to recover from the attack can offset the strain on the relationship between you and the customer and, in some circumstances, enhance the relationship.

Vendors. One of the more interesting aspects of reporting incidents is the involvement of vendors. For example, if you only use Cisco routers and switches and suffer a breach that is directly associated to a vulnerability in their product, you want them to know about your discovery in order to fix it. In the event they already know, you can become more involved in the remedy process. Of course, you must first overcome the "if they knew, why did I have to get attacked" argument.

Another characteristic of vendor notification is the discovery of the vulnerability through a noncatastrophic incident and having to decide how long they have to fix the vulnerability prior to notifying the public. In many cases, this situation evolves from the discovery of a vulnerability through testing and not the exploitation via an active attack. In the event the vulnerability was determined through a recorded incident, the target organization usually is very patient in allowing the vendor to provide a fix. The patience is mostly due to the desire to let the vendor announce the vulnerability and the fix — making the vendor look good — relieving the victim of the responsibility and exposure and alleviating the vulnerability. If the vulnerability is detected through testing, the testers were usually looking for a weakness to discover. Therefore, in many scenarios, the testers want people to know their discovery; and waiting around for a vendor to provide patches runs against that desire.

In all fairness, it is very common for a vulnerability to be discovered and shared with the vendor prior to letting the general public know. There have been occasions when it has taken the vendor a year to get the fix addressed due to its complexity. The person who discovered the vulnerability was

assured they were working on the fix and was ultimately hired to assist in the mitigation. For vendors that want to have a chance to fix something before the vulnerability is exposed and there are no protection options for their customers, it is necessary to communicate on all levels.

Do not ignore the people who provided you the information. For someone who has expended effort in discovering a vulnerability, the feeling that they are not being taken seriously will definitely expedite the public's awareness of your weakness. One example was a large organization that had a firewall product and received an e-mail detailing a vulnerability and a request for an audience to discuss rectifying the proposed serious hole. After many attempts to gain the much-desired attention, the person became frustrated and turned to the public to ensure that someone would know the existence of the vulnerability. The consequence of ignoring the first contact resulted in customers — some of whom had validated the vulnerability — flooding the vendor with demands for assistance, only to realize the vendor had accomplished very little to date. This entire fiasco reflected badly on the vendor by publicizing its incompetence and inability to meet customer demands with its product.

Partners. Partners are usually companies that establish an alliance with your company to reach a similar objective or augment each other's offerings to customers. Partners can be affected by incidents, especially when there are connections between the entities or the sharing of applications that were impacted. If an incident hinders business operations to a point where a partner's success or safe operation is in jeopardy, a notification with details must be communicated.

It is a crucial priority to advise partners of increased exposure to threats because of an incident on your network. Reporting to the partner the incident and the impact it may directly have on them needs to be addressed in the incident response policies.

Employees. Employees (or contractors) are people who perform the necessary functions required by the company to accomplish the defined business objectives. In nearly every situation, where there is an incident that affects multiple users, employees are typically informed immediately with instructions. The reality is that word-of-mouth and rumor will beat you to it, but providing a comprehensive explanation of the incident and procedures they must follow to protect the company's information assets is necessary.

Managers. Managers are typically informed when the incident can lead to more serious business ramifications that may not be technically related. For example, if an attack is detected that results in the exposure of the entire payroll, employees may get very upset — understandably. It is necessary to control the exposure of information of this nature to the general

population to limit unfounded rumors. Additionally, it must be assumed that there is a strong probability the attacker is an employee. Communication of the incident to the general staff could alert the perpetrators and provide time to eliminate any evidence of their involvement. Obviously, it is necessary for the person or department responsible for the investigation to report to managers to allow them the opportunity to make informed decisions. This is especially critical when the data collected in preliminary investigations may provide evidence of internal misconduct.

Public. One of the more interesting aspects of reporting incidents is communicating to the public the exposures to new threats. In most circumstances, reporting security incidents to the public is not required. For example, a privately held company may experience an event that does not directly impact production, the quality of their product, or the customer's access to that product. Therefore, there is little reason to express the issue, generally speaking. However, it depends on the scope of your company. Following are some examples.

- *Product vendor.* Beyond debate, if a product vendor discovers a vulnerability with its implementation, the vendor is inescapably responsible to communicate this to its clientele. Granted, it is best to develop a solution — quickly — to provide something more than a warning when contacting customers. Sometimes, the general public represents the audience. A clear example is Microsoft and its reaction to security vulnerabilities that will virtually impact everyone.
- *Service providers.* Information service providers, such as application service providers (ASPs), Internet service providers (ISPs), etc., are responsible to their customers to make them aware of an exposure that may affect them. Some very large service providers must disseminate information to a global audience. In addition to the possible scope of a provider's clientele, other service providers can greatly benefit from knowing the impact and process associated with the incident in their attempt to avoid a similar incident. A perfect example is the distributed denial-of-service (DDoS) attack. Now that service providers as well as the developmental community understand the DDoS type of attack, it is easier to mitigate the risk, ultimately gaining more credibility for the industry from the customer's perception.
- *Public companies.* After the ENRON and Arthur Andersen debacle, the sensitivity of disclosing information has reached a new peak. In a short time the trend moved from concern over information accuracy to include information breadth. Consequently, if an incident occurs in an organization that is publicly traded, the repercussions of not clearly reporting incidents could cause problems on many levels.

Content and Timing

What you report and when are driven by the type of incident, scope, and the type of information collected. For internal incidents, ones that affect your organization only, it is typical to provide a preliminary report to management outlining the event and the current tasks being performed to mitigate or recover. The timing is usually as soon as possible to alert all those who are directly associated with the well-being of business operations.

As you can see, the content and timing are difficult to detail due to the close relation to other attributes of the incident. Nevertheless, a rule of thumb is to notify management with as much information as practical to allow them to work with the incident team in formulating future communications. As time passes and the audience is more displaced from the effects of the incident, the information is typically more general and is disseminated once recovery is well on its way.

COMMUNICATION

In communications there should always be a single point within an organization that handles information management between entities. A marketing department is an example of a group that is responsible for interpreting information detailed from internal sources to formulate a message that best represents the information conveyed to the audience. With incident management, a triage team must be identified that serves as the single gateway of information coming into the team and controls what is shared and with whom based on the defined policies. The combination of a limited team, armed with a framework to guide them, ensures that information can be collected into a single point to create a message to the selected audience at the appropriate time.

Reporting an incident, and determining the audience and the details to communicate, must be described in a disclosure policy. The disclosure policy should detail the recipients of a report and the classification of the incident. It should also note whether the report would span audiences and whether the primary audience should be another incident response group internally or a national group such as CERT/CC.

The CERT/CC is a major reporting center for Internet security problems. The CERT/CC can provide technical assistance and coordinate responses to security compromises, identify trends in intruder activity, work with other security experts to identify solutions to security problems, and disseminate information to the broad community. The CERT/CC also analyzes product vulnerabilities, publishes technical documents, and presents training courses. Formerly known as the Computer Emergency Response Team of Carnegie Mellon University, it was formed

at the Software Engineering Institute (SEI) by the Defense Advanced Research Projects Agency (DARPA) in 1988.

Incident response groups will often need to interact and communicate with other response groups. For example, a group within a large company may need to report incidents to a national group; and a national incident response team may need to report incidents to international teams in other countries to deal with all sites involved in a large-scale attack.

Additionally, a response team will need to work directly with a vendor to communicate improvements or modifications, to analyze the technical problem, or to test provided solutions. Vendors play a special role in handling an incident if their products' vulnerabilities are involved in the incident.

Communication of information of this nature requires some fundamental security practices. The information and the associated data must be classified and characterized to properly convey the appropriate message.

Classification

Data classification is an important component of any well-established security program. Data classification details the types of information — in its various states — and defines the operational requirements for handling that information.

A data classification policy would state the levels of classification and provide the requirements associated with the state of the data. For example, a sensitive piece of information may only exist on certain identified systems that meet rigorous certification processes. Additionally, it is necessary to provide the distinctive characteristics that allow people to properly classify the information. The data classification policy must be directly correlated with the incident management policy to ensure that information collected during investigation is assigned the appropriate level of security.

Included in the policy is a declassification process for the information for investigative processes. For example, the data classification policy may state that operating system DLL files are sensitive and cannot have their security levels modified. If the DLL becomes a tool or target of an attack, it may be necessary to collect the data that may need to be reported. It is at this point the incident response management policy usually takes precedence. Otherwise, bureaucracy can turn the information collection of the incidence response team into an abyss, leading to communication and collaboration issues that could hinder the response process.

Identification and Authentication

Prior to sharing information, it should be considered a requirement to authenticate the recipient(s) of the information. Any response organization, including your own, should have some form of identification that can be authenticated.

Certificates are an exceptional tool that can be utilized to identify a remote organization, group, individual, or role. Authentication can be provided by leveraging the supporting public key infrastructure (PKI) to authenticate via a trusted third party through digital signatures. Very similar to PKI — and also based on asymmetrical encryption — pretty good privacy (PGP) can authenticate based on the ability to decrypt information or sign data proving the remote entity is in possession of the private key.

Confidentiality

Once you have asymmetrical keys and algorithms established for authentication, it is a short step to use that technology to provide confidentiality. Encryption of sensitive data is considered mandatory, and the type of encryption will more than likely use large keys and advanced algorithms for increased security.

Symmetrical as well as asymmetrical encryption can be used to protect information in transit. However, given the sensitivity, multiple forms of communication, and characteristics of information exchange, asymmetrical encryption is typically the algorithm of choice. (The selection default to asymmetrical also simplifies the communication process, because you can use the same keys for encryption that were used for the authentication.

CONCLUSION

Incident reporting is a small but critical part of a much more comprehensive incident management program. As with anything related to information security, the program cannot survive without detailed policies and procedures to provide guidance before, during, and after an incident occurs. Second only to the policy is the technology. Properly configured network elements that deliver the required information to understand the event and scope are essential.

Collecting the information from various sources and managing that information based on the policies are the preliminary steps to properly reporting the incident. Reporting is the final frontier. Clearly understanding the content and the audience that requires the different levels of information are essentially the core concerns for the individuals responsible for sharing vital and typically sensitive information.

Reporting incidents is not something that many organizations wish to perform outside the company, but this information is critical to the

advancement and awareness of the security industry as a whole. Understanding what attacks people are experiencing will help many others, through increased consciousness of product vendors, developers, and the security community as a whole, to further reduce the seriousness of security incidents to the entire community.

ABOUT THE AUTHOR

James S. Tiller, CISSP, MSCE+I, is the Global Portfolio and Practice Manager for International Network Services in Tampa, Florida.

Incident Response Management

Alan B. Sterneckert, CISA, CISSP, CFE, CCCI

Incident response management is the most critical part of the enterprise risk management program. Frequently, organizations form asset protection strategies focused primarily on perceived rather than actual weaknesses, while failing to compare incident impact with continuing profitable operations. In the successful implementation of risk management programs, all possible contingencies must be considered, along with their impact on the enterprise and their chances of occurring.

By way of illustration, in the 1920s and 1930s, France spent millions of francs on the construction of the Maginot Line defenses, anticipating an invasion similar to the World War I German invasion. At that time, these fortifications were considered impregnable. During the 1940 German army invasion, they merely bypassed the Maginot Line, rendering these expensive fortifications ineffective. The Maginot planners failed to consider that invaders would take a route different than previous invasions, resulting in their defeat.

RISK MANAGEMENT PROJECT

Risk management is not a three-month project; it is not a project that, when completed, becomes shelved and never reviewed again. Rather, it is a continuous process requiring frequent review, testing, and revision. In the most basic terms, risk has two components: the probability of a harmful incident happening and the impact the incident will have on the enterprise.

TOP-DOWN RISK MANAGEMENT PROJECT PLANNING

Beginning at the end is a description of top-down planning. Information technology (IT) professionals must envision project results at the highest level by asking, what are my deliverables? Information risk management deliverables are simply defined: confidentiality, integrity, and availability (CIA). CIA, and the whole risk management process, must be first considered

in the framework of the organization's strategic business plans. A formula for success is to move the risk management program forward with a clear vision of the business deliverables and their effect on the organization's business plans.

The concept of risk management is relatively simple. Imagine that the organization's e-mail service is not functioning or that critical data has been destroyed, pilfered, or altered. How long would the organization survive? If network restoration is achieved, what was the business loss during the restoration period? It is a situation in which one hopes for the best but expects the worst. Even the best risk management plan deals with numerous *what-if* scenarios. What if a denial-of-service (DoS) attacks our network? Or what if an employee steals our customer list? What if a critical incident happens — who is responsible and authorized to activate the incident response team?

In the world of risk management, the most desirable condition is one in which risks are avoided. And if risks cannot be avoided, can their frequency be increased and can their harmful effects be mitigated?

RISK MANAGEMENT KEY POINTS

These are general key points in developing a comprehensive risk management plan:

- Document the impact of an extended outage on profitable business operations in the form of a business impact analysis. Business impact analysis measures the effects of threats, vulnerabilities, and the frequency of their occurrence, against the organization's assets.
- Remember that risk management only considers risks at a given moment. These risks change as the business environment changes, necessitating the constantly evolving role of risk management.
- Complete a gap analysis, resulting in the measured difference between perceived and actual weaknesses and their effects on key assets.

OVERALL PROJECT PLANNING

Incident response planning is no different than other planning structures. There are four basic key phases:

1. Assess needs for asset protection within the organization's business plan
2. Plan
3. Implement
4. Revise

In assessing needs, representatives of the affected departments should participate in the initial stage and should form the core of the project team.

Additional experts can be added to the project team on an ad hoc basis. This is also a good time to install the steering committee that has overall responsibility for the direction and guidance of the project team. The steering committee acts as a buffer between the project team and the various departmental executives. The early stage is the time for hard and direct questions to be asked by the project team members in detailing the business environment, corporate culture, and the minimum organizational infrastructure required for continuing profitable operations.

It becomes important to decide the project's owners at the outset. Project ownership and accountability are based on two levels: one is the line manager who oversees the project team, and the other is the executive who handles project oversight. This executive–owner is a member of the steering committee and has departmental liaison responsibilities. Project scope, success metrics, work schedules, and other issues should be decided by the project team. Project team managers, acting in cooperation with the steering committee, should keep the project focused, staffed, and progressing.

Planning is best conducted in an atmosphere of change control. The project team's direction will become lost if formal change control procedures are not instituted and followed. Change controls decide what changes may be made to the plan, who may approve changes, why these changes are being made, and the effect of these changes. It is critical that change controls require approvals from more than one authority, and that these changes are made part of any future auditing procedure. Once changes are proposed, approved, and adopted, they must be documented and incorporated as part of the plan.

With planning completed, implementation begins. Implementations do not usually fail because of poor planning; rather, they fail due to lack of accountability and ownership. Initial testing is conducted as part of the implementation phase. During the implementation step, any necessary modifications must be based on test results. Specific testing activities should include defining the test approach, structuring the test, conducting the test, analyzing the test results, and defining success metrics with modifications as required. In an organizational setting, the testing process should be executed in a quarantined environment, where the test is not connected to the work platforms and the data used for the test is not actual data. During testing, criteria should be documented so performance can be measured and a determination made as to where the test succeeded or failed.

With the implementation and testing completed, the project moves toward final adjustments that are often tuned to the changing business environment. Remember to maintain change controls in this phase also. More than one engineer has been surprised to find two identical hosts offering the same services with different configurations.

ENTERPRISE RISK

Risk is the possibility of harm or loss. Risk analysis often describes the two greatest sources of risk as human causes and natural causes. Before a risk can be managed, consideration must be given to the symptom as well as the result. Any risk statement must include what is causing the risk and the expected harmful results of that risk.

KEY ASSETS

Key assets are those enterprise assets required to ensure that profitable operations continue after a critical incident. Define, prioritize, and classify the organization's key assets into four general areas: personnel, data, equipment, and physical facilities. Schedule, in the form of a table, the priority of the organization's key assets and their associated threats and vulnerabilities. This table will serve the purpose of identifying security requirements associated with different priority levels of assets.

In developing asset values, the asset cost is multiplied by the asset exposure factor, with the resulting product being the single loss expectancy. The asset value is the replacement value of a particular asset, while the exposure factor is the measure of asset loss resulting from a specific harmful event. Multiplying this single loss expectancy by the annualized rate of occurrence will result in the annualized loss expectancy. An example of this equation is as follows: assume the replacement value of a server facility, complete with building, equipment, data, and software, is $10 million. This facility is located in a geographic area prone to hurricanes that have struck three times in the past ten years and resulted in total facility losses. Annualized expectancy is the loss of the facility, data, and equipment once every three years, or 33 percent annually.

Step two of our four-step process is a threat assessment. Threats are simply defined as things that can possibly bring harm upon assets. Threats should be ranked by type, the impact they have on the specific asset, and their probability of occurrence. Even the most effective risk management plan cannot eliminate every threat; but with careful deliberation, most threats can be avoided or their effects minimized.

Identify vulnerabilities (weaknesses) in the security of the enterprise's key assets. Vulnerable areas include physical access, network access, application access, data control, policy, accountability, regulatory and legal requirements, operations, audit controls, and training. Risk levels should be expressed as a comparison of assets to threats and vulnerabilities. Create a column in the table (Exhibit 48-1) providing a relative metric for threat frequency. Once completed, this table provides a measurement of the level of exposure for a particular key asset.

Exhibit 48-1. Measurement of the level of exposure.

Asset	Threat	Frequency	Vulnerability	Impact
Name and Replacement Value	Type	Annualized	Type and Ranking: High, Medium, Low	Ranking: High, Medium, Low

Avoidance and mitigation steps are processes by which analyses are put into action. Having identified the organization's key assets, threats to these assets, and potential vulnerabilities, there should be a final analytical step detailing how the specific risk can be avoided. If risk avoidance is not possible, then can the chance of its occurrence be extended?

From the outset it is recommended to include auditors. Audits must be scheduled and auditors' workpapers amended, assuring compliance with laws, regulations, policies, procedures, and operational standards.

RISK MANAGEMENT BEST PRACTICES DEVELOPMENT

As part of risk management best practices, there are three principle objectives: avoiding risk, reducing the probability of risk, and reducing the impact of the risk.

Initiate and foster an organizational culture that names every employee as a risk manager. Employee acceptance of responsibility and accountability pays short- and long-term dividends. In some circumstances, the creation of this risk manager culture is more important than developing and issuing extensive policies and procedures.

In a general sense, there are four key best practice areas that should be addressed: organizational needs, risk acceptance, risk management, and risk avoidance.

Organizational needs determine the requirement for more risk study and more information in ascertaining the characteristics of risk before taking preventive or remedial action.

Risk acceptance is defined in these terms: if these risks occur, can the organization profitably survive without further action?

Risk management is defined as efforts to mitigate the impact of the risk should it occur.

Risk avoidance includes the steps taken to avoid the risk from happening.

RISK CONTROLS

Avoidance controls are proactive in nature and attempt to remove, or at least minimize, the risk of accidental and intentional intrusions. Examples of these controls include encryption, authentication, network security architecture, policies, procedures, standards, and network services interruption prevention.

Assurance controls are actions, such as compliance auditing, employed to ensure the continuous effectiveness of existing controls. Examples of these controls include application security testing, standards testing, and network penetration testing.

Detection controls are tools, procedures, and techniques employed to ensure early detection, interception, containment, and response to unauthorized intrusions. Examples of these controls include intrusion detection systems (IDSs) and remotely managed security systems.

Recovery controls involve response-related steps in rapidly restoring secure services and investigating the circumstances surrounding information security breaches. Included are legal steps taken in the criminal, civil, and administrative arenas to recover damages and punish offenders. Examples of these recovery controls include business continuity planning, crisis management, recovery planning, formation of a critical incident response team, and forensic investigative plans.

CRITICAL INCIDENT RESPONSE TEAM (CIRT)

A CIRT is a group of professionals assembled to address network risks. A CIRT forms the critical core component of the enterprise's information risk management plan. Successful teams include management personnel having the authority to act; technical personnel having the knowledge to prevent and repair network damage; and communications experts having the skills to handle internal and external inquiries. They act as a resource and participate in all risk management phases. CIRT membership should be composed of particular job titles rather than specifically named individuals. The time for forming a CIRT, creating an incident response plan, notification criteria, collecting tools, training, and executive-level support is not the morning after a critical incident. Rather, the CIRT must be ready for deployment before an incident happens. Rapidly activating the CIRT can

mean the difference between an outage costing an organization its livelihood or being a mere annoyance.

Organizational procedures must be in place before an incident so the CIRT can be effective when deployed. This point is essential, because organizations fail to address critical incidents even when solid backup and recovery plans are in place. The problem is usually found to be that no one was responsible to activate the CIRT.

The CIRT plan must have clearly defined goals and objectives integrated in the organization's risk management plan. CIRT's mission objectives are planning and preparation, detection, containment, recovery, and critique. As part of its pre-incident planning, the CIRT will need: information flowcharts, hardware inventory, software inventory, personnel directories, emergency response checklists, hardware and software tools, configuration control documentation, systems documentation, outside resource contacts, organization chart, and CIRT activation and response plans. For example, when arriving on the scene, the CIRT should be able to review its documentation ascertaining information flow and relevant critical personnel of the organization's employee healthcare benefits processing unit.

Considering the nature, culture, and size of the organization, an informed decision must be made about when to activate the CIRT. What is the extent of the critical incident before the CIRT is activated? Who is authorized to make this declaration? Is it necessary for the whole CIRT to respond? Included in the CIRT activation plan should be the selection of team members needed for different types or levels of incidents.

If circumstances are sensitive or if they involve classified materials, then the CIRT activation plan must include out-of-band (OOB) communications. OOB communications take place outside the regular communications channels. Instead, these OOB communication methods include encrypted telephone calls, encrypted e-mail not transmitted through the organization's network, digital signatures, etc. The purpose of OOB communications is to ensure nothing is communicated through routine business channels that would alert someone having normal access to any unusual activity.

INCIDENT RESPONSE STEPS

The goals of incident response must serve a variety of interests, balancing the organization's business concerns with those of individual rights, corporate security, and law enforcement officials. An incident response plan will address the following baseline items:

- Determine if an incident has occurred and the extent of the incident.
- Select which CIRT members should respond.
- Assume control of the incident and involve appropriate personnel, as conditions require.
- Report to management for the decision on how to proceed.
- Begin interviews.
- Contain the incident before it spreads.
- Collect as much accurate and timely information as possible.
- Preserve evidence.
- Protect the rights of clients, employees, and others, as established by law, regulations, and policies.
- Establish controls for the proper collection and handling of evidence.
- Initiate a chain of custody of evidence.
- Minimize business interruptions within the organization.
- Document all actions and results.
- Restore the system.
- Conduct a post-incident critique.
- Revise response as required.

Pre-incident preparation is vital in approaching critical incidents. Contingency plans that are tested and revised will be invaluable in handling incidents where a few minutes can make the difference between disaster and a complete restoration of key services. Network administrators should be trained to detect critical incidents and contact appropriate managers so a decision can be made relative to CIRT deployment. Some of the critical details that administrators should note are the current date and time, nature of the incident, who first noticed the incident, the hardware and software involved, symptoms, and results.

Suspected incidents will usually be detected through several processes, including intrusion detection systems (IDSs), system monitors, and firewalls. Managers should decide whether the administrators should attempt to isolate the affected systems from the rest of the network. Trained, experienced administrators can usually perform these preliminary steps, thereby preventing damage from spreading (see Exhibit 48-2).

At the time of the initial response by the CIRT, no time should be lost looking for laptops, software, or tools. They should arrive at the scene with their plan, tools, and equipment in hand. CIRT members will begin interviews immediately in an effort to determine the nature and extent of any damage. It is important that they document these interviews for later action or as evidence. The CIRT will obtain and preserve the most volatile evidence immediately. After an initial investigation, the CIRT will formulate the best response and obtain management approval to proceed with further investigation and restoration steps.

Exhibit 48-2. Immediate actions to be taken by administrators to contain an incident.

1. Extinguish power to the affected systems. This is a drastic but effective decision in preventing any further loss or damage.
2. Disconnect the affected equipment from the network. There should be redundant systems so users will have access to their critical services.
3. Disable specific services being exploited.
4. Take all appropriate steps to preserve activity and event logs.
5. Document all symptoms and actions by administrators.
6. Notify system managers. If authorized, notify the CIRT for response.

CRITICAL INCIDENT INVESTIGATION

The goals of law enforcement officers and private investigators are basically the same. Both types of investigators want to collect evidence and preserve it for analysis and presentation at a later date. Evidence is simply defined as something physical and testimonial, material to an act. It is incumbent upon the CIRT to establish liaison with the appropriate levels of law enforcement to determine the best means of evidence collection, preservation, and delivery. If there are circumstances where law enforcement officers are not going to be involved, then the CIRT members should consider the wisdom of either developing forensic analysis skills or contracting others to perform these functions. Evidence collection and analysis are critical because incorrect crime scene processing and analysis can render evidence useless. Skilled technicians with specialized knowledge, tools, and equipment should accomplish collecting, processing, and analyzing evidence. Frequently, investigators want to be present during evidence collection and interviews; consequently, CIRT members should establish liaison with law enforcement and private investigators to establish protocols well in advance of a critical incident.

Evidence may be voluntarily surrendered, obtained through the execution of a search warrant, through a court order or summons, or through subpoenas. It is a common practice for investigators to provide a receipt for evidence that has been delivered to them. This receipt documents the transfer of items from one party to another and supports the chain of custody. It is important to note that only law enforcement investigators use search warrants and subpoenas to obtain evidence. Once received, the investigator will usually physically mark the evidence for later identification. Marking evidence typically consists of the receiving investigators placing the date and their initials on the item. In the case of electronic media, the item will be subjected to special software applications, causing a unique one-way identifier to be created and written to the media, thereby identifying any subsequent changes in the media's contents.

Does the investigator have the right to seize the computer and examine its contents? In corporate environments this right may be granted by policy. The enterprise should have a policy stating the ownership of equipment, data, and systems. It is a usual practice that organizations have policies requiring employees to waive any right to privacy as a condition of their employment. If the organization has such policies, it is important that its legal and human resources officers are consulted before any seizure takes place.

Under current United States law and the Fourth Amendment to the U.S. Constitution, the government must provide a judge or magistrate with an affidavit detailing the facts and circumstances surrounding the alleged crime. Search warrants are two-part documents. The first part is the search warrant, which bears a statutory description of the alleged crime, a description of the place to be searched, and the items or persons to be seized. At the conclusion of the search warrant execution, a copy of this search warrant document must be deposited at the premises, regardless of whether it was occupied. Affidavits are the second part of the search warrant and are statements where the officer or agent, known as the *affiant*, swears to the truth of the matter. The law does not require the affiant to have first-hand knowledge of the statement's details, merely that the affiant has reliable knowledge. Search warrants are granted based upon the establishment of probable cause. It is important to note that the affidavit must stand on its own; all relevant information must be contained within its borders.

Questions surrounding search warrants are these: is it probable that a crime has been committed, and is it probable that fruits, instrumentalities, or persons connected to that crime are located at a given location now? Unless there are unusual circumstances, search warrants may only be executed in daylight hours from 6 a.m. to 10 p.m. If unusual circumstances exist, then these must be submitted to the court. Such circumstances include the possibility of extreme danger to the officers or the likelihood of evidence destruction. Search warrants must be announced, and authorities must declare their purpose. At the completion of the search warrant, the officers are required to deposit a copy of the search warrant and an inventory of the items seized at the searched premises. Under special circumstances, the search warrant will be *sealed* by the issuing court. This means the sworn statement is not public record until unsealed by the issuing court. If the affidavit is not sealed, then it is a public document and retrievable from the court's office. At the conclusion of the search warrant, a return is completed and accompanied by an inventory of the seized items. This search warrant return is part of the original search warrant document and reflects the date, by whom, and where it was executed. Along with the search warrant return, an inventory of seized items is filed with the court, where it is available for public review. Law enforcement and non-law enforcement personnel, depending upon the nature of the investigation, may obtain court orders and summons. These documents are

based upon applications made to the court of jurisdiction and may result in orders demanding evidence production by the judge or magistrate. Similar to search warrants, court orders are usually two-part documents with an application stating the reason the judge should issue an order to a party to produce items or testimony. The second part is the actual court order document. Court orders state the name of the case, the items to be brought before the court, the date the items are to be brought before the court, the location of the court, the name of the presiding judge, and the seal of the court. Summonses are similar to court orders and vary from jurisdiction to jurisdiction. Subpoenas are generally categorized as one of two types: one resulting from a grand jury investigation, and the second resulting from a trial or other judicial proceeding. Both documents carry the weight of the court — meaning these documents are demands that, if ignored, can result in contempt charges filed against persons or other entities. Grand juries are tasked with hearing testimony and reviewing evidence, hence their subpoenas are based upon investigative need. Their members are selected from the local community, and they are impaneled for periods of several months. Items or persons may be subpoenaed before a grand jury for examination. It is possible for a motion to quash the subpoena to be filed, causing the court to schedule a hearing where the subpoena's merits are heard. Different than grand jury subpoenas, judicial subpoenas are issued for witnesses and evidence to be presented at trial or other hearings. Testimony is obtained through interviews, depositions, and judicial examinations. Interviewing someone is a conversation directed toward specific events. Interviews may be recorded in audio or video form, or the investigator may take carefully written notes. In the latter case, the interviewer's notes are reduced to a report of the interview. This report serves as the best recollection of the investigator and is not generally considered a verbatim transcript of the interview.

Depositions are more formal examinations and are attended by attorneys, witnesses, and persons who create a formal record of the proceedings. Usually, depositions are part of civil and administrative proceedings; however, in unusual circumstances they may be part of a criminal proceeding. Attorneys ask questions of the witnesses, with the plaintiff and defense attempting to ask questions that will cause the witness to provide an explanation favorable to their side. Judicial examinations are made before a judge or magistrate judge, and the witnesses are sworn to tell the whole truth while the proceedings are recorded.

It is important to note that providing mischaracterizations, lies, or withholding information during interviews may be considered grounds for criminal prosecution. In a similar vein, the CIRT and others must be very careful interviewing potential subjects and collecting evidence. If interviews are conducted or evidence is collected through coercion, these actions could be considered as intimidating and may be considered for charges.

FORENSIC EXAMINATION

There are several schools of thought in completing the forensic examination of evidence. Regardless, one rule remains steadfast — no examination should be conducted on original media; and the media, constituting evidence, must remain unchanged. There are several ways to obtain copies of media. There are forensic examination suites designed to perform exact bit-by-bit duplication; and there are specific software utilities used in duplicating media and hardware-copying devices that are convenient, but these are generally limited to the size and characteristics of the disks they can clone. There are also utilities that are part of some operating platforms that can produce bit-by-bit media duplications. It is important to remember that all forensic examination processes must be documented in the form of an activity log and, in the case of some very sensitive matters, witnessed by more than one examiner.

Forensic examiners must ensure that their media is not contaminated with unwanted data; so many have a policy that, before any evidence is copied, media will be cleansed with software utilities or a degaussing device designed for such purposes. In this fashion, the examiner can testify that appropriate precautions were taken to prevent cross-contamination from other sources.

As in the case of all evidence-handling practices, a chain of custody is prepared. Chain of custody is merely a schedule of the evidence, names, titles, reason for possession, places, times, and dates. From the time of the evidence seizure, the chain of custody is recorded and a copy attached to the evidence. The chain of custody documentation is maintained regardless of how the evidence was seized or whether the evidence is going to be introduced in criminal, civil, or administrative proceedings.

A covert search is one targeting a specific console or system involving real-time monitoring, and it is usually conducted discreetly. In a practical example, an organization may suspect one of its employees of downloading inappropriate materials in violation of its use policy. After examining logs, an exact workstation cannot be identified. There are two ways to conduct a covert search after authorization is obtained. One method copies the suspected hard drive and replaces it with the copy, with the original considered as evidence. The second method duplicates the suspect's hard drive while it remains in the computer. The duplicate is considered evidence and is duplicated again for examination. In either method it is important to ascertain that the organization has the right to access the equipment and that the suspect does not have any reasonable expectation of privacy. This topic must be fully addressed by the legal and human resources departments.

After having seized the evidence, the examiner decides to either conduct an analysis on the premises or take the media to another location. The advantage of having the examination take place where the evidence is seized is obvious. If there is something discovered requiring action, it can be addressed immediately. However, if the examination takes place in the calm of a laboratory, with all the tools available, then the quality of the examination is at its highest.

The CIRT and other investigators must consider the situation of sensitive or classified information that is resident on media destined for a courtroom. Sometimes, this consideration dissuades some entities from reporting criminal acts to the authorities. However, there are steps that can be legally pursued to mitigate the exposure of proprietary or sensitive information to the public.

CRIMINAL, FORFEITURE, AND CIVIL PROCESSES

Criminal acts are considered contrary to publicly acceptable behavior and are punished by confinement, financial fines, supervised probation, and restitution. Felonies are considered major crimes and are usually punished by periods of confinement for more than one year and fines of more than $1000. In some jurisdictions, those convicted of felonies suffer permanent loss of personal rights. Misdemeanors are minor crimes punishable by fines of less than $1000 and confinement of less than one year.

Sentencing may include confinement, fines, or a period of probation. The length of sentence, fines, and victim restitution depends upon the value of the crime. If proprietary information is stolen and valued at millions of dollars, then the sentence will be longer with greater fines than for an act of Web page defacement. There are other factors that can lengthen sentencing. Was the defendant directing the criminal actions of others? Was the defendant committing a crime when he committed this crime? Has the defendant been previously convicted of other crimes? Was the defendant influencing or intimidating potential witnesses? There are also factors that can reduce a sentence. Has the defendant expressed remorse? Has the defendant made financial restitution to the victim? Has the defendant cooperated against other possible defendants? Under the laws of the United States, the length of sentence, the type of sentence, and fines are determined in a series of weighted numerical calculations and are codified in the Federal Sentencing Guidelines. At the time of sentencing, a report is usually prepared and delivered to the sentencing judge detailing the nature of the crime and the extent of the damage. It is at the judge's discretion whether to order financial restitution to the victim; however, in recent times, more and more judges are inclined to order financial restitution as part of sentencing (see Exhibit 48-3).

Exhibit 48-3. Partial list of applicable federal criminal statutes.

- 18 United States Code Section 1030 Fraud Activities with Computers
- 18 United States Code Section 2511 Unlawful Interception of Communications
- 18 United States Code Section 2701 Unlawful Access to Stored Electronic Communications
- 18 United States Code Section 2319 Criminal Copyright Infringement
- 18 United States Code Section 2320 Trafficking in Counterfeit Goods or Services
- 18 United States Code Section 1831 Economic Espionage
- 18 United States Code Section 1832 Theft of Trade Secrets
- 18 United States Code Section 1834 Criminal Asset Forfeiture
- 18 United States Code Section 1341 Mail Fraud
- 18 United States Code Section 1343 Wire Fraud
- 18 United States Code Sections 2251–2253 Sexual Exploitation of Children Act
- 18 United States Code Section 371 Criminal Conspiracy

Frequently, organizations ask if there are statutory requirements for reports of criminal activities. Under the criminal codes of the United States, Title 18, Section 4, it states: "Whoever having knowledge of the actual commission of a felony cognizable by a court of the United States, conceals and does not as soon as possible make known the same to some judge or other person in civil or military authority under the United States, shall be fined under this title or imprisoned not more than three years or both." Many jurisdictions have similar statutes requiring the reporting of criminal activities.

Civil matters are disputes between parties that are resolved by the exchange of money or property. Civil suits may have actual, punitive, and statutory damages. In the case of actual damages, the plaintiff must prove to a preponderance of evidence (51 percent) that they suffered specific losses. Punitive damages are amounts that punish the defendant for harming the plaintiff. Statutory damages are those prescribed by law.

Many jurisdictions have laws allowing the simultaneous criminal prosecution of a defendant, a civil suit naming the same defendant, and allowing forfeiture proceedings to take place. This type of multifaceted prosecution is known as parallel-track prosecution.

Pursuant to criminal activities, many jurisdictions and the U.S. federal government, file concurrent forfeiture actions against offending entities. These proceedings also impact the relationship between CIRT members and the court system. Depending upon the specific jurisdiction, these actions may take the form of the criminal's assets being indicted, or civil suits filed against those assets, or those assets being administratively forfeited. An example of this type of parallel-track prosecution is illustrated with the person who unlawfully enters an organization's network and steals sensitive protected information that is subsequently sold to a competitor.

Investigators conduct a thorough investigation, and the perpetrator is indicted. In this same case, a seizure warrant is obtained; and the defendant's computer equipment, software, and the crime's proceeds are seized. Depending upon the laws, the perpetrator may suffer confinement, loss of money resulting from the information sale, the forfeiture of his equipment or other items of value, restitution to the victim, and fines. It is also a reasonable and acceptable process that the subject is civilly sued for damages while he is criminally prosecuted and his assets forfeited.

USE OF MONITORING DEVICES

The enterprise must have policies governing the use of its system resources and the conduct of its employees. Pursuant to those policies, the CIRT may monitor network use by suspected employee offenders. The use of monitoring techniques is governed by the employees' reasonable expectation of privacy and is defined by both policy and law. Techniques used to monitor employee activities should be made part of audit and executive-level review processes to make certain these monitoring practices are not abused. Before implementing computer monitoring, it is wise to consult the organization's human resources and legal departments because, if these policies are not implemented correctly, computer monitoring can run afoul of legal, policy, and ethical standards. Under federal statutes, network administrators are granted the ability to manage their systems. They may access and control all areas of their network and interact with other administrators in the performance of their duties. Because unauthorized system intruders do not have an expectation of privacy, their activities are not subject to such considerations. If administrators discover irregularities, fraud, or unauthorized software such as hacking tools on their systems, they are allowed to take corrective actions and report the offenders.

However, this is not the case for government agencies wanting access to network systems and electronic communications. Depending upon the state of the electronic communications, they may be required to obtain a court order, search warrant, or subpoena.

It is important to note that most jurisdictions do not allow retributive actions. For example, if a denial-of-service attack causes the organization to suffer losses, it may be considered unlawful for the organization to return a virus to the offender.

NATURE OF CRIMINAL INCIDENTS

Viruses and worms have been in existence for many years. Since the introduction of the Morris Worm in 1988, managers and administrators have paid attention to their potential for harm. In years past, viruses and worms were ignored by law enforcement, and treated as merely a nuisance. However, in more recent times, following the outbreaks of Melissa and the

Love Bug, persons responsible for their creation and proliferation are being investigated and prosecuted.

Insider attacks usually consist of employees or former employees gaining access to sensitive information. Because they are already located inside the network, it is possible they have already bypassed many access barriers; and, by elevating their privileges, they may gain access to the organization's most valuable information assets. Among the insiders are those who utilize the organization's information assets for their own purposes. Downloading files in violation of use policies wastes valuable resources and, depending upon their content, may be a violation of law.

Outsider attacks are more than an annoyance. A determined outsider may hammer at the target's systems until an entry is discovered. Attackers may be malicious or curious. Regardless, their efforts have the same results in that unauthorized entry is made. Often, their attacks cause serious damage to information systems and compromise sensitive data. Attackers do not need thorough systems knowledge because there are many Web sites that provide the necessary tools for intrusions and DoS attacks.

Unauthorized interception of communications may take place when an unauthorized intrusion takes place and software is installed allowing the intruder to monitor keystrokes and communications traffic. Because this activity is performed without the permission of the system owners, it may have the same net effect as an illegal wiretap.

DoS attacks gained significant negative publicity recently as unscrupulous persons targeted high-profile Web sites, forcing them offline. In some cases, perpetrators were unwitting participants, wherein their broadband assets were compromised by persons installing software executing distributed DoS attacks. These attacks flood their target systems with useless data launched from single or multiple sources, causing the target's network to crash.

CONCLUSION

Risk management consists of careful planning, implementation, testing, and revision. The most critical part of risk management is critical incident response. The principal purpose of risk management is avoidance and mitigation of harm. Incident response, with the development of a solid response strategy, outside liaison, and a well-trained CIRT, can make the difference between a manageable incident and a disaster costing the organization its future.

ABOUT THE AUTHOR

Alan B. Sterneckert, CISA, CISSP, CFE, CCCI, is the owner and general manager of Risk Management Associates located in Salt Lake City, Utah. A retired Special Agent, Federal Bureau of Investigation, Mr. Sterneckert is a

professional specializing in risk management, IT system security, and systems auditing. In 2003, Mr. Sterneckert will complete a book about critical incident management, to be published by Auerbach.

157

Ethics and the Internet

Micki Krause, CISSP

The research for this chapter was done entirely on the Internet. The Net is a powerful tool. This author dearly hopes that the value of its offerings is not obviated by those who would treat the medium in an inappropriate and unethical manner.

Ethics: Social values; a code of right and wrong

Introduction

The ethical nature of the Internet has been likened to "a restroom in a downtown bus station," where the lowest of the low congregate and nothing good ever happens. This manifestation of antisocial behavior can be attributed to one or more of the following:

- The relative anonymity of those who use the Net
- The lack of regulation in cyberspace
- The fact that one can masquerade as another on the Internet
- The fact that one can fulfill a fantasy or assume a different persona on the net, thereby eliminating the social obligation to be accountable for one's own actions

Whatever the reason, the Internet, also known as the "Wild West" or the "untamed frontier," is absent of law and therefore is a natural playground for illicit, illegal, and unethical behavior.

In the ensuing pages, we will explore the types of behavior demonstrated in cyberspace, discuss how regulation is being introduced and by whom, and illustrate the practices that businesses have adopted in order to minimize their liability and encourage their employees to use the Net in an appropriate manner.

The Growth of the Internet

When the Internet was born approximately 30 years ago it was a medium used by the government and assorted academicians, primarily to perform and share research. The user community was small and mostly self-regulated. Thus, although a useful tool, the Internet was not considered "mission-critical," as it is today. Moreover, the requirements for availability and reliability were not as much a consideration then as they are now, because Internet usage has grown exponentially since the late 1980s.

The increasing opportunities for productivity, efficiency and world-wide communications brought additional users in droves. Thus, it was headline news when a computer worm, introduced into the Internet by Robert Morris, Jr., in 1988, infected thousands of Net-connected computers and brought the Internet to its knees.

In the early 1990s, with the advent of commercial applications and the World Wide Web (WWW), a graphical user interface for Internet information, the number of Internet users soared. Sources such as the *Industry Standard*, "The Newsmagazine of the Internet Economy," published the latest Nielsen Media Research Com-

EXHIBIT 157.1 GenX Internet Use

A Higher Percentage of Gen-Xers Use the Web…	
	Used the Web in the past 6 months
Generation X	61%
Total U.S. Adults	49%
… More Regularly…	
	Use the Web regularly
Generation X	82%
Baby Boomers	52%
… Because it's the Most Important Medium	
	Most Important Media
Internet	55%
Television	39%

Source: *The Industry Standard*, M.J. Thompson, July 10, 1998.

merce Net study in late 1998, which reported the United States Internet population at 70.5 million (out of a total population of 196.5 million).

Today, the Internet is a utility, analogous to the electric company, and "dotcom" is a household expression. The spectrum of Internet users extends from the kindergarten classroom to senior citizenry, although the Gen-X generation, users in their 20s, are the fastest adopters of Net technology (see Exhibit 157.1).

Because of its popularity, the reliability and availability of the Internet are critical operational considerations, and activities that threaten these attributes, e.g., spamming, spoofing, hacking and the like, have grave impacts on its user community.

Unethical Activity Defined

Spamming, in electronic terminology, means electronic garbage. Sending unsolicited junk electronic mail, for example, such as an advertisement, to one user or many users via a distribution list, is considered spamming.

One of the most publicized spamming incidents occurred in 1994, when two attorneys (Laurence Carter and Martha Siegel) from Arizona, flooded the cyber waves, especially the Usenet newsgroups,* with solicitations to the immigrant communities of the United States to assist them in the green card lottery process to gain citizenship. Carter and Siegel saw the spamming as "an ideal, low-cost and perfectly legitimate way to target people likely to be potential clients" (*Washington Post*, 1994). Many Usenet newsgroup users, however, saw things differently. The lawyers' actions resulted in quite an uproar among the Internet communities primarily because the Internet has had a long tradition of noncommercialism since its founding. The attorneys had already been ousted from the American Immigration Lawyers' Association for past sins, and eventually they lost their licenses to practice law.

There have been several other spams since the green card lottery, some claiming "MAKE MONEY FAST," others claiming "THE END OF THE WORLD IS NEAR." There have also been hundreds, if not thousands, of electronic chain letters making the Internet rounds. The power of the Internet is the ease with which users can forward data, including chain letters. More information about spamming occurrences can be found on the Net in the Usenet newsgroup (alt.folklore.urban).

Unsolicited Internet e-mail has become so widespread that lawmakers have begun to propose sending it a misdemeanor. Texas is one of 18 states considering legislation that would make spamming illegal. In February 1999, Virginia became the fourth state to pass an antispamming law. The Virginia law makes it a misdemeanor for a spammer to use a false online identity to send mass mailings, as many do. The maximum penalty would be a $500 fine. However, if the spam is deemed malicious and results in damages to the victim in excess of $2500 (e.g., if the spam causes unavailability of computer service), the crime would be a felony, punishable by up to five years in prison. As with the Virginia law, California law allows for the jailing of spammers. Laws in Washington and Nevada impose civil fines.

*Usenet newsgroups are limited communities of Net users who congregate online to discuss specific topics.

This legislation has not been popular with everyone, however, and has led organizations such as the American Civil Liberties Union (ACLU), to complain about its unconstitutionality and threat to free speech and the First Amendment.

Like spamming, threatening electronic mail messages have become pervasive in the Internet space. Many of these messages are not taken as seriously as the one that was sent by a high school student from New Jersey, who made a death threat against President Clinton in an electronic mail message in early 1999. Using a school computer that provided an option to communicate with a contingent of the U.S. government, the student rapidly became the subject of a Secret Service investigation.

Similarly, in late 1998, a former investment banker was convicted on eight counts of aggravated harassment when he masqueraded as another employee and sent allegedly false and misleading Internet e-mail messages to top executives of his former firm.

Increasingly, businesses are establishing policy to inhibit employees from using company resources to perform unethical behavior on the Internet. In an early 1999 case, a California firm agreed to pay a former employee over $100,000 after she received harassing messages on the firm's electronic bulletin board, even though the company reported the incident to authorities and launched an internal investigation. The case is a not-so-subtle reminder that businesses are accountable for the actions of their employees, even actions performed on electronic networks.

Businesses have taken a stern position on employees surfing the Web, sending inappropriate messages, and downloading pornographic materials from the Internet. This is due to a negative impact on productivity, as well as the legal view that companies are liable for the actions of their employees. Many companies have established policies for appropriate use and monitoring of computers and computing resources, as well as etiquette on the Internet, or "Netiquette."

These policies are enhancements to the Internet Advisory Board's (Request for Comment) RFC 1087, "Internet Ethics," January 1989, which proposed that access to and use of the Internet is a privilege and should be treated as such by all users of the system. The IAB strongly endorsed the view of the Division Advisory Panel of the National Science Foundation Division of Network Communications Research and Infrastructure. That view is paraphrased below.

Any activity is characterized as unethical and unacceptable that purposely:

- Seeks to gain unauthorized access to the resources of the Internet
- Disrupts the intended use of the Internet
- Wastes resources (people, capacity, computers) through such actions
- Destroys the integrity of computer-based information
- Compromises the privacy of users
- Involves negligence in the conduct of Internet-wide experiments*

A sample "Appropriate Use of the Internet" policy is attached as Appendix A. Appendix B contains the partial contents of RFC 1855, "Netiquette Guidelines," a product of the Responsible Use of the Network (RUN) Working Group of the Internet Engineering Task Force (IETF).

In another twist on Internet electronic mail activity, in April 1999 Intel Corporation sued a former employee for doing a mass e-mailing to its 30,000 employees, criticizing the company over workers' compensation benefits. Intel claims the e-mail was an assault and form of trespass, as well as an improper use of its internal computer resources. The former employee contends that his e-mail messages are protected by the First Amendment. "Neither Intel nor I can claim any part of the Internet as our own private system as long as we are hooked up to this international network of computers," said Ken Hamidi in an e-mail to *Los Angeles Times* reporters. The case was not settled as of this writing ("Ruling is Due on Mass E-mail Campaign Against Intel," Greg Miller, *Los Angeles Times*, April 19, 1999).

Using electronic media to stalk another person is known as "cyber stalking." This activity is becoming more prevalent, and the law has seen fit to intercede by adding computers and electronic devices to existing stalking legislation. In the first case of cyber stalking in California, a Los Angeles resident, accused of using his computer to harass a woman who rejected his romantic advances, is the first to be charged under a new cyber stalking law that went into effect in 1998. The man was accused of forging postings on the Internet, on America Online (AOL), and other Internet services, so that the messages appeared to come from the victim. The message provided the woman's address and other identifying information, which resulted in at least six men visiting

* *Source*: RFC 1087, "Ethics and the Internet," Internet Advisory Board, January 1989.

EXHIBIT 157.2

Information Collected when You Send Us an E-Mail Message

When inquiries are e-mailed to us, we again store the text of your message and e-mail address information, so that we can answer the question that was sent in, and send the answer back to the e-mail address provided. If enough questions or comments come in that are the same, the question may be added to our Question and Answer section, or the suggestions are used to guide the design of our Web site.

We do not retain the messages with identifiable information or the e-mail addresses for more than 10 days after responding unless your communication requires further inquiry. If you send us an e-mail message in which you ask us to do something that requires further inquiry on our part, there are a few things you should know.

The material you submit may be seen by various people in our Department, who may use it to look into the matter you have inquired about. If we do retain it, it is protected by the Privacy Act of 1974, which restricts our use of it, but permits certain disclosures.

Also, e-mail is not necessarily secure against interception. If your communication is very sensitive, or includes personal information, you might want to send it by postal mail instead.

her home uninvited. The man was charged with one count of stalking, three counts of solicitation to commit sexual assault, and one count of unauthorized access to computers.

In another instance where electronic activity has been added to existing law, the legislation for gambling has been updated to include Internet gambling. According to recent estimates, Internet-based gambling and gaming has grown from about a $500 million-a-year industry in the late 1990s, to what some estimate could become a $10 billion-a-year enterprise by 2000. Currently, all 50 states regulate in-person gambling in some manner. Many conjecture that the impetus for the regulation of electronic gambling is financial, not ethical or legal.

Privacy on the Internet

For many years, American citizens have expressed fears of invasion of privacy, ever since they realized that their personal information is being stored on computer databases by government agencies and commercial entities. However, it is just of late that Americans are realizing that logging on to the Internet and using the World Wide Web threatens their privacy as well. Last year, the Center for Democracy and Technology (CDT), a Washington, D.C. advocacy group, reported that only one third of federal agencies tell visitors to their Web sites what information is being collected about them.

AT&T Labs conducted a study early last year, in which they discovered that Americans are willing to surrender their e-mail address online, but not much more than that. The study said that users are reluctant to provide other personal information, such as a phone number or credit card number.

The utilization of technology offers the opportunity for companies to collect specific items of information. For example, Microsoft Corporation inserts tracking numbers into its Word program documents. Microsoft's Internet Explorer informs Web sites when a user bookmarks them by choosing the "Favorites" option in the browser. In 1998, the Social Security Administration came very close to putting a site on line that would let anyone find out another person's earnings and other personal information. This flies in the face of the 1974 Privacy Act, which states that every agency must record "only such information about an individual as is relevant and necessary to accomplish a purpose of the agency required to be accomplished by statute or by executive order of the President."

There is a battle raging between privacy advocates and private industry aligned with the U.S. government. Privacy advocates relate the serious concern for the hands-off approach and lack of privacy legislation, claiming that citizens are being violated. Conversely, the federal government and private businesses, such as American Online, defend current attempts to rely on self-regulation and other less government-intrusive means of regulating privacy, for example, the adoption of privacy policies. These policies, which state intent for the protection of consumer privacy, are deployed to raise consumer confidence and increase digital trust. The CDT has urged the federal government to post privacy policies on each site's home page, such as is shown in Exhibit 157.2 from the Health and Human Services Web site from the National Institute of Health (www.nih.gov).

HHS Web Privacy Notice

(as of April 13, 1999)

Thank you for visiting the Department of Health and Human Services Web site and reviewing our Privacy Policy. Our Privacy Policy for visits to www.hhs.gov is clear:
We will collect no personal information about you when you visit our Web site unless you choose to provide that information to us.
Here is how we handle information about your visit to our Web site:

Information Collected and Stored Automatically

If you do nothing during your visit but browse through the website, read pages, or download information, we will gather and store certain information about your visit automatically. This information does not identify you personally. We automatically collect and store only the following information about your visit:

- The Internet domain (for example, "xcompany.com" if you use a private Internet access account, or "yourschool.edu" if you connect from a university's domain), and IP address (an IP address is a number that is automatically assigned to your computer whenever you are surfing the Web) from which you access our Web site
- The type of browser and operating system used to access our site
- The date and time you access our site
- The pages you visit
- If you linked to our Web site from another Web site, the address of that Web site

We use this information to help us make our site more useful to visitors — to learn about the number of visitors to our site and the types of technology our visitors use. We do not track or record information about individuals and their visits.

Links to Other Sites

Our Web site has links to other federal agencies and to private organizations. Once you link to another site, it is that site's privacy policy that controls what it collects about you.

Anonymity on the Internet

Besides a lack of privacy, the Internet promulgates a lack of identity. Users of the Internet are virtual, meaning that they are not speaking with, interacting with, or responding to others, at least not face to face. They sit behind their computer terminals in the comfort of their own home, office, or school. This anonymity makes it easy to masquerade as another, since there is no way of proving or disproving who you are or who you say you are.

Moreover, this anonymity lends itself to the venue of Internet chat rooms. Chat rooms are places on the Net where people congregate and discuss topics common to the group, such as sports, recreation, or sexuality. Many chat rooms provide support to persons looking for answers to questions on health, bereavement, or disease and, in this manner, can be very beneficial to society.

Conversely, chat rooms can be likened to sleazy bars, where malcontents go seeking prey. There have been too many occurrences of too-good-to-be-true investments that have turned out to be fraudulent. Too many representatives of the dregs of society lurk on the net, targeting the elderly or the innocent, or those who, for some unknown reason, make easy marks.

A recent *New Yorker* magazine ran a cartoon showing a dog sitting at a computer desk, the caption reading "On the Internet, no one knows if you're a dog." Although the cartoon is humorous, the instances where child molesters have accosted their victims by way of the Internet are very serious. Too many times, miscreants have struck up electronic conversations with innocent victims, masquerading as innocents themselves, only to lead them to meet in person with dire results. Unfortunately, electronic behavior mimics conduct that has always

occurred over phone lines, through the postal service, and in person. The Internet only provides an additional locale for intentionally malicious and antisocial behavior. We can only hope that advanced technology, as with telephonic caller ID, will assist law enforcement in tracking anonymous Internet "bad guys."

Attempts at self-regulation have not been as successful as advertised, and many question whether the industry can police itself. Meanwhile, there are those within the legal and judicial systems that feel more laws are the only true answer to limiting unethical and illegal activities on the Internet. How it will all play out is far from known at this point in time. The right to freedom of speech and expression has often been at odds with censorship. It is ironic, for example, that debates abound on the massive amounts of pornography available on the Internet, and yet, in early 1999, the entire transcript of the President Clinton impeachment hearings was published on the Net, complete with sordid details of the Monica Lewinsky affair.

Internet and the Law

The Communications Decency Act of 1996 was signed into law by President Clinton in early 1996 and has been challenged by civil libertarian organizations ever since. In 1997, the United States Supreme Court declared the law's ban on indecent Internet speech unconstitutional.

The Childrens' Internet Protect Act (S.97, January 1999), introduced before a recent Congress, requires "the installation and use by schools and libraries of a technology for filtering or blocking material on the Internet on computers with Internet access to be eligible to receive or retain universal service assistance."

Monitoring the Web

Additionally, many commercial businesses have seen the opportunity to manufacture software products that will provide parents the ability to control their home computers. Products such as Crayon Crawler, Family-Connect, and KidsGate are available to provide parents with control over what Internet sites their children can access, although products like WebSense, SurfControl and Webroot are being implemented by companies that choose to limit the sites their employees can access.

Summary

Technology is a double-edged sword, consistently presenting us with benefits and disadvantages. The Internet is no different. The Net is a powerful tool, providing the ability for global communications in a heartbeat; sharing information without boundaries; a platform for illicit and unethical shenanigans.

This chapter has explored the types of behavior demonstrated in cyberspace, antisocial behavior, which has led to many discussions about whether or not this activity can be inhibited by self-regulation or the introduction of tougher laws. Although we do not know how the controversy will end, we know it will be an interesting future in cyberspace.

Appendix A "Appropriate Use and Monitoring of Computing Resources"

Policy

The Company telecommunications systems, computer networks, and electronic mail systems are to be used only for business purposes and only by authorized personnel. All data generated with or on the Company's business resources are the property of the Company; and may be used by the Company without limitation; and may not be copyrighted, patented, leased, or sold by individuals or otherwise used for personal gain.
Electronic mail and voice mail, including pagers and cellular telephones, are not to be used to create any offensive or disruptive messages. The Company does not tolerate discrimination, harassment, or other offensive messages and images relating to, among other things, gender, race, color, religion, national origin, age, sexual orientation, or disability.

The Company reserves the right and will exercise the right to review, monitor, intercept, access, and disclose any business or personal messages sent or received on Company systems. This may happen at any time, with or without notice.
It is the Company's goal to respect individual privacy, while at the same time maintaining a safe and secure workplace. However, employees should have no expectation of privacy with respect to any Company computer or communication resources. Materials that appear on computer, electronic mail, voice mail, facsimile and the like, belong to the Company. Periodically, your use of the Company's systems may be monitored.
The use of passwords is intended to safeguard Company information, and does not guarantee personal confidentiality.
Violations of company policies detected through such monitoring can lead to corrective action, up to and including discharge.

Appendix B
Netiquette

RFC 1855
Netiquette Guidelines

Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Abstract

This document provides a minimum set of guidelines for Network Etiquette (Netiquette) which organizations may take and adapt for their own use. As such, it is deliberately written in a bulleted format to make adaptation easier and to make any particular item easy (or easier) to find. It also functions as a minimum set of guidelines for individuals, both users and administrators. This memo is the product of the Responsible Use of the Network (RUN) Working Group of the IETF.

1.0 Introduction

In the past, the population of people using the Internet had "grown up" with the Internet, were technically minded, and understood the nature of the transport and the protocols. Today, the community of Internet users includes people who are new to the environment. These "newbies" are unfamiliar with the culture and do not need to know about transport and protocols. To bring these new users into the Internet culture quickly, this Guide offers a minimum set of behaviors which organizations and individuals may take and adapt for their own use. Individuals should be aware that no matter who supplies their Internet access, be it an Internet Service Provider through a private account, or a student account at a University, or an account through a corporation, that those organizations have regulations about ownership of mail and files, about what is proper to post or send, and how to present yourself. Be sure to check with the local authority for specific guidelines.
We have organized this material into three sections: One-to-one communication, which includes mail and talk; One-to-many communications, which includes mailing lists and NetNews; and Information Services, which includes ftp, WWW, Wais, Gopher, MUDs and MOOs. Finally, we have a Selected Bibliography, which may be used for reference.

2.0 One-to-One Communication (Electronic Mail, Talk)

We define one-to-one communications as those in which a person is communicating with another person as if face-to-face: a dialog. In general, rules of common courtesy for interaction with people should be in force for any situation and on the Internet it is doubly important where, for example, body language and tone of voice must be inferred. For more information on Netiquette for communicating via electronic mail and talk, check references [1,23,25,27] in the Selected Bibliography.

2.1 User Guidelines

2.1.1 For Mail:

- Unless you have your own Internet access through an Internet provider, be sure to check with your employer about ownership of electronic mail. Laws about the ownership of electronic mail vary from place to place.
- Unless you are using an encryption device (hardware or software), you should assume that mail on the Internet is not secure. Never put in a mail message anything you would not put on a postcard.
- Respect the copyright on material that you reproduce. Almost every country has copyright laws.
- If you are forwarding or reposting a message you have received, do not change the wording. If the message was a personal message to you and you are reposting to a group, you should ask permission first. You may shorten the message and quote only relevant parts, but be sure you give proper attribution.
- Never send chain letters via electronic mail. Chain letters are forbidden on the Internet. Your network privileges will be revoked. Notify your local system administrator if your ever receive one.
- A good rule of thumb: Be conservative in what you send and liberal in what you receive. You should not send heated messages (we call these "flames") even if you are provoked. On the other hand, you should not be surprised if you get flamed and it is prudent not to respond to flames.
- In general, it is a good idea to at least check all your mail subjects before responding to a message. Sometimes a person who asks you for help (or clarification) will send another message which effectively says "Never Mind." Also make sure that any message you respond to was directed to you. You might be cc:ed rather than the primary recipient.
- Make things easy for the recipient. Many mailers strip header information which includes your return address. To ensure that people know who you are, be sure to include a line or two at the end of your message with contact information. You can create this file ahead of time and add it to the end of your messages. (Some mailers do this automatically.) In Internet parlance, this is known as a ".sig" or "signature" file. Your .sig file takes the place of your business card. (And you can have more than one to apply in different circumstances.)
- Be careful when addressing mail. There are addresses which may go to a group but the address looks like it is just one person. Know to whom you are sending.
- Watch "CCs" when replying. Do not continue to include people if the messages have become a two-way conversation.
- In general, most people who use the Internet do not have time to answer general questions about the Internet and its workings. Do not send unsolicited mail asking for information to people whose names you might have seen in RFCs or on mailing lists.
- Remember that people with whom you communicate are located across the globe. If you send a message to which you want an immediate response, the person receiving it might be at home asleep when it arrives. Give them a chance to wake up, come to work, and log in before assuming the mail didn't arrive or that they do not care.
- Verify all addresses before initiating long or personal discourse. It is also a good practice to include the word "long" in the subject header so the recipient knows the message will take time to read and respond to. Over 100 lines is considered "long".
- Know whom to contact for help. Usually you will have resources close at hand. Check locally for people who can help you with software and system problems. Also, know whom to go to if you receive anything questionable or illegal. Most sites also have "Postmaster" aliased to a knowledgeable user, so you can send mail to this address to get help with mail.
- Remember that the recipient is a human being whose culture, language, and humor have different points of reference from your own. Remember that date formats, measurements, and idioms may not travel well. Be especially careful with sarcasm.
- Use mixed case. UPPER CASE LOOKS AS IF YOU ARE SHOUTING.
- Use symbols for emphasis. That *is* what I meant. Use underscores for underlining. _War and Peace_ is my favorite book.

- Use smileys to indicate tone of voice, but use them sparingly. :-) is an example of a smiley (Look sideways). Do not assume that the inclusion of a smiley will make the recipient happy with what you say or wipe out an otherwise insulting comment.
- Wait overnight to send emotional responses to messages. If you have really strong feelings about a subject, indicate it via FLAME ON/OFF enclosures. For example:
  FLAME ON
  This type of argument is not worth the bandwidth it takes to send it. It is illogical and poorly reasoned. The rest of the world agrees with me.
- FLAME OFF
- Do not include control characters or non-ASCII attachments in messages unless they are MIME attachments or unless your mailer encodes these. If you send encoded messages make sure the recipient can decode them.
- Be brief without being overly terse. When replying to a message, include enough original material to be understood but no more. It is extremely bad form to simply reply to a message by including all the previous message: edit out all the irrelevant material.
- Limit line length to fewer than 65 characters and end a line with a carriage return.
- Mail should have a subject heading which reflects the content of the message.
- If you include a signature keep it short. Rule of thumb is no longer than four lines. Remember that many people pay for connectivity by the minute, and the longer your message is, the more they pay.
- Just as mail (today) may not be private, mail (and news) are (today) subject to forgery and spoofing of various degrees of detectability. Apply common sense "reality checks" before assuming a message is valid.
- If you think the importance of a message justifies it, immediately reply briefly to an e-mail message to let the sender know you got it, even if you will send a longer reply later.
- "Reasonable" expectations for conduct via e-mail depend on your relationship to a person and the context of the communication. Norms learned in a particular e-mail environment may not apply in general to your e-mail communication with people across the Internet. Be careful with slang or local acronyms.
- The cost of delivering an e-mail message is, on the average, paid about equally by the sender and the recipient (or their organizations). This is unlike other media such as physical mail, telephone, TV, or radio. Sending someone mail may also cost them in other specific ways like network bandwidth, disk space or CPU usage. This is a fundamental economic reason why unsolicited e-mail advertising is unwelcome (and is forbidden in many contexts).
- Know how large a message you are sending. Including large files such as Postscript files or programs may make your message so large that it cannot be delivered or at least consumes excessive resources. A good rule of thumb would be not to send a file larger than 50 kb. Consider file transfer as an alternative, or cutting the file into smaller chunks and sending each as a separate message.
- Do not send large amounts of unsolicited information to people.
- If your mail system allows you to forward mail, beware the dreaded forwarding loop. Be sure you have not set up forwarding on several hosts so that a message sent to you gets into an endless loop from one computer to the next to the next.

Selected Bibliography

This bibliography was used to gather most of the information in the sections above as well as for general reference. Items not specifically found in these works were gathered from the IETF-RUN Working Group's experience.

1. Angell, D. and B. Heslop, *The Elements of E-mail Style,* New York: Addison-Wesley, 1994.
2. Answers to Frequently Asked Questions about Usenet" Original author: jerry@eagle.UUCP (Jerry Schwarz) Maintained by: netannounce@deshaw.com (Mark Moraes) Archive-name: usenet-faq/part1
3. Cerf, V., "Guidelines for Conduct on and Use of Internet," at: http://www.isoc.org/policy/conduct/conduct.html

Computer Ethics

Peter S. Tippett

The computer security professional needs both to understand and to influence the behavior of everyday computer users. Traditionally, security managers have concentrated on building security into the system hardware and software, on developing procedures, and on educating end users about procedures and acceptable behavior. Now, the computer professional must also help develop the meaning of ethical computing and help influence computer end users to adopt notions of ethical computing into their everyday behavior.

Fundamental Changes to Society

Computer technology has changed the practical meaning of many important, even fundamental, human and societal concepts. Although most computer professionals would agree that computers change nothing about human ethics, computer and information technologies have caused and will pose many new problems. Indeed, computers have changed the nature and scope of accessing and manipulating information and communications. As a result, computers and computer communications will significantly change the nature and scope of many of the concepts most basic to society. The changes will be as pervasive and all encompassing as the changes accompanying earlier shifts from a society dependent on hunters and gatherers to one that was more agrarian to an industrial society.

Charlie Chaplin once observed, "The progress of science is far ahead of man's ethical behavior." The rapid changes that computing technology and the digital revolution have brought and will bring are at least as profound as the changes prompted by the industrial revolution. This time, however, the transformation will be compressed into a much shorter time frame.

It will not be known for several generations whether the societal changes that follow from the digital revolution will be as fundamental as those caused by the combination of easy transportation, pervasive and near-instantaneous news, and inexpensive worldwide communication brought on by the industrial and radio revolutions. However, there is little doubt that the digital age is already causing significant changes in ways that are not yet fully appreciated.

Some of those changes are bad. For example, combining the known costs of the apparent unethical and illegal uses of computer and information technology — factors such as telephone and PBX fraud, computer viruses, and digital piracy — amounts to several billion dollars annually. When these obvious problems are combined with the kinds of computing behavior that society does not yet fully comprehend as unethical and that society has not yet labeled illegal or antisocial, it is clear that a great computer ethics void exists.

No Sandbox Training

By the time children are six years old, they learn that eating grasshoppers and worms is socially unacceptable. Of course, six-year-olds would not say it quite that way. To express society's wishes, children say something more like: "Eeewwww!, Yich! Johnny, you are not going to eat that worm are you?"

As it turns out, medical science shows that there is nothing physically dangerous or wrong with eating worms or grasshoppers. Eating them would not normally make people sick or otherwise cause physical harm. But children quickly learn at the gut level to abhor this kind of behavior — along with a whole raft of other behavior. What is more, no obvious rule exists that leads to this gut-feeling behavior. No laws, church doctrine, school curriculum, or parental guides specifically address the issue of eating worms and grasshoppers. Yet, even without structured rules or codes, society clearly gives a consistent message about this. Adults take the concept as being so fundamental that it is called common sense.

By the time children reach the age of ten, they have a pretty clear idea of what is right and wrong, and what is acceptable and unacceptable. These distinctions are learned from parents, siblings, extended families, neighbors, acquaintances, and schools, as well as from rituals like holiday celebrations and from radio, television, music, magazines, and many other influences.

Unfortunately, the same cannot be said for being taught what kind of computing behavior is repugnant. Parents, teachers, neighbors, acquaintances, rituals, and other parts of society simply have not been able to provide influence or insight based on generations of experience. Information technology is so new that these people and institutions simply have no experience to draw on. The would-be teachers are as much in the dark as those who need to be taught.

A whole generation of computer and information system users exists. This generation is more than one hundred million strong and growing. Soon information system users will include nearly every literate individual on earth. Members of this new generation have not yet had their sandbox

training. Computer and information users, computer security professionals included, are simply winging it.

Computer users are less likely to know the full consequences of many of their actions than they would be if they could lean on the collective family, group, and societal experiences for guidance. Since society has not yet established much of what will become common sense for computing, individuals must actively think about what makes sense and what does not. To decide whether a given action makes sense, users must take into account whether the action would be right not only for themselves personally but also for their peers, businesses, families, extended families, communities, and society as a whole. Computer users must also consider short-term, mid-term, and long-term ramifications of each of the potential actions as they apply to each of these groups. Since no individual can conceivably take all of this into consideration before performing a given action, human beings need to rely on guides such as habit, rules, ritual, and peer pressure. People need to understand without thinking about it, and for that, someone needs to develop and disseminate ethics for the computer generation.

Computer security professionals must lead the way in educating the digital society about policies and procedures and behavior that clearly can be discerned as right or wrong. The education process involves defining those issues that will become gut feelings, common sense, and acceptable etiquette of the whole society of end users. Computer professionals need to help develop and disseminate the rituals, celebrations, habits, and beliefs for users.

In other words, they are the pivotal people responsible for both defining computer ethics and disseminating their understanding to the computer-using public.

COMMON FALLACIES OF THE COMPUTER GENERATION

The lack of early, computer-oriented, childhood rearing and conditioning has led to several pervasive fallacies that generally (and loosely) apply to nearly all computer and digital information users. The generation of computer users includes those from 7 to 70 years old who use computing and other information technologies. Like all fallacies, some people are heavily influenced by them, and some are less so. There are clearly more fallacies than those described here, but these are probably the most important. Most ethical problems that surface in discussions show roots in one or more of these fallacies.

The Computer Game Fallacy

Computer games like solitaire and game computers like those made by Nintendo and Sega do not generally let the user cheat. So it is hardly

surprising for computer users to think, at least subliminally, that computers in general will prevent them from cheating and, by extension, from otherwise doing wrong.

This fallacy also probably has roots in the very binary nature of computers. Programmers in particular are used to the precise nature that all instructions must have before a program will work. An error in syntax, a misplaced comma, improper capitalization, and transposed characters in a program will almost certainly prevent it from compiling or running correctly once compiled. Even non-programming computer users are introduced to the powerful message that everything about computers is exact and that the computer will not allow even the tiniest transgression. DOS commands, batch file commands, configuration parameters, macro commands, spreadsheet formulas, and even file names used for word processing must have precisely the right format and syntax, or they will not work.

To most users, computers seem entirely black and white — sometimes frustratingly so. By extension, what people do with computers seems to take on a black-and-white quality. But what users often misunderstand while using computers is that although the computer operates with a very strict set of inviolable rules, most of what people do with computers is just as gray as all other human interactions.

It is a common defense for malicious hackers to say something like "If they didn't want people to break into their computer at the [defense contractor], they should have used better security." Eric Corley, the publisher of the hacker's *2600 Magazine*, testified at hearings for the House Telecommunications and Finance Subcommittee (June 1993) that he and others like him were providing a service to computer and telecommunication system operators when they explored computer systems, found faults and weaknesses in the security systems, and then published how to break these systems in his magazine. He even had the audacity while testifying before Congress to use his handle, Emanuel Goldstein (a character from the book *1984*), never mentioning that his real name was Eric Corley.

He, and others like him, were effectively saying "If you don't want me to break in, make it impossible to do so. If there is a way to get around your security, then I should get around it in order to expose the problem."

These malicious hackers would never consider jumping over the four-foot fence into their neighbor's backyard, entering the kitchen through an open kitchen window, sitting in the living room, reading the mail, making a few phone calls, watching television, and leaving. They would not brag or publish that their neighbor's home was not secure enough, that they found a problem or loophole, or that it was permissible to go in because it was possible to do so. However, using a computer to perform analogous activities makes perfect sense to them.

The computer game fallacy also affects the rest of the members of the computer-user generation in ways that are a good deal more subtle. The computer provides a powerful one-way mirror behind which people can hide. Computer uses can be voyeurs without being caught. And if what is being done is not permissible, the thinking is that the system would somehow prevent them from doing it.

The Law-Abiding Citizen Fallacy

Recognizing that computers can't prevent everything that would be wrong, many users understand that laws will provide some guidance. But many (perhaps most) users sometimes confuse what is legal, which defines the minimum standard about which all can be justly judged, with what is reasonable behavior, which clearly calls for individual judgment. Sarah Gordon, one of the leaders of the worldwide hobbyist network FidoNet said, "In most places, it is legal to pluck the feathers off of a live bird, but that doesn't make it right to do it."

Similarly, people confuse things that they have a right to do with things that are right to do. Computer virus writers do this all the time. They say: "The First Amendment gives me the constitutional right to write anything I want, including computer viruses. Since computer viruses are an expression, and a form of writing, the constitution also protects the distribution of them, the talking about them, and the promotion of them as free speech."

Some people clearly take their First Amendment rights too far. Mark Ludwig has written two how-to books on creating computer viruses. He also writes a quarterly newsletter on the finer details of computer virus authors and runs a computer virus exchange bulletin board with thousands of computer viruses for the user's downloading pleasure. The bulletin board includes source code, source analysis, and tool kits to create nasty features like stealthing, encryption, and polymorphism. He even distributes a computer virus CD with thousands of computer viruses, a source code, and some commentary.

Nearly anyone living in the United States would agree that in most of the western world, people have the right to write almost anything they want. However, they also have the responsibility to consider the ramifications of their actions and to behave accordingly. Some speech, of course, is not protected by the constitution — like yelling "fire" in a crowded theater or telling someone with a gun to shoot a person. One would hope that writing viruses will become nonprotected speech in the future. But for now, society has not decided whether virus writing, distribution, and promotion should be violently abhorred or tolerated as one of the costs of other freedoms.

The Shatterproof Fallacy

How many times have computer novices been told "Don't worry, the worst you can do with your computer is accidentally erase or mess up a file — and even if you do that, you can probably get it back. You can't really hurt anything."

Although computers are tools, they are tools that can harm. Yet most users are totally oblivious to the fact that they have actually hurt someone else through actions on their computer. Using electronic-mail on the Internet to denigrate someone constitutes malicious chastisement of someone in public. In the nondigital world, people can be sued for libel for these kinds of actions; but on the Internet, users find it convenient to not be held responsible for their words.

Forwarding E-mail without at least the implied permission of all of its authors often leads to harm or embarrassment of participants who thought they were conferring privately. Using E-mail to stalk someone, to send unwanted mail or junk mail, and to send sexual innuendoes or other material that is not appreciated by the recipient all constitute harmful use of computers.

Software piracy is another way in which computer users can hurt people. Those people are not only programmers and struggling software companies but also end users who must pay artificially high prices for the software and systems they buy and the stockholders and owners of successful companies who deserve a fair return on their investment.

It is astonishing that a computer user would defend the writing of computer viruses. Typically, the user says, "My virus is not a malicious one. It does not cause any harm. It is a benign virus. The only reason I wrote it was to satisfy my intellectual curiosity and to see how it would spread." Such users truly miss out on the ramifications of their actions. Viruses, by definition, travel from computer to computer without the knowledge or permission of the computer's owner or operator.

Viruses are just like other kinds of contaminants (e.g., contaminants in a lake) except that they grow (replicate) much like a cancer. Computer users cannot know they have a virus unless they specifically test their computers or diskettes for it. If the neighbor of a user discovers a virus, then the user is obliged to test his or her system and diskettes for it and so are the thousand or so other neighbors that the user and the user's neighbors have collectively.

The hidden costs of computer viruses are enormous. Even if an experienced person with the right tools needs only 10 minutes to get rid of a virus — and even if the virus infects only 4 or 5 computers and only 10 or 20 floppy disks in a site (these are about the right numbers for a computer

virus incident in a site of 1000 computers), then the people at the site are obliged to check all 1,000 computers and an average of 35,000 diskettes (35 active diskettes per computer) to find out just which five computers are infected.

As of early 1995, there were demonstrably more than a thousand people actively writing, creating, or intentionally modifying the more than 6000 computer viruses that currently exist — and at least as many people knowingly participated in spreading them. Most of these people were ignorant of the precise consequences of their actions.

In 1993, there was a minor scandal in the IRS when clerical IRS employees were discovered pulling computerized tax returns of movie stars, politicians, and their neighbors — just for the fun of it. What is the harm? The harm is to the privacy of taxpayers and to the trust in the system, which is immeasurably damaged in the minds of U.S. citizens. More than 350 IRS employees were directly implicated in this scandal. When such large numbers of people do not understand the ethical problem, then the problem is not an isolated one. It is emblematic of a broad ethical problem that is rooted in widely held fallacies.

The shatterproof fallacy is the pervasive feeling that what a person does with a computer could hurt at most a few files on the machine. It stems from the computer generation's frequent inability to consider the ramifications of the things we do with computers before we do them.

The Candy-from-a-Baby Fallacy

Guns and poison make killing easy (i.e., it can be done from a distance with no strength or fight) but not necessarily right. Poisoning the water supply is quite easy, but it is beyond the gut-level acceptability of even the most bizarre schizophrenic.

Software piracy and plagiarism are incredibly easy using a computer. Computers excel at copying things, and nearly every computer user is guilty of software piracy. But just because it is easy does not mean that it is right.

Studies by the Software Publisher's Association (SPA) and Business Software Alliance (BSA) show that software piracy is a multibillion dollar problem in the world today — clearly a huge problem.

By law and by any semblance of intellectual property held both in Western societies and most of the rest of the world, copying a program for use without paying for it is theft. It is no different than shoplifting or being a stowaway on an airliner, and an average user would never consider stealing a box of software from a computer store's display case or stowing away on a flight because the plane had empty seats.

The Hacker's Fallacy

The single most widely held piece of The Hacker's Ethic is "As long as the motivation for doing something is to learn and not to otherwise gain or make a profit, then doing it is acceptable." This is actually quite a strong, respected, and widely held ethos among people who call themselves non-malicious hackers.

To be a hacker, a person's primary goal must be to learn for the sake of learning — just to find out what happens if one does a certain thing at a particular time under a specific condition (Emmanuel Goldstein, *2600 Magazine,* Spring 1994). Consider the hack on Tonya Harding (the Olympic ice skater who allegedly arranged to have her archrival, Nancy Kerrigan, beaten with a bat). During the Lillehammer Olympics, three U.S. newspaper reporters, with the *Detroit Free Press, San Jose Mercury News,* and *The New York Times,* discovered that the athletes' E-mail user IDs were, in fact, the same as the ID numbers on the backs of their backstage passes. The reporters also discovered that the default passwords for the Olympic Internet mail system were simple derivatives of the athlete's birthdays. Reporters used this information to gain access to Tonya Harding's E-mail account and discovered that she had 68 messages. They claim not to have read any of them. They claim that no harm was done, nothing was published, no privacy was exploited. As it happens, these journalists were widely criticized for their actions. But the fact is, a group of savvy, intelligent people thought that information technology changed the ground rules.

The Free Information Fallacy

There is a common notion that information wants to be free, as though it had a mind of its own. The fallacy probably stems from the fact that once created in digital form, information is very easy to copy and tends to get distributed widely. The fallacy totally misses the point that the wide distribution is at the whim of people who copy and disseminate data and people who allow this to happen.

ACTION PLAN

The following procedures can help security managers encourage ethical use of the computer within their organizations:

- Developing a corporate guide to computer ethics for the organization.
- Developing a computer ethics policy to supplement the computer security policy.
- Adding information about computer ethics to the employee handbook.
- Finding out whether the organization has a business ethics policy, and expanding it to include computer ethics.
- Learning more about computer ethics and spreading what is learned.

- Helping to foster awareness of computer ethics by participating in the computer ethics campaign.
- Making sure the organization has an E-mail privacy policy.
- Making sure employees know what the E-mail policy is.

Exhibits 1 through 6 contain sample codes of ethics for end users that can help security managers develop ethics policies and procedures.

RESOURCES

The following resources are useful for developing computer-related ethics codes and policies.

Computer Ethics Institute

The Computer Ethics Institute is a non-profit organization concerned with advancing the development of computers and information technologies within ethical frameworks. Its constituency includes people in business, the religious communities, education, public policy, and computer professions. Its purpose includes the following:

- The dissemination of computer ethics information.
- Policy analysis and critique.
- The recognition and critical examination of ethics in the use of computer technology.
- The promotion of identifying and applying ethical principles for the development and use of computer technologies.

In 1991 the Computer Ethics Institute held its first National Computer Ethics Conference in Washington, D.C. The conference theme was "In Pursuit of a 'Ten Commandments' of Computer Ethics." These commandments were drafted by Dr. Ramon C. Barquin, founder and president of the Institute, as a working document for that conference. Since then, they have been among the most visible guidelines for computer ethics. The following are the ten commandments:

1. Thou shalt not use a computer to harm other people.
2. Thou shalt not interfere with other people's computer work.
3. Thou shalt not snoop around in other people's computer files.
4. Thou shalt not use a computer to steal.
5. Thou shalt not use a computer to bear false witness.
6. Thou shalt not copy or use proprietary software for which you have not paid.
7. Thou shalt not use other people's computer resources without authorization or proper compensation.
8. Thou shalt not appropriate other people's intellectual output.
9. Thou shalt think about the social consequences of the program you are writing or the system you are designing.
10. Thou shalt use a computer in ways that ensure consideration and respect for your fellow humans.

Exhibit 1. The Ten Commandments of Computer Ethics

In an effort to define responsible computing behavior in terms that are easy to grasp, the Working Group on Computer Ethics created the End User's Basic Tenets of Responsible Computing. These tenets are not intended as a panacea for the myriad of complex information ethics dilemmas; rather, they are intended to address many of the day-to-day problems faced by individual end users.

Responsible and ethical computing is not a black and white issue. However, many problems can be avoided by abiding by the following basic tenets:

1. I understand that just because something is legal, it isn't necessarily moral or right.
2. I understand that people are always the ones ultimately harmed when computers are used unethically. The fact that computers, software, or a communications medium exists between me and those harmed does not in any way change my moral responsibility toward my fellow humans.
3. I will respect the rights of authors, including authors and publishers of software as well as authors and owners of information. I understand that just because copying programs and data is easy, it is not necessarily right.
4. I will not break into or use other people's computers or read or use their information without their consent.
5. I will not write or knowingly acquire, distribute, or allow intentional distribution of harmful software like bombs, worms, and computer viruses.

Exhibit 2. The End User's Basic Tenets of Responsible Computing

---

The National Conference on Computing and Values proposed four primary values for computing. These were originally intended to serve as the ethical foundation and guidance for computer security. However, they seem to provide value guidance for all individuals who create, sell, support, use, or depend upon computers. That is, they suggest the values that will tend to improve and stabilize the computer and information world and to make these technologies and systems work more productively and appropriately for society.

The four primary values state that we should strive to:

1. Preserve the public trust and confidence in computers.
2. Enforce fair information practices.
3. Protect the legitimate interests of the constituents of the system.
4. Resist fraud, waste, and abuse.

Exhibit 3. Four Primary Values

In January 1989, the Internet Activities Board (IAB) published a document called Ethics and the Internet (RFC 1087). It proposes that access to and use of the Internet is a privilege and should be treated as such by all users of this system. The IAB "strongly endorses the view of the Division Advisory Panel of the National Science Foundation Division of Network, Communications Research and Infrastructure." That view is paraphrased here. Any activity is characterized as unethical and unacceptable that purposely:

- Seeks to gain unauthorized access to the resources of the Internet.
- Disrupts the intended use of the Internet.
- Wastes resources (people, capacity, computer) through such actions.
- Destroys the integrity of computer-based information.
- Compromises the privacy of users.
- Involves negligence in the conduct of Internetwide experiments.

Exhibit 4. Unacceptable Internet Activities

Donn Parker, who is with SRI International and is the author of "Ethical Conflicts in Information and Computer Science, Technology and Business" (QED Information Sciences, Inc.), defined several principles for resolving ethical conflicts. The following summarizes this work:

You are probably aware of the obvious unethical information activities you should avoid, such as violating others' privacy by accessing their computers and causing others losses by giving away copies of the software others own or sell. But how do you deal with the really tough problems of deciding the best action in complex or unclear situations where a decision may be okay in one respect but not in another? These are the more difficult decisions to make. The following principles of ethical information conduct and examples may help you as a periodic review to make fairer decisions when needed or as a checklist for a methodical approach to solve a problem and reach a decision. You may not remember all of these principles on every occasion, but reading them now and every once in a while or having them handy when making a decision can help you through a difficult process.

1. Try to make sure that those people affected are aware of your planned actions and that they don't disagree with your intentions even if you have rights to do these things (informed consent).
2. Think carefully about your possible alternative actions and select the most beneficial necessary one that would cause the least or no harm under the worst circumstances (higher ethic in the worst case).
3. Consider that an action you take on a small scale or by you alone might result in significant harm if carried out on a larger scale or by many others (change of scale).
4. As a person who owns or is responsible for information, always make sure that the information is reasonably protected and that ownership of it and rights to it are clear to all users (owners' conservation of ownership).
5. As a person who uses information, always assume it is owned by others and their interests must be protected unless you explicitly know it is public or you are free to use it in the way you wish (users' conservation of ownership).

Exhibit 5. Considerations for Conduct

In 1973 the Secretary's Advisory Committee on Automated Personal Data Systems for the U.S. Department of Health, Education & Welfare recommended the adoption of a "Code of Fair Information Practices" to secure the privacy and rights of citizens. The Code is based on four principles:

1. There must be no personal data record-keeping systems whose very existence is secret.
2. There must be a way for a person to find out what information about the person is in a record and how it is used.
3. There must be a way for a person to prevent information about the person that was obtained for one purpose from being used or made available for other purposes without the person's consent.
4. Any organization creating, maintaining, using, or disseminating records of identifiable personal data must assure the reliability of the data for their intended use and must take precautions to prevent misuses of the data.

Exhibit 6. The Code of Fair Information Practices

To meet these purposes, the Computer Ethics Institute conducts seminars, convocations, and the annual National Computer Ethics Conference. The Institute also supports the publication of proceedings and the development and publication of other research. In addition, the Institute participates in projects with other groups with similar interests. The following are ways to contact the institute:

Dr. Patrick F. Sullivan
Executive Director
Computer Ethics Institute
P.O. Box 42672
Washington, D.C. 20015
Voice and fax: 301-469-0615
psullivan@brook.edu

Internet Listserve:cei-1@listserv.american.edu

This is a listserv on the Internet hosted by American University in Washington, D.C., on behalf of the Computer Ethics Institute. Electronic mail sent to this address is automatically forwarded to others interested in computer ethics and in activities surrounding the Computer Ethics Institute. To join the list, a person should send E-mail to:

listserv@american.edu

The subject field should be left blank. The message itself should say:

subscribe cei-1 <yourname>

The sender will receive postings to the list by E-mail (using the return address from the E-mail site used to send the request).

The National Computer Ethics and Responsibilities Campaign (NCERC)

The NCERC is a campaign jointly run by the Computer Ethics Institute and the National Computer Security Association. Its goal is to foster computer ethics awareness and education. The campaign does this by making tools and other resources available for people who want to hold events, campaigns, awareness programs, seminars, and conferences or to write or communicate about computer ethics.

The NCERC itself does not subscribe to or support a particular set of guidelines or a particular viewpoint on computer ethics. Rather, the Campaign is a nonpartisan initiative intended to foster increased understanding of the ethical and moral issues peculiar to the use and abuse of information technologies.

The initial phase of the NCERC was sponsored by a diverse group of organizations, including (alphabetically) The Atterbury Foundation, The Boston Computer Society, The Business Software Alliance, CompuServe,

The Computer Ethics Institute, Computer Professionals for Social Responsibility, Merrill Lynch, Monsanto, The National Computer Security Association, Software Creations BBS, The Software Publisher's Association, Symantec Corporation, and Ziff-Davis Publishing. The principal sponsor of the NCERC is the Computer Ethics Institute.

Other information about the campaign is available on CompuServe (GO CETHICS), where a repository of computer privacy, ethics and similar tools, codes, texts, and other materials are kept.

Computer Ethics Resource Guide

The Resource Guide to Computer Ethics is available for $12. (Send check or credit card number and signature to: NCERC, 10 S. Courthouse Ave., Carlisle, PA, 17013, or call 717-240-0430 and leave credit card information as a voice message.) The guide is meant as a resource for those who wish to do something to increase the awareness of and discussion about computer ethics in their workplaces, schools, universities, user groups, bulletin boards, and other areas.

The National Computer Security Association

The National Computer Security Association (NCSA) provides information and services involving security, reliability, and ethics. NCSA offers information on the following security-related areas: training, testing, research, product certification, underground reconnaissance, help desk, and consulting services. This information is delivered through publications, conferences, forums, and seminars — in both traditional and electronic formats. NCSA manages a CompuServe forum (CIS: GO NCSA) that hosts private online training and seminars in addition to public forums and libraries addressing hundreds of issues concerning information and communications security, computer ethics, and privacy.

The information about computer ethics that is not well suited to electronic distribution can generally be obtained through NCSA's InfoSecurity Resource Catalog, which provides one-stop-shopping for a wide variety of books, guides, training, and tools. (NCSA: 10 S. Courthouse Ave., Carlisle, PA, 17013, 717-258-1816).

SUMMARY

Computer and information technologies have created many new ethical problems. Compounding these problems is the fact that computer users often do not know the full consequences of their behavior.

Several common fallacies cloud the meaning of ethical computing. For example, many computer users confuse behavior that they have a right to perform with behavior that is right to perform and fail to consider the

ramifications of their actions. Another fallacy that is widely held by hackers is that as long as the motivation is to learn and not otherwise profit, any action using a computer is acceptable.

It is up to the system managers to destroy these fallacies and to lead the way in educating end users about policies and procedures and behavior that can clearly be discerned as right or wrong.

4. Dern, D., *The Internet Guide for New Users,* New York: McGraw-Hill, 1994.
5. "Emily Postnews Answers Your Questions on Netiquette" Original author: brad@looking.on.ca (Brad Templeton) Maintained by: netannounce@deshaw.com (Mark Moraes) Archive-name: emily-postnews/part1
6. Gaffin, A., *Everybody's Guide to the Internet,* Cambridge, Mass., MIT Press, 1994.
7. "Guidelines for Responsible Use of the Internet" from the US House of Representatives gopher, at: gopher://gopher.house.gov:70/OF-1%3a208%3aInternet%20Etiquette
8. How to find the right place to post (FAQ) by buglady@bronze.lcs.mit.edu (Aliza R. Panitz) Archive-name: finding-groups/general
9. Hambridge, S. and J. Sedayao, "Horses and Barn Doors: Evolution of Corporate Guidelines for Internet Usage," LISA VII, Usenix, November 1-5, 1993, pp. 9-16. ftp://ftp.intel.com/pub/papers/horses.ps or horses.ascii>
10. Heslop, B. and D. Angell, *The Instant Internet Guide: Hands-on Global Networking,* Reading, Mass., Addison-Wesley, 1994.
11. Horwitz, S., "Internet Etiquette Tips," ftp://ftp.temple.edu/pub/info/help-net/netiquette.infohn
12. Internet Activities Board, "Ethics and the Internet," RFC 1087, IAB, January 1989. ftp://ds.internic.net/rfc/rfc1087.txt
13. Kehoe, B., *Zen and the Art of the Internet: A Beginner's Guide*, Netiquette information is spread through the chapters of this work. 3rd ed. Englewood Cliffs, NJ., Prentice-Hall, 1994.
14. Kochmer, J., *Internet Passport: NorthWestNet's Guide to Our World Online,* 4th ed. Bellevue, WA, NorthWestNet, Northwest Academic Computing Consortium, 1993.
15. Krol, Ed, *The Whole Internet: User's Guide and Catalog,* Sebastopol, CA, O'Reilly & Associates, 1992.
16. Lane, E. and C. Summerhill, *Internet Primer for Information Professionals: A Basic Guide to Internet Networking Technology,* Westport, CT, Meckler, 1993.
17. LaQuey, T. and J. Ryer, The Internet companion, Chapter 3 in *Communicating with People,* pp 41-74. Reading, MA, Addison-Wesley, 1993.
18. Mandel, T., "Surfing the Wild Internet," SRI International Business Intelligence Program, Scan No. 2109. March, 1993. gopher://gopher.well.sf.ca.us:70/00/Communications/surf-wild
19. Martin, J., "There's Gold in them thar Networks! or Searching for Treasure in all the Wrong Places," FYI 10, RFC 1402, January 1993. ftp://ds.internic.net/rfc/rfc1402.txt
20. Pioch, N., "A Short IRC Primer," Text conversion by Owe Rasmussen. Edition 1.1b, February 28, 1993. http://www.kei.com/irc/IRCprimer1.1.txt
21. Polly, J., "Surfing the Internet: an Introduction," Version 2.0.3. Revised May 15, 1993. ftp://ftp.nysernet.org/pub/resources/guides/surfing.2.0.3.txt
22. "A Primer on How to Work With the Usenet Community" Original author: chuq@apple.com (Chuq Von Rospach) Maintained by: netannounce@deshaw.com (Mark Moraes) Archive-name: usenet-primer/part1
23. Rinaldi, A., "The Net: User Guidelines and Netiquette," September 3, 1992. http://www.fau.edu/rinaldi/net/index.htm
24. "Rules for posting to Usenet" Original author: spaf@cs.purdue.edu (Gene Spafford) Maintained by: netannounce@deshaw.com (Mark Moraes) Archive-name: posting-rules/part1
25. Shea, V., *Netiquette,* San Francisco: Albion Books, 1994?.
26. Strangelove, M., with A. Bosley, "How to Advertise on the Internet," ISSN 1201-0758.
27. Tenant, R., "Internet Basics," ERIC Clearinghouse of Information Resources, EDO-IR-92-7. September, 1992. gopher://nic.merit.edu:7043/00/introducing.the.Internet/Internet.basics.eric-digest gopher://vega.lib.ncsu.edu:70/00/library/reference/guides/tennet
28. Wiggins, R., *The Internet for Everyone: A Guide for Users and Providers,* New York, McGraw-Hill, 1995.

Domain 10
Physical Security

The Physical Security Domain discusses the importance of physical security in the protection of valuable information assets of the business enterprise. It provides protection techniques for the entire facility, from the outside perimeter to the inside office space, including the data center or server room.

In the early days of computers, much of the security focus was built on providing physical security protections. Think of the data center that contained the mainframe servers and all the information processed and stored on the system. In this environment, the majority of the protections were for physical protection of that one area, such as restricting personnel from the area, enforcing physical access controls with locks and alarms, and implementing environmental controls to ensure the equipment was protected from heat and moisture. The advent of distributed systems changed this focus; resources and information were now in various places within the organization, and in many cases, not even contained within the building. For example, mobile devices, such as laptops and personal digital assistants, provided the ability to carry information outside a limiting physical environment.

According to many information system security surveys, the majority of threats occur from insiders — that is, those individuals who have physical access to their own resources. Because of this, physical security is just as relevant today as it was 30 years ago. It is still necessary to protect server rooms by limiting access and installing appropriate locks.

Another factor impacting physical security is the new government and private-sector initiatives to protect critical infrastructures, such as power and water supplies. Because information system assets require some type of power source to operate, the need for clean, constant power is a primary physical security concern. Threats to infrastructures are evolving and pose different types of threats. Although this may appear to be dramatic, chemical and biological threats have become increasingly more viable methods of attack.

One of the challenges for information system security professionals is to understand the security challenges associated with the physical environment. Although physical security is documented according to some specific technologies, such as closed-circuit television (CCTV) and alarm systems, there has not been much literature that combines the physical security field with the information system security field. There is also a dichotomy between the "traditional" security professionals who focus primarily on personnel and access controls and the information system security professionals who focus on logical controls. Many organizations still struggle for control over who will provide security — the traditional security divisions or the information management divisions. This lack of coordination and, in many cases, political maneuvering, has created difficulties for organizations to accomplish goals. However, as most security professionals will note, if both sides (security and information management) begin to work together, they will realize that indeed their goals are the same — and what is needed is better communication and coordination about how to achieve those goals. That is, by capitalizing on the strength and knowledge of both functions, they will achieve the goals of information system security — protecting the organization's valuable resources.

Although the challenges have changed along with the technologies, physical security still plays a critical role in protecting the resources of an organization. It requires solidly constructed buildings, emergency preparedness, adequate environmental protection, reliable power supplies, appropriate climate control, and external and internal protection from intruders.

Contents

158

Physical Security: A Foundation for Information Security

Christopher Steinke, CISSP

Physical security can be defined as the measures taken to ensure the safety and material existence of something or someone against theft, espionage, sabotage, or harm. In the context of information security, this means about information, products, and people.

Physical security is the oldest form of protection. For ages, people have been protecting themselves from harm and their valuables from theft or destruction. In the past, physical security was all the protection someone needed to have safety. However, with technology, physical security alone is not effective. Information security is an approach that deploys many different layers of security to achieve its goal; hence the phrase "security in layers." With the common acceptance that nothing is 100 percent secure, information security uses the depth of its layers to achieve the highest form of security. A weakness in any one of these layers will cause security to break. Physical protection is the first step in the layered approach of information security. If it is nonexistent, weak, or exercised in malpractice, information security will fail.

Approaching Physical Security

Physical security is a continuous process that cannot be approached in an unpremeditated manner. The approach must be consistent with the goals of the organization and be applied in accordance with the standards and guidelines set forth in the information security policy.

Because there is little change in the world of physical security (at least not as quickly as the rest of the controls within information security), it is often considered to be boring or unimportant. This misunderstanding often causes physical security to be neglected or practiced haphazardly. Typically, the greatest weakness of any information security control is not the control itself, but the improper application of a control. Physical security must be approached with the same energy, focus, and seriousness as any other information security control. In fact, security controls must be approached and applied in a consistent and predetermined manner to achieve predictable, repeatable, and effective information security.

Locks, guards, surveillance cameras, and identification badges are merely the tools and equipment of physical security. To plan and design physical security, the following questions should be answered:

- What are you protecting?
- How important is the information being protected (in terms of economic, political, or public safety)?
- For whom are you protecting and what is more important to them? Confidentiality, integrity, or availability?
- What and who are you protecting it from?

Granted, not all places need the physical security of Fort Knox (who would want to work there?), but physical security should be applied in proportion to the importance and sensitivity of the people and information it

protects. This chapter discusses the risks posed by common threats and vulnerabilities in information security, and how good physical security can provide a foundation for addressing those risks.

Psychology of Physical Security

When planning and designing physical security, keep in mind that it is as much psychological as it is physical. It is important to consider the advantages that the psychological impact can have. If one can design physical security in such a way as to make it highly visible (while safeguarding the details), one can announce that your organization is well guarded, rendering it less of a target to threatening activity. This is an indirect way to eliminate the desire to commit a crime against that organization. The effectiveness of physical security, as with any security control, is measured in terms of eliminating the opportunity; the psychology of physical security is measured in terms of eliminating the desire.

Facility Physical Security

The diversity of the modern workplace often makes it impractical to establish universal, rigid physical security standards. Nonetheless, adequate physical security at every location is necessary for achieving a complete, secure environment. This chapter section outlines the types of facilities, how they differ, and ways to approach physical security for each.

Facility Classification

Facilities can be grouped into one of these general classifications:

- *Owned facility.* Owned facilities are probably the simplest structure to maintain physical security. The ease of security management is inherent, due to the occupant having complete administrative control over the facility. This allows the flexibility to implement whatever type of physical security control, in any fashion, the owner/occupant feels will accomplish their protection goals. The main downfall of an owned facility is that the owner/occupant must take complete responsibility if physical security fails. A good example of an owned facility is a large corporate headquarters.
- *Nonowned facility.* Nonowned facilities can be a little more challenging to physically protect. The occupant and the owner will have their own lists of responsibilities that hold them liable if physical security fails. For example, if a water pipe bursts and floods a computer room, the occupant may hold the owner liable for the damages if it is discovered that the owner did not adequately maintain the plumbing. In this case, nonowned facilities may offer the advantage of legal recourse for failed physical security. Examples of nonowned facilities are buildings an occupant leases but does not own.
- *Shared facility.* Shared facilities are probably the most diverse and threatening of facilities to occupy, yet they account for the majority of structures. These facilities have more than one occupant, with some of the occupants possibly being competitors. Because the facility must provide equal access to all occupants (in certain areas), physical security becomes very challenging. Good examples of shared facilities could be nonowned facilities with multiple occupants, central offices, and co-locations.

When classifying facilities, one takes the first step in developing a strategy for risk mitigation. By understanding the threats that may be inherent to certain facilities, one gains insight into protecting against the risks. Because some facilities may fit more than one classification description, one is not bound by strict adherence to this classification scheme. What one should then be aware of are any new inherent strengths and weaknesses that these hybrid classes might create.

Facility Location

Not only should one be concerned with what kind of facility one occupies, but also the location. A particular location may harbor more threats than another. Below are some location-based threats to consider when choosing an area for one's facility:

- *Vulnerability to crime, riots, and terrorism.* Research crime and terrorism statistics for each location being considered. If the location of the facility is in an area that is frequented by these activities, the

chances of physical security being breached increases. For example, frequent demonstrations or riots near a facility could erupt into random acts of violence (e.g., fires, crime, etc.) that may threaten the facility, its employees, and possibly its customers. Even in information security, the protection and safety of people should always come before anything else.

- *Adjacent buildings and businesses*. This issue relates to the previously discussed classification of facilities (particularly shared facilities) and the previous issue of crime and riot vulnerability. It is good practice to know who one's neighbors are and what they do. For example, one may not want to locate a corporate data center next to a competitor, a nuclear power plant, or a freeway or railway that is a route for hazardous chemical transportation. Also, these concerns come to mind about connected buildings. Are their physical security controls as strong as yours? Can someone get into the facility if they break into an adjacent building? What about the roof? These should all be in the forefront of one's mind when choosing a location.
- *Emergency support response*. This is simply defined as the time it takes emergency support (i.e., fire, police, and medical personnel) to reach the facility. Know the mileage and time the driving distance (during the heaviest traffic) from emergency support locations to the facility. This information allows one to implement physical security measures that not only will detect and deter, but also delay and minimize damage or harm until emergency support arrives.
- *Environmental support*. Environmental support is the clean air, water, and power that service the facility. Ensure that the location has room for growth in all of these areas. In particular, for high-availability facilities, look for locations from which to draw from two separate power grids.
- *Vulnerability to natural disasters*. Check local geological and weather statistics for patterns of natural disasters in preferred location(s) for the past 100 years. Granted, natural disasters cannot be predicted or totally avoided, but one can minimize their effect by choosing a location where such disasters are less likely to occur.

Facility Threats and Controls

From the previous discussion, one sees how certain locations can harbor more or fewer threats. What follows here is a list of threats and controls in their basic forms. This is to demonstrate that if one can eliminate a threat at its root, one can effectively eliminate several others at the same time. But also notice that the opposite can happen when one threat manifests another. The controls are simple and basic in nature, but keep in mind that controls, as a whole, should be able to deter, detect, delay, and react to a given threat. There are three classes of threats, those being natural, man-made, and environmental failure.

Natural Threats

Good physical security has a psychological advantage against some threats. Unfortunately, natural threats are not one of them. This threat cannot be deterred or discouraged. At one time or another, Mother Nature will threaten the facility. The only option is to implement controls that will minimize the impact and facilitate a quick recovery. Natural threats and some of their controls include:

- Fire causes the following risks:
  - — Heat
  - — Smoke
  - — Suppression agent (e.g., fire extinguishers and water) damage
- Fire controls include:
  - — Installing smoke detectors near equipment
  - — Installing fire extinguishers and training employees in their proper use
  - — Using gaseous (nonliquid) extinguishing systems near information systems
  - — Conducting regular fire evacuation exercises
  - — Storing all backup media offsite (with a bonded third party)
  - — Developing and exercising a disaster recover plan
- Severe Weather causes the following risks:
  - — Lightning

  - Heavy winds
  - Hail
  - Flooding
- Severe weather controls include:
  - Monitoring weather conditions
  - Keeping equipment in areas that are weather-proofed and capable of withstanding strong winds
    - Ensuring equipment is properly grounded
  - Installing surge suppressors and uninterruptible power supplies (UPS) or diesel generators
  - Installing raised flooring
  - Conducting regular weather evacuation exercises
  - Storing all backup media offsite (with a bonded third party)
  - Developing and exercising a disaster recovery plan
- Earthquakes are particularly dangerous because of their ability to spur other natural disasters, such as fires. In addition to collateral damage from quake-induced fires, some additional risks include:
  - Limited or no response from emergency agencies
  - Permanent structural physical damage to facilities and information systems
  - Nullify threat controls (e.g., disables fire-suppression capability)
  - Personnel evacuation is limited
- Earthquake controls include:
  - Keeping information systems equipment off elevated surfaces (without proper mounting)
  - Keeping information systems equipment away from glass windows
  - Installing earthquake-proof or antivibration devices on equipment and infrastructure
  - Conducting routine earthquake drills
  - Storing all backup media offsite (with a bonded third party)
  - Developing and exercising a disaster recovery plan

Natural threats are not always the dramatic events listed above. They can often take a much more subtle and unforeseen form. An example of this is the exposure to dry heat, moisture, and light winds over time. These less-severe threats may not be cause for immediate alarm, yet one should be aware of their potential impact.

Man-Made Threats

The second threat class is called man-made. This type of threat is often the most dynamic and challenging, due to ties in human nature. This is drawn from a conclusion that there are three motivating agents of man-made threats, those being malice, opportunity, and accidental. Man-made threats and some of the controls include:

- Theft/fraud causes the following risks:
  - Reduction or loss of information systems capabilities
  - Loss of sensitive information or trade secrets
  - Loss of revenue
- Theft/fraud controls include:
  - Posted signs that state the premises are monitored and persons may be inspected upon leaving or entering the facility
  - Visible closed circuit television cameras (CCTVs)
  - Security- and safety-conscious employees
  - Identification badges
  - Guards
  - Minimizing the use of location signs
  - Routine audits
  - Good inventory control practices
  - Good lock and key practices
  - Insurance

  - — Separation of duties/job rotation
  - — Employee hiring/termination practices
- Espionage causes the following risks:
  - — Loss of sensitive information or trade secrets
  - — Loss of competitive advantage
  - — Loss of revenue
- Espionage controls include:
  - — Posted signs that state the premises are monitored and persons may be inspected upon leaving or entering the facility
  - — Visible closed circuit television cameras (CCTVs)
  - — Security- and safety-conscious employees
  - — Identification badges
  - — Minimizing the use of location signs
  - — Guards
  - — Employee hiring/termination practices
  - — Separation of duties/job rotation
  - — Routine audits
- Sabotage causes the following risks:
  - — Reduction or loss of information systems capabilities
  - — Loss of sensitive information or trade secrets
  - — Loss of revenue
- Sabotage controls include:
  - — Posted signs that state the premises are monitored and persons may be inspected upon leaving or entering the facility
  - — Visible closed circuit television cameras (CCTVs)
  - — Security- and safety-conscious employees
  - — Minimizing the use of location signs
  - — Identification badges
  - — Guards
  - — Insurance
  - — Separation of duties/job rotation
- Workplace violence causes the following risks:
  - — Harm or death to employees
  - — Loss of productivity
  - — Loss of revenue
- Workplace violence controls include:
  - — Posted signs that state the premises are monitored and persons may be inspected upon leaving or entering the facility
  - — Visible closed circuit television cameras (CCTVs)
  - — Security- and safety-conscious employees
  - — Awareness of warning signs
  - — Guards
  - — Employee hiring/termination practices

The ingenuity and adaptive nature of the human mind makes man-made threats difficult to control. An organization must maintain vigilance with its protection program by conducting routine assessments on the controls implemented against these threats.

Environmental Threats

The third threat class is labeled environmental threats. Environmental controls are important to the operation and safeguarding of information and its systems. Without clean air, water, power, and reliable climate controls, information systems would suffer inconsistent performance or complete failure.

- Climate failure causes the following risks:
  - Equipment and infrastructure malfunction or failure from overheating
  - Damage to storage/backup media
  - Damage to sensitive equipment components
- Climate controls include:
  - Monitoring temperatures of information systems equipment
  - Keeping all rooms containing information systems equipment at reasonable temperatures (60 to 75°F, or 10 to 25°C)
  - Maintaining humidity levels between 20 and 70 percent
  - Considering turning off unnecessary lights in rooms containing information system equipment
  - Conducting routine preventive maintenance and inspections of climate control system
  - Storing all backup media offsite (with a bonded third party)
  - Developing and exercising a disaster recovery plan
- Water and liquid leakage causes the following risks:
  - Equipment and infrastructure failure from excessive exposure to water or other forms of liquid
  - Damage to storage/backup media and critical hardcopy information
  - Damage to critical equipment components
- Water and liquid leakage controls include:
  - Keeping liquid-proof covers near equipment
  - Installing drains, water detectors, and raised flooring in rooms that house critical information systems equipment
  - Conducting routine inspections of plumbing
  - Using gaseous or dry pipe extinguishing systems near information systems
  - Storing all backup media offsite (with a bonded third party)
  - Developing and exercising a disaster recovery plan
- Electrical interruption causes the following risks:
  - Damage to critical equipment components
  - Damage to software and storage/backup media
  - Loss of climate controls
  - Loss of physical access controls and monitoring devices (i.e., surveillance cameras, door alarms, ID/card readers)
- Electrical interruption controls include:
  - Installing and testing uninterruptible power supplies (UPS) or diesel generators
  - Using surge suppressors
  - Installing electrical line filters to control voltage spikes
  - Using static guards and antistatic carpeting where applicable
  - Ensuring that all equipment is properly grounded
  - Having circuit boxes and wiring routinely inspected
  - Drawing power from two separate grids (if possible)
  - Storing all backup media offsite (with a bonded third party)
  - Developing and exercising a disaster recover plan

Environmental failure, in and of itself, is a threat that can cause considerable damage to information systems. However, it can be also be manifested by natural or man-made threats. Therefore, it is important to approach all threats with a layered approach that has defense-in-depth. This not only ensures that controls cover most of the threats, but that those controls are thorough in their coverage as well.

Facility Protection Strategy

Developing an overall strategy for physical protection is one of the many steps taken toward achieving good information security. One's protection strategy will be comprised of many principles and should center on

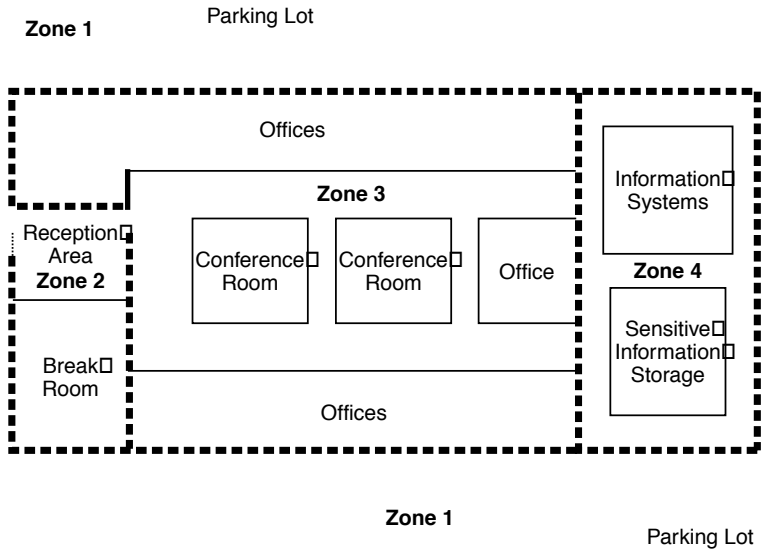


EXHIBIT 158.1 Using zoning for role-based access control.

whether confidentiality, integrity, or availability of the information is of greater importance. Zoning is a strategy that can be used to set a foundation for efficient and effective physical information protection.

Zoning

Zoning is not a new concept. Traditionally, zoning refers to a process used for installing fire detection alarms to identify hidden locations of smoke or fire (above ceiling, under floor, etc.). Additionally, a concept called cross-zoning has been used that allows one to reduce false alarms by requiring two or more alarms to be activated before the fire department is notified.

Zoning is sufficiently flexible to facilitate the simplest to the most detailed security model. Because of this, one can apply all other physical security controls to this concept (e.g., motion detectors, physical intrusion detection alarms, CCTVs, etc.). The biggest advantage is with role-based access control models. In role-based access control schemes, users are assigned access to systems, information, and physical areas according to their role in the organization.

Exhibit 158.1 displays a basic example of the use of zoning for role-based access control. In this example, the zones are labeled 1 through 4, 4 being the most restrictive. In this facility, every employee has access to zones 1, 2, and 3; however, the Information Technology Director, IT staff, and Security Manager, have access to zones 1, 2, 3, and 4 because of their roles.

The natural progression of security is obvious; the zones become more restrictive as one moves further into the facility (from left to right). Once this exercise is completed, the next step would be to determine the controls that should be put in place to support access control zones. Keep in mind that the more restrictive the zone, the stronger and more reliable the controls should be.

By combining physical access controls, role-based models, and zoning, one can build a thorough and centralized system to physically protect one's information and assets. Zoning can be a very important part of one's information security strategy. However, prior to conducting a zoning exercise, one should have already conducted a risk analysis (to understand the threats to and vulnerabilities of one's assets), and developed a risk mitigation strategy. Only then will zoning provide for a solid foundation from which an organization can achieve its information security goals.

Information Systems Physical Security

The second part of physical security is the physical protection of information systems. As discussed, protection should come in layers. If the physical integrity of just one of an organization's computers is com-

promised, information security could be at risk. If someone were to gain unauthorized physical access to a computer, that person could also gain access to all of the information on that computer and possibly any other resource that computer is connected to (including file servers, mainframes, and e-mail).

Information System Classification

Information systems can be classified into three types:

1. *Servers/mainframes:* Usually the most physically secure class of systems. This is due to the common practice of placing them in a location that has some form of access and environmental control. Although this class may be the most physically secure, their overall security is dependent on the physical security of the workstations and portable devices that access them.
2. *Workstations*: Usually located in more open or accessible areas of a facility. Because of their availability within the workplace, workstations can be prone to physical security problems if used carelessly.
3. *Portable devices*: Can be an organization's security nightmare. Although issuing laptops and PDAs to employees facilitates flexibility and productivity in an organization, it poses several serious risks with regard to physical security. With users accessing the company's internal information systems from anywhere, a breach in physical security on one of these devices could undermine an organization's information security. Extreme care must be taken with this class.

Information Systems Physical Threats and Controls

Classifying information systems helps determine which threats pose a greater risk to which systems. This provides a guideline for applying controls. Probably the biggest threat to information systems is that of the user. Keep in mind that if any user fails to practice due diligence in physically protecting their computing assets, nearly all controls will become ineffective, rendering the device vulnerable. This chapter section outlines the basic threats and controls for information systems.

- Loss/theft/destruction poses the following risks:
  - Loss of sensitive information or trade secrets
  - Loss of productivity
  - Loss of revenue
- Loss/theft/destruction controls include:
  - Physical locks for devices
  - Marking and tagging devices
  - Minimize use of location signs
  - Encryption for sensitive information storage
  - Data classification and handing procedures for sensitive information
  - Insurance
  - Awareness training
  - Visible closed circuit television cameras (CCTVs)
  - Guards
  - Alarm systems
  - Routine audits
- Unauthorized access poses the following risks:
  - Loss of sensitive information or trade secrets
  - Information tampering
  - Malware
  - Loss of revenue
- Unauthorized access controls include:
  - Locking consoles
  - Good password practices

— Awareness training
— Data classification and handing procedures for sensitive information
— Minimizing the use of location signs
— Visible closed circuit television cameras (CCTVs)
— Encryption for sensitive information storage
— Strong authentication and access controls

Awareness Training

Although information systems are more prevalent in the world today than ever before (and continue to become ever more so), we nonetheless still live in a physical world. All employees affect physical security, which directly impacts their organization's information security. It is common to find that a majority of physical security failures are due to unaware employees circumventing the controls. Ensuring that all employees receive regular awareness training reduces unintentional security bypasses, while providing an economical way to mitigate risks. No matter how well an information security program is designed and implemented, it only takes one unknowing employee to render it ineffective. Physical security must be among the topics presented in an awareness program, which should also include the following:

- Demonstrate to all employees how even the smallest disregard for physical security can quickly develop into an information security incident or loss of life.
- Educate employees on the security standards and guidelines for the organization. Ensure that employees understand the responsibilities expected of them.
- Distribute monthly publications regarding information security to all employees. Include physical security as a regular topic.
- Provide special orientation for upper management, taking them on tours and offering them a behind-the-scenes look at how information security is done. This rallies support.

Taking the time and effort to provide awareness training will boost the effectiveness of not only one's physical security, but also the entire information security program. By making employees cognizant of the their responsibilities, one can instill a sense of ownership and duty. This transforms the human factor from a disadvantage to an advantage.

Summary

Physical security is more than a niche of information security. In some cases, an organization will have good, strong physical security, but lack many other components of information security. As a practitioner of information security, one must understand the scope and know how to use physical security for protecting assets. Complete physical security will protect all assets, setting a good foundation upon which to build other forms of protection. It is clear that physical security is the foundation for information security.

Bibliography

1. Fennelly, Lawrence J. et al., *Effective Physical Security, Second Edition,* Butterworth-Heinemann, 1997.
2. Fites, P. and Kratz, M.P.J., *Information Systems Security: A Practitioner's Reference*, International Thomson Computer Press, 1996.
3. Tipton, Harold and Krause, Micki, Eds., *Information Security Management Handbook,* 4th edition, Auerbach Publications, 2000.
4. Department of Education, National Center for Education Statistics, *Protecting Your System: Physical Security* (online), 1998. Available from World http://nces.ed.gov/pubs98/safetech/chapter5.html.
5. Tipton, Harold and Krause, Micki, Eds., *Information Security Management Handbook*, Auberbach Publications, 1999.
6. Linux Documentation Project, *Security How-To: Physical Security* (online). Available http://www.linuxdoc.org/HOWTO/Security-HOWTO-3.html.

159

Physical Security: Contolled Access and Layered Defense

Bruce R. Matthews, CISSP

> *Security* (si kyoor'e tē) *n.*, pl. –ties 1. A feeling secure; freedom from fear, doubt, etc. 2. Protection; safeguard.

The above Webster's definition can be restated for the security practitioner as controlled access. In fact, every aspect of an IT security practitioner's job revolves around the process of defining, implementing, and monitoring access to information. This includes physical access. When to use it, how much, and the best way to integrate it with traditional IT security methods, are concepts the IT security professional must be familiar with. The IT security specialist need not be an expert, someone else will fill that role, but effective policies and strategies should take into account the benefits as well as limitations of physical protection. Success depends on close collaboration with the physical security office; they have more than just IT security on their minds and a mutual respect for each other's duties goes a long way. Thus cross training can prove invaluable, particularly when an incident occurs. In essence, a layered, multidisciplined approach can provide a secure feeling; freedom from fear, doubt, etc. Controlled access is security.

Security Is Controlled Access

When one thinks of security, one often thinks of it only in terms of implementation. In IT security, one thinks of passwords and firewalls. In personal security, one thinks of avoiding rape and muggers by staying away from dark alleys and suspicious-looking characters. However, to place physical security in the context of IT security, one must examine what security is — not just how one implements it. In the simplest of terms, it boils down to: security is controlled access. Implementing security, therefore, is the process of controlling access. Passwords and firewalls control access to network and data resources. Avoiding dark alleys and suspicious characters control access to our bodies and possessions. Likewise, security in the home generally refers to locks on the doors and windows. With the locks, one is controlling the access of persons into the protected area. Everyone is denied entry unless they can produce the proper key. By issuing keys to only those persons one desires, one is controlling access. Because one normally does not want anyone entering through the windows after-hours (although a teenager may have a different viewpoint), there is typically no key lock on windows and the level of control is total denial of access. Home alarm systems are gaining increased popularity these days. They also control access by restricting the movements of an intruder who is trying to avoid detection.

The definition — security is controlled access — also holds true for the familiar information security concepts of availability, integrity, and confidentiality. Availability is ensuring access to the data when needed. Integrity implies that the data has been unmodified; thus, access to change the data is limited to only authorized persons or programs.

Confidentiality implies that the information is seen only by those authorized. Thus, confidentiality is controlling access to read the data. All of these concepts are different aspects of controlling access to the data. In a perfect world, one could equate assurance with the degree of control one has over access. However, this is not a perfect world, and it may be more appropriate to equate assurance with the level of confidence one has in the controls. A high level of assurance equates to a high level of confidence that the access controls are working and vice versa. For example, locking the window provides only moderate assurance because one knows that a determined intruder can easily break the window. But a degree of access control is gained because the intruder risks detection from the sound of breaking glass.

Bear in mind, and this is important, that more security is not necessarily less access. That is, controlled access does not equal denied access. The locked window is certainly a control that denies access — totally (with respect to intent, not assurance). On the other hand, Social Security provides security by guaranteeing access to a specified sum of money in old age, or should one say the "golden years." (However, the degree of confidence that this access control will provide the requisite security is left as an exercise for the reader.) It is obvious that practically all controls fall somewhere in between providing complete access and total denial. Thus, it is the level of control over access — not the amount of access — that provides security. Confidence in those controls provides assurance.

This leads to the next topic: a layered defense.

A Layered Defense

A layered defense boosts the confidence level in access controls by providing some redundancy and expanded protection. The details of planning a layered defense for physical security is beyond the scope of this chapter and should be handled by an experienced physical security practitioner. However, the IT security specialist should be able to evaluate the benefits of a layered defense and the security it will and will not provide. When planning a layered defense, the author breaks it into three basic principles: breadth, depth, and deterrence.

Think of applying "breadth" as plugging the holes across a single wall. Each hole represents a different way in or different type of vulnerability. Breadth is used because a single type of control rarely eliminates all vulnerabilities. Relating this first in the familiar IT world, suppose one decides to control read access to data by using a log-on password. But the log-on password does not afford protection if one sends the data over the Internet. A different type of control (i.e., encryption) would therefore provide the additional coverage needed. Physical security works much the same way. For example, suppose one needs to control access to a hot standby site housed in a small one-story warehouse. The facility has a front door, a rear door, a large garage door, and fixed windows that do not open. Locks on the doors control one type of pathway to the inside, but offer no protection for the breakable windows. Thus, bars would be/could be an additional control to provide complete coverage.

The second principle, depth, is commonly ignored yet often the most important aspect for a layered defense. To be realistic with security, one must believe in failure. Any given control is not perfect and will fail, sooner or later. Thus, for depth, one adds layers of additional access controls as a backstop measure. In essence, the single wall becomes several walls, one behind the other. To illustrate on the familiar ground, take a look at the user password. The password will not stay secret forever, often not for a single day, because users have a habit of writing them down or sharing them. Face it; everyone knows that no amount of awareness briefings or admonishments will make the password scheme foolproof. Thus, we embrace the common dictum, "something you have, something you know, and something you are." The password is the "something you know" part; the others provide some depth to the authentication scheme. Depth is achieved by adding additional layers of protection such as a smart card — "something you have." If the password alone is compromised, access control is still in place. But recognize that this too has limitations, so one invokes auditing to verify the controls. Again, physical security works the same way.

For physical security, depth usually works from the outer perimeter, areas far away from the object to be protected, to the center area near the object to be protected. In theory, each layer of access control forms a concentric ring toward the center (although very few facilities are entirely round). The layers are often defined at the perimeter of the grounds, the building entrance and exterior, the building floors, the office suites, the individual office, and the file cabinets or safes.

Deterrence, the third principle, is simply putting enough controls in place that the cost or feasibility of defeating them without getting caught is more than the prize is worth. If the prize to be stolen is a spare $5000

server that could be sold (fenced) in the back alleys for only $1000, it may not be worth it to an employee to try sneaking it out it out a back door with a camera on it when loss of the job and jail time may cost that employee $50,000. Notice here that the deterring factor was the potential cost to the employee, not to the company. A common mistake made even by physical security managers is to equate value only to the owner. Owner value of the protected item is needed for risk analysis to weigh the cost of protection to the cost of recovery/replacement. One does not want to spend $10,000 protecting a $5000 item. However, the principle of deterrence must also consider the value to the perpetrator with respect to their capability — the bad guy's own risk assessment. In this case, maybe an unmonitored $300 camera at the back door instead of a $10,000 monitored system would suffice.

A major challenge is determining how much of the layered defense is breadth and depth in contrast to deterrence. One must examine each layer's contribution to detection, deterrence, or delay, and then factor in a threat's motivation and capabilities. The combined solution is a balancing act called analytical risk management.

Physical Security Technology

Security Components

Locks

Physical security controls are largely comprised of locks (referred to as locking devices by the professionals). In terms of function, there are day access locks, after-hours locks, and emergency egress locks. Day locks permit easy access for authorized persons — such as a keypad or card swipe. After-hours locks are not intended to be opened and closed frequently and are often more substantial. Examples are key locks, locked deadbolts, padlocks, combination padlocks, or high-security combination locks like one would see on safes or vault doors. Emergency egress locks allow easy access in one direction (i.e., away from the fire), but difficult access in the other direction. A common example is the push or "crash" bar style seen at emergency exits in public facilities. Just push the bar to get out, but one needs a key to get back in.

In terms of types, locks can be mechanical or electrical. A mechanical lock requires no electric power. Most of the locks used daily with a key or combination are mechanical. An electric lock requires electricity to move the locking mechanism, usually with a component called a solenoid. A solenoid is a coil of wire around a shaft. The shaft moves in or out when electric current flows through the coil. Another type of electric lock uses a large electromagnet to hold a door closed. The advantage is few moving parts with considerable holding power.

The way people authenticate themselves to a lock (to use an IT term) is becoming more sophisticated each day. Traditionally, people used a key or mechanical combination. Now there are combination locks that generate electricity when one spins the dial to power internal microprocessors and circuits. There are also electronic keypads, computers, biometrics, and card keys to identify people. Although this is more familiar territory to the IT security professional, it all boils down to activating a locking device. Collectively, authentication combined with door locking devices is referred to as a "door control system."

Barriers

Barriers include walls, fences, doors, bollards, and gates. A surprising amount of technology and thought goes into the design of barriers. The physics behind barriers can involve calculations for bomb blasts, fire resistance, and forced entry. Installation concerns such as floor loading, wind resistance, and aesthetics can play a role as well. Making sense of the myriad of options requires the answer to the following question: Who or what is the barrier intended to stop, and for how long?

To supply the answer, think of the barrier as an element of access control. It is not a door to the office, but something to control "whom" or "what" is allowed into the office. Is valuable data stored in the office, such as backup tapes, or is the concern with theft of hardware? Is the supposed thief an employee, or is it a small company where a break-in is more likely? Is the office in a converted wooden house where liability for data lost in fire is the primary concern? If so, how long does one need to keep the fire at bay (i.e., what is the fire department response time)? Know these answers.

Alarms

Barriers and the locks that secure them directly control access. Alarms are primarily for letting us know if that control is functioning properly — that is, has it been breached? Alarms tell us when some sort of action must

be taken, usually by a human. A fire alarm may automatically activate sprinklers as well as the human response by the fire department. In terms of a layered defense, the presence of alarms also adds to the deterrence. Alarms are usually divided into two parts: the controller and the sensors. The sensors detect the alarm condition, such as an intruder's movements or the heat from a fire, and report it to the controller. The controller then initiates the response, such as an alarm bell or dialing the police department. A facility that monitors several control units is referred to as a "central monitoring" facility.

As indicated, sensors usually detect environmental conditions or intrusion. Environmental conditions include temperature, moisture, and vibration. Temperature not only protects against fire, but can alert us to the air conditioner failing in a server room. Moisture may indicate flooding due to rains or broken plumbing. Vibration sensors are used both in environmental sensors, to protect sensitive hardware, and in intrusion detectors such as glass breakage sensors or on fences to detect climbing. Other intrusion sensors detect human motion by measuring changes in heat or ultrasonic sound within a room. In fact, many intrusion sensors are really just environmental sensors configured for human activity. Thus, innocuous items such as coffee pots not turned off or room fans can generate false alarms.

Doors are usually monitored with magnetic switches. A magnet is mounted on the door, and a switch made of thin metal strips is mounted on the doorframe. When the door is shut, the magnet pulls the metal strips closed, completing a circuit (or pushes them open to break a circuit).

The perimeter of an area can be monitored with microwave or infrared beams that are broken when a person passes through them. Cables can be buried in the ground that detect people passing overtop. Animals are a source of false detection for these perimeter sensors.

An important feature of many alarm systems is how the sensors communicate with the controller — wireless or wired. Wireless systems are generally cheaper to install, but can suffer radio frequency interference or intentional jamming. Wired systems can be expensive or impractical to install but can be made quite secure, especially if the wires are in conduit. Whether wired or wireless, the better systems will incorporate some method for the controller to monitor the integrity of the system. The sensors can be equipped with tamper switches and the communication links can be verified through "line monitoring."

The key question for alarms is: who and what is it supposed to detect, and what is the intended response? The "who" will define the sophistication of the alarm system, and the "what" may dictate the sensitivity of the sensors. Provided with this, the alarm specialist can then determine the appropriate mix and placement of sensors.

A major task of the alarm controller is to arm and disarm the system, which really means to act upon or ignore the information from the sensors. With such a vital function, one must have some means to authenticate the person's authority to turn off the alarm system. Like the locks in the previous chapter section, the methods to do this are essentially the same as for authenticating to any information system, ranging from passwords to smart cards to biometrics, with all the same pros and cons.

Lights and Cameras

Lights and cameras are combined because they serve essentially the same function: they allow us to see. In addition, lighting is a critical element for cameras. Poor light or too much light, such as glare, can mean not seeing something as big as a truck. Proper camera lighting is a field unto itself; and for high-security situations, data from lighting and camera manufacturers should be consulted. A common misuse of cameras is assuming that they will detect an intruder. With a camera, the possibility certainly exists; in terms of deterrence, both lights and cameras increase the risk to perpetrators that they will be seen. For many low-threat situations, this is sufficient; however, as threat or risk increases, they cannot be relied upon. If a guard's attention is focused elsewhere (and often is), the event will go unnoticed. If ever in doubt, try putting a camera outside an access door without a buzzer for people to ring. People will become rapidly annoyed that the guard does not notice them and open the door fast enough. Cameras are best suited for assessing a situation — a tool to extend the eyes (and sometimes ears) of the guard force.

Antitheft, Antitamper, and Inventory Controls

It is obvious that the theft of computers and peripherals can directly affect the availability and confidentiality of data. However, tampering is also an issue, particularly with data integrity. Physical access affords the opportunity to bypass many traditional IT security measures by inserting modems, wireless network cards, or additional hard drives to steal password files, boot up on alternate operating systems, and allow unauthorized

network access — the list goes on and on. Physical access to security peripherals such as routers may enable someone to log in locally and modify the settings.

The retail and warehouse industries have created a wide range of products to prevent theft and tampering. Antitamper devices control access to ensure the integrity of the protected asset, whereas antitheft devices and inventory controls are intended to limit movement to a confined area. The technologies behind these products have rapidly spilled over into new product lines designed to protect IT assets.

Antitheft devices include locked cages, cabinets, housings, cables, and anchors. Labels and inventory controls such as barcodes discourage theft. More sophisticated devices include vibration or motion sensors, power line monitoring, and electronic article surveillance (EAS) systems. Power line monitoring alerts us when someone has unplugged the power cord of a computer or other protected asset. EAS systems alert us when a protected asset is moved from a designated area. The most familiar EAS devices are probably those little tags attached to clothes or merchandise in retail stores. They cause that annoying alarm when one departs the store if the clerk forgets to disable it.

Antitamper devices include locked cabinets, locking covers, microswitches, vibration or motion sensors, and antitamper screws.

The Role of Physical Security

A basic role of physical security is to keep unwanted people out, and to keep "insiders" honest. In terms of IT security, the role is not that much different. One could change "people" to "things" to include fire, water, etc., but the idea is the same. The greatest difference is expanding the assets to be protected. Physical security must not only protect people, paper, and property, but it must also protect data in forms other than paper.

So where does one start? Recall the above descriptions of depth in a layered defense where one countermeasure or barrier backstops the preceding one. In a textbook analysis, sufficient depth is determined by security response time. The physical security practitioners view each control or countermeasure as a delaying action. The amount of the time it takes for the guard force to respond is equivalent to the minimum delay needed. Although a tried and true strategy in the physical security realm, it was only recently proposed as an IT security strategy.[1]

For the physical world, it works like this. Suppose one has an estimated response time of ten minutes by the local police. One discounts the perimeter wall as only a deterrent because there are no alarms there. The first alarm is at the front door, which one estimates will take two minutes to get past. Thus, one needs an additional eight minutes worth of inside layers between the door and the cash for the police to apprehend the thief.

For the IT world, layering brings to mind firewalls backed up by routers, backed up by proxies, etc. Notice that physical controls were backed up by additional physical controls and "cyber" controls were backed up by more cyber controls. This is okay to a point; but for data security, the roles of physical and cyber controls should be to complement one another. They become interleaved in a multidisciplinary defense.

A Multidisciplinary Defense

In a multidisciplinary defense, more than one skill set or expertise is brought to bear on the security problem. Physical security is comprised of several disciplines, ranging from barrier technology to antitamper devices. Each discipline aids another. Each component has a purpose to be used in concert with another. The basic relationship between components at each layer is the need to prevent a security event, detect a security event, and assess a security event. For example, there is a locked door with alarm contacts and a camera. The door blocks the way to prevent entry. If the door is opened, the alarm alerts the guard. The guard then uses the camera to assess the situation and decide on an appropriate response. Multiple technologies are integrated to prevent, detect, and assess.

Now take a broader view and consider physical security as a single discipline and IT security as a single discipline. Although separate disciplines, one cannot have one without the other. For example, the payroll office is using Windows NT. The administrator has installed the password filter to ensure that users create quality passwords. Auditing is turned on; file and directory permissions are set. The administrator is aware that the passwords, and hence the network, are still vulnerable because the computer can be booted from removable media (i.e., the floppy drive or CD-ROM). Once booted from a floppy, the password files can be

stolen and cracked. There are always a number of people working late at the company, with a night shift on the factory floor, but payroll employees are generally gone by 4 p.m. (except before payday).

One solution is to disable the floppy and CD-ROM. But this idea is met with a polite yet firm "not if you value your job…" from management. One could modify the boot function from the bios, install a switch and use the tamper alarm option on the motherboard, and replace the computer case screws with a tamper-resistant type. That is one example of a multidisciplinary approach; but considering the number of clients, one does not relish the extra work — particularly when one is constantly servicing the machines. So think more physical security and back up one layer. Put a high-security deadbolt on the payroll office door. Okay, this example seems fairly intuitive, but are we finished? If one has a guard service, then one would want to brief them on the importance of ensuring that the door is closed after normal hours and to make note of a nonpayroll employee who seems to be rebooting or using a payroll computer. How does the guard know who is an authorized payroll (or systems admin) employee? Provide a list. These "extra" physical security details can be easily forgotten.

Now turn the tables. You are chatting with the guards who are quite happy with the new card-access system (the result of a backroom deal with payroll). They have absolute accountability and control over who enters the various sensitive offices. You are happy; your payroll information is secure. Physical security is quite impressive with this set up-and-forget security wonder. There are fewer guards (okay, not all the guards were so happy) and they no longer wander the hallways all night. But then you begin to wonder, where is this card-access system computer located? You learn it is in a closet down the hall and it too is running Windows NT — with a blank password administrator account and no auditing. Hmm, are your payroll files still safe from a computer-savvy, disgruntled employee? From an ex-guard who is now working in janitorial? Perhaps the remaining guards need some IT security assistance.

The Economic Espionage Act of 1996 brings to bear the importance of protecting data, both physically and electronically. The act makes the theft of trade secrets an act of espionage if the benefactor is a foreign government. However, contained in the definition of "trade secret" is the following statement:

> (A) the owner thereof has taken reasonable measures to keep such information secret; unfortunately, there is no firm legal definition of "reasonable measures," but as a starting point, Mr. Patrick W. Kelley, J.D., LL.M., M.B.A, FBI's chief of the Administrative Law Unit, Office of General Counsel, at FBI Headquarters in Washington, D.C., in 1997 provided the following guidance to their field agents: Advise businesses that "owners must take affirmative steps to mark clearly information or materials that they regard as proprietary, protect the physical property in which trade secrets are stored, limit employees' access to trade secrets to only those who truly have a need to know in connection with the performance of their duties, train all employees on the nature and value of the firm's trade secrets, and so on."[2]

This is good advice to protect any valuable information, trade secret or not. In fact, Mr. Kelley's advice is common-sense security practice. One can capture this common sense with the following tenets: identify it, label it, secure it, track it, and know it. These tenets represent the practical side of controlling access. Below are some common physical security implementations, along with their IT security counterparts.

1. Identify it.
   a. *Physical security.* The U.S. government refers to this as classification guidelines. Decide what needs to be protected, and create guidelines on how to recognize it by subject matter or keywords. The guidelines should enable a company novice to determine, based on content, the sensitivity of a document. For example, perhaps any document that describes the project goal or the name of the client is "company confidential" whereas the project name is not sensitive.
   b. *IT security.* The same as physical security, except create an electronic classification guide. Hyperlink it by subject and keyword so a user can easily determine (by answering a series of questions) the material's sensitivity and what is required in terms of the policies.
2. Label it.
   a. *Physical security.* Use a rubber stamp or stickers to identify sensitive documents. Document folders should be distinctive (color or colored band) and labeled. Labels should indicate special handling requirements, dates for downgrading sensitivity, and who has authorized access to it.
   b. *IT security.* Use automatic document headers/footers or cover pages for sensitive data. Automatically print out cover pages.

3. Secure it.
    a. *Physical security.* Create the physical layers of defense based on risk. The following is a list of possibilities for each physical security layer; it does not imply that everyone needs all this stuff. Working from the outer ring inward, these are common options that form layers of physical security that the IT security practitioner should be aware of.
        i. *Perimeter.* Perimeter access controls include physical barriers such as fences, walls, barbed wire, gates, and ID checks. Alarms and cameras are used at the perimeter.
        ii. *Building grounds.* Within the building grounds, cameras, lights, alarms, and roving guards can be deployed, along with physical barriers to control traffic flow (foot or vehicle).
        iii. *Building entrance.* In closer is the facility building where there are doors, locks, barred windows, cameras, alarms, and perhaps another ID check or a card-access system (common in many hotels to gain entry to a room instead of a key).
        iv. *Building floors.* Deeper into the building one might have access limited by floor, with special keys for the elevator as in some hotels and alarmed stairwells. Stairwells and hallways may be monitored with cameras.
        v. *Office suites.* Access controls for the office suite include card-access systems, locks, and keypads that require a code to be entered, human receptionists, and steel or solid-core doors. Wooden doors are typically hollow inside to reduce weight, making them easier to swing and providing less wear-and-tear on the hinges. However, the locks, including deadbolts, do not have much to grab onto and are easily pushed open. Solid cores strengthen the doors considerably. Within the suite may be individual offices with keypads, cards, or regular locks.
        vi. *Office physical security.* Once inside the office, there may be lockable file cabinets, safes, vaults, antitheft/tamper devices, and alarm systems. Lock up any sensitive disks, CD-ROMs, or media. Consider fire/water-resistant storage containers. Use paper shredders.
        vii. *IT security.* Create the IT layers of defense based on risk. Make use of firewalls, proxy servers, routers, network address translation, switches, network monitoring, etc. Use passwords or user authentication, invoke file rights and permissions, anti-virus, data backups, data encryption, or overwrite utilities. Monitors away from observable windows, emergency power source (UPS or generator), spare equipment.
4. Track it.
    a. *Physical security.* Access lists (need-to-know), checkout lists, inventory controls, audits, and registered or insured mail.
    b. *IT security.* Auditing, digital certificates/signatures, file permissions, etc.
5. Know it.
    a. For both physical and IT security, make sure people know what to do and why. Create the policies to implement the protection. Policies should spell out the required access controls and handling procedures. Different jobs have different responsibilities, so vary the presentations and training accordingly.
    b. *Physical security.* Handling procedures should cover issues such as copying, mailing, how long material will be sensitive, and destruction requirements.
    c. *IT security.* Policies for electronic handling, such as copying, e-mailing, posting on Web sites, and deleting files, should be created.

Integrating Physical Security with IT Security Policy

Policies created to fulfill the "know it" tenet provide the necessary roadmaps to implement the other tenets. Policies instruct us to take the steps outlined in the other tenets. With each tenet, there were physical security examples and corresponding IT security examples. Thus, the policies to protect information must address both physical and IT security requirements. Why protect information in digital form, and then not write policy to protect it in paper form? Policy should cover both. They should be consistent in approach, but not always identical in application. For example, suppose there is a policy to ensure that project confidential information is delivered securely to project partners. For the paper world, a sealed envelope might be sufficient; but for the digital world, robust encryption is needed. So why not encrypt the envelope as well? Certainly, the delivery cyclist is capable of tearing open an envelope; so should it not have the same protection? The reason is the

scale of risk. The cyclist can be identified, is probably bonded, and if he or she should drop it, very few people would likely ever see the contents. However, when sending data across the Internet, one has no idea who might come in contact with it, and it can be replicated and redistributed in enormous quantities with amazing speed at virtually no cost to an unethical person. The approach to the "secure it" tenet is the same for digital and nondigital information: deliver it securely; however, the implementations for each are tailored to individual risk.

On the digital side of policy, one cannot divorce oneself from physical access control. For example, a high-level policy states: "Users must be uniquely identified for gain network access." From this emerge standards for passwords, password receipts, and password storage. However, as illustrated previously in the payroll scenario, success for the high-level policy is not assured until one includes standards for protecting physical access to the computer, be it disabling floppy drives or locking the office door. Ensure that IT security policies and standards address avenues of access control in both the physical and digital worlds; this enhances the depth and breadth. Breadth is also improved if standards and policies are applied across the board. If the standards were applied to all networked computing assets in the payroll scenario, the alarm system computer would be covered as well.

Pitfalls of Physical Security

When implementing physical security, be aware of some common limitations and failings.

1. *Social engineering*. As in IT security, social engineering works quite well to bypass physical security controls. Typically, as long as a person appears to belong, no one will question him. If the person provides a plausible story, a guard may concede. Day-access combination locks and electronic card key systems do not suffer guilt when denying access. However, someone can be conned into sharing the combination or opening the door.
2. *Compromise of combinations*. Like passwords, combinations are often written down or posted. They can be also observed by "shoulder surfing."
3. *Tailgating*. A common practice is to "tailgate" into a facility. To tailgate, just wait until an authorized person enters, then walk in behind that person before the door shuts. Often, that person will even hold the door for the tailgater. Following a group is even easier; just feign impatience with them as they take time to get through in front of you. They might let you go first!
4. *Weather/environmental conditions*. Foul weather, bright sunlight, reflections, fog, etc. can render cameras useless or generate false alarms in the sensors. Like a dirty automobile windshield, dust and dirt on a camera lens compound the glare when looking toward the sun. Excessive heat or cold can cause equipment to malfunction. Trees or branches can interfere with perimeter alarms, as can animals and birds.
5. *Appliances*. Appliances that get hot or cold can affect motion detectors and give unwanted alarms. Therefore, take particular care to turn off coffee pots and hotplates after work hours. Moving appliances (fans) or furnishings (window blinds blowing around) generate unwanted alarms, as can a cold wind blowing into a warm room. Interference from electrical noise, like that generated by faulty refrigerator compressors, or acoustical sound such as steam escaping from heating radiators, can cause false indications in sensors.
6. *Complacency*. Either unwanted alarms or false alarms intentionally induced by bad guys creates a loss of faith in the alarm system. For example, whacking a fence equipped with a vibration sensor would generate alarms. After repeated checking and finding no one climbing a fence, the alarms are soon ignored. Long periods of inactivity can also cause complacency or slow response. Occasional drills or competitions may help break the monotony.
7. *Notification of video surveillance*. Similar to notifying users of their lack of privacy when on the company computer system, people should be informed that they are under video surveillance. If the camera and view is not in a public area, it may be a legal requirement. Consult an attorney.
8. *User acceptance*. Users might balk at security measures they feel are too intrusive, difficult, or unsafe — whether their concern is justified or not. If they consider something as ugly, it might be vandalized or management might elect to remove it (or not approve it in the first place). One may have to gain approval from a labor union as well. If they will not accept it, despite efforts at education, one might have to rely on a different security layer or become very creative. At times, it may be a risk deemed acceptable.

IT and Physical Security Teamwork

"Hey! That is the least of my concerns." "Take a number." "Ooh, he is armed and dangerous with a floppy." "<sigh>, Rent-a-Cops. They just do not get it." "<sigh> Computer dweebs. They just do not get it." In fact, none of us ever truly "gets it." If we did, we would be doing the other guy's job. Granted, in small organizations, we probably will be doing the other guy's job; but in larger organizations, with separate physical and IT security personnel, there must be teamwork. Okay, that is a cliché, but teamwork is more than understanding each other's needs and expounding on the virtues of synergy. Teamwork means starting with the understanding that one will never be at the top of another person's priority list. Seek to understand where you *should* fit into each other's priority list. If one works within that framework, then maybe one can achieve some realistic progress.

Well-written policies establish a starting point for teamwork. The policies will identify the specific roles and responsibilities for the physical security team and security officers. A comparison of the physical and IT security requirements articulated in policies may reveal areas of common ground between the two, such as incident response. Whether or not clear policies exist, one can build teamwork on the following triad: education, collaboration, and implementation.

Education

Invite the physical security practitioners, both designers and officers, to attend some computer security courses. Encourage them into the IT world so they can understand where they fit in. A classroom environment is a great place for sharing perceptions and becoming accustomed to the IT practitioner's mindset. Bring them into the courses as mentors, not just as students; they bring a different perspective to the classroom problems. Professional security officers can be quite creative (read "devious") when challenged to think like the opposition; a challenge they frequently engage.

In addition to coursework, educate the physical security crew to in-house IT vulnerabilities that are closely related to their work, such as the susceptibility of outside diskettes to introducing viruses or the potential theft of backup tapes of sensitive data. Do not merely tell them that it is a bad thing and could wipe out the entire corporate profits if taken. Be specific. Show them exactly where the vulnerability exists. If possible, demonstrate it so that they understand the time involved for someone to pull off the crime and what resources they would need. For example, if modems are not permitted in a particular facility, or if breaking into the operating system requires removing the computer case, let them know. Show them what a modem looks like, in comparison to a network interface card. Keep it in their language without being condescending; that is, "You know that little jack on the wall your phone plugs into? Well, a modem card at the rear of a computer will have two of those, one for the telephone and one for the phone line. If it has just one, it is probably the network card, which is okay."

Collaboration

Developing procedures and access controls is enhanced by close collaboration between IT and physical security personnel. If consistency is apparent to users, there will be a greater buy-in on their part. If one labels sensitive documents with a specific color, then labels for diskettes containing the electronic version of those documents should also be the same color. If one requires sensitive documents to be stored in a specific locked file cabinet, perhaps keep the electronic versions in the same or similar locked cabinet.

Collaboration is also helpful for the risk assessments. Applying the principles of a layered defense can become quite complicated and, at times, quite expensive. To design physical protection that is appropriate and creative, a risk management exercise should be completed. In practice, a physical security practitioner may not understand the true value of an item such as a spare server, and the tendency will be to look at the cost of hardware replacement. What if the spare server contained corporate data? What if it was staged for use as a warm standby situation? On the other hand, the IT security practitioner may not recognize creative ways to implement or bypass physical security controls or the extent of insider pilfering. The physical security practitioner generally has a better handle on the costs and practicality of security systems. Maybe a perimeter alarm system sounds great until one finds out too late the additional costs of burying cables under a driveway. Thus, if a company or organization is large enough to have a physical security office or manager, ensure they take part in the process. If hiring a risk assessment company, or providing those services, make sure there is a physical security

expert on staff and that they consult with the client security officers. The security officers may have on-site knowledge of vulnerabilities, emergency service response times, and threats unknown to the hired consultant.

During collaboration, do not forget to address issues such as incident response, particularly with respect to laws and statutes, and contingency plans. Agree on what types of incidents will be pursued aggressively and which will be dealt with at a lower level or as time permits. One does not want one office jumping up and down while the other puts it on the back burner. Identifying competing priorities is also important to identify and iron out at this stage. Maybe the theft of a spare server becomes a low-priority incident to the IT office when it confirms the thief did not intrude on the network and the server had no data. But when the physical security office discovers that the thief broke a fire door, rendering the alarm system inoperable, it becomes a huge life-safety issue. The security office needs to let the IT staff know their priority on pursuing an investigation or prosecution because it may affect issues of evidence where the server was stored. Establish a process for communicating these tactical issues.

Implementation

Whatever is decided during collaboration, make it happen. Test it. See what does not work well; then jump back to the education and collaboration steps to resolve it. Fine-tuning the implementation is a continual process.

Shopping for More Information

A good place to start is with the American Society for Industrial Security (ASIS); it can be found at www.asisonline.org. The ASIS promotes education in security management and offers an ASIS Certified Protection Professional (CPP) Program. At its Web page, one will find an abundance of reference material and publications.

Another organization is the Overseas Security Advisory Council (OSAC). OSAC, established in 1985 by the Department of State, is a joint venture between the U.S. government and the American private sector operating abroad to foster the exchange of security-related information. Administered by the Bureau of Diplomatic Security, the OSAC provides information to organizations to help them protect their investment, facilities, personnel, and intellectual property abroad. Additional information can be found at www.ds-osac.org.

When hiring a physical security consultant, look for the CPP certification combined with experience in the IT sector. A certification that includes expertise in both IT and physical security is the Certified Information Systems Security Professional (CISSP). If a consultant is not professionally certified, look at his or her experience and background. Former law enforcement, military, federal or government investigators, and security engineers are examples of good backgrounds for a consultant. These backgrounds coupled, with professional certification, can make a great package.

The National Center for Education Statistics has some good tips and a checklist for physical security at http://nces.ed.gov/pubs98/safetech/chapter5.html. Although it is intended for schools, which are often strapped for cash and security resources, many of the tips are applicable anywhere.

If one is interested in locks, there is a nice beginner tutorial at http://www.rc3.org/archive/inform/5/4.html. Originally published in a now-defunct hacking zine, *Informatik*, it covers basic lock types and methods of defeating them. It is about ten years old and does not cover high-security locking devices, but it is a quick read and informative.

Infosyssec.org, http://www.infosyssec.org/infosyssec/physfac1.htm, lists a dizzying array of links to physical security companies and information. This should not be the first stop for the physical security novice; but for experienced practitioners, this is a good place to locate a particular vendor or seek specific information.

Conclusion

When challenged to secure data, a wise IT security manager will heed the contributions of physical security. Understand that security is controlled access and that it is best implemented through a layered defense. The layered defense features breadth, depth, and deterrence to ensure that all areas are covered, and that the coverage has fallback contingencies. There is an abundance of technologies to draw upon for each layer. For small or low-equity assets, the choices may be as simple as a lock on the door; but as the value and associated risk

increase, the role of each component becomes more important. Is there a need to detect or assess a situation, or is deterrence the primary objective? If one knows the roles, one can determine how they complement one's IT security strategy and where one's security strategies still fall short or need shoring up. Using the simple tenets — identify it, label it, secure it, track it, and know it — as a template against an existing strategy or to create a new one, will help in assessing how physical and digital security complement each other and help root out those remaining gaps as well. None of the gaps, however, will be adequately filled in practice unless there is detailed collaboration and cooperation between those responsible for physical and digital security. Policies and procedures should establish the relationship, and cross-training should foster it. The benefits and, perhaps more importantly, the limitations of each discipline can be derived from cross-training. Remember: the common goal is to control access. Achieving this, both physically and digitally, gets us much closer to providing a feeling secure; freedom from fear, doubt, etc.

Note

This chapter is dedicated to my father, Floyd V. Matthews, Jr., Professor Emeritus, Cal Poly University, Pomona, California.

Notes

1. Winn Schwartau goes into great detail of detection vs. reaction time for network security in his book, *Time Based Security*, Interpact Press, Florida, 1999.
2. Kelly, Patrick W., J.D., LL.M., MBA, *The Economic Espionage Act of 1996 Law Enforcement Bulletin* (July 1997), FBI Library, Washington, D.C., 1997.

160

Computing Facility Physical Security

Alan Brusewitz, CISSP, CBCP

Most information security practitioners are experienced in and concentrate on logical issues of computer and telecommunications security while leaving physical security to another department. However, most of us would agree that a knowledgeable person with physical access to a console could bypass most of our logical protective measures by simply rebooting the system or accessing the system that is already turned on with root or administrator access in the computer room. Additionally, an unlocked wiring closet could provide hidden access to a network or a means to sabotage existing networks.

Physical access controls and protective measures for computing resources are key ingredients to a well-rounded security program. However, protection of the entire facility is even more important to the well-being of employees and visitors within the workplace. Also, valuable data is often available in hard copy on the desktop, by access to applications, and by using machines that are left unattended. Free access to the entire facility during or after work hours would be a tremendous asset to competitors or people conducting industrial espionage. There is also a great risk from disgruntled employees who might wish to do harm to the company or to their associates.

As demonstrated in the September 11, 2001 attack on the World Trade Center, greater dangers now exist than we may have realized. External dangers seem more probable than previously thought.

Physical access to facilities, lack of control over visitors, and lack of identification measures may place our workplaces and our employees in danger. Additionally, economic slowdowns that cause companies to downsize may create risks from displaced employees who may be upset about their loss of employment.

Physical security is more important than ever to protect valuable information and even more valuable employees. It must be incorporated into the total information security architecture. It must be developed with several factors in mind such as cost of remedies versus value of the assets, perceived threats in the environment, and protective measures that have already been implemented. The physical security plan must be developed and sold to employees as well as management to be successful. It must also be reviewed and audited periodically and updated with improvements developed to support the business of the organization.

Computing Centers

Computing centers have evolved over the years, but they still remain as the area where critical computing assets are enclosed and protected from random or unauthorized access. They have varying degrees of protection and protective measures, depending on the perceptions of management and the assets they contain.

Members of the technical staff often demand computing center access during off-hours, claiming that they might have to reboot systems. Members of management may also demand access because their position in the company requires that they have supervisory control over company assets. Additionally, computer room access is granted to nonemployees such as vendors and customer engineers to service the systems. Keeping track of authorized access and ensuring that it is kept to a minimum is a major task for the information security

department. Sometimes, the task is impossible when the control mechanisms consist of keys or combination locks.

Computing Center Evolution

In the days of large mainframes, computing centers often occupied whole buildings with some space left around for related staff. Those were the days of centralized computing centers where many people were required to perform a number of required tasks. Operators were required to run print operations, mount and dismount tapes, and manage the master console. Production control staffs were required to set up and schedule jobs. In addition, they required staffs of system programmers and, in some cases, system developers. Computer security was difficult to manage, but some controls were imposed with physical walls in place to keep the functions separate. Some of these large systems still remain; however, physical computer room tasks have been reduced through automation and departmental printing.

As distributed systems evolved, servers were installed and managed by system administrators who often performed all system tasks. Many of these systems were built to operate in office environments without the need for stringent environmental controls over heat and humidity. As a result, servers were located in offices where they might not be placed behind a locked door. That security was further eroded with the advent of desktop computing, when data became available throughout the office. In many cases, the servers were implemented and installed in the various departments that wanted control over their equipment and did not want control to go back to the computing staff with their bureaucratic change controls, charge-backs, and perceived slow response to end-user needs.

As the LANs and distributed systems grew in strategic importance, acquired larger user bases, needed software upgrades and interconnectivity, it became difficult for end-user departments to manage and control the systems. Moreover, the audit department realized that there were security requirements that were not fulfilled in support of these critical systems. This resulted in the migration of systems back to centralized control and centralized computer rooms.

Although these systems could withstand environmental fluctuations, the sheer number of servers required some infrastructure planning to keep the heat down and to provide uninterruptible power and network connectivity. In addition, the operating systems and user administration tasks became more burdensome and required an operations staff to support. However, these systems no longer required the multitudes of specialized staffs in the computer rooms to support them. Print operations disappeared for the most part, with data either displayed at the desktop or sent to a local printer for hard copy.

In many cases, computer centers still support large mainframes but they take up a much smaller footprint than the machines of old. Some of those facilities have been converted to support LANs and distributed UNIX-based systems. However, access controls, environmental protections, and backup support infrastructure must still be in place to provide stability, safety, and availability. The security practitioner must play a part ensuring that physical security measures are in place and effective.

As stated before, the computing center is usually part of a facility that supports other business functions. In many cases, that facility supports the entire business. Physical security must be developed to support the entire facility with special considerations for the computing center that is contained within. In fact, protective measures that are applied in and around the entire facility provide additional protection to the computing center.

Environmental Concerns

Most of us do not have the opportunity to determine where our facilities will be located because they probably existed prior to our appointment as an information security staff member. However, that does not prevent us from trying to determine what environmental risks exist and taking action to reduce them. If lucky, you will have some input regarding relocation of the facilities to areas with reduced exposure to threats such as airways, earthquake faults, and floodplains.

Community

The surrounding community may contribute to computer room safety as well as risks. Communities that have strong police and fire services will be able to provide rapid response to threats and incidents. Low crime rates

and strong economic factors provide safety for the computing facilities as well as a favorable climate for attracting top employees.

It is difficult to find the ideal community, and in most cases you will not have the opportunity to select one. Other businesses in that community may provide dangers such as explosive processes, chemical contaminants, and noise pollution. Community airports may have landing and takeoff flight paths that are near the facility. High crime rates could also threaten the computing facility and its inhabitants. Protective measures may have to be enhanced to account for these risks.

The security practitioner can enhance the value of community capabilities by cultivating a relationship with the local police and fire protection organizations. A good relationship with these organizations not only contributes to the safety of the facilities, but also will be key to safety of the staff in the event of an emergency. They should be invited to participate in emergency drills and to critique the process.

The local police should be invited to tour the facilities and understand the layout of the facilities and protective measures in place. In fact, they should be asked to provide suggested improvements to the existing measures that you have employed. If you have a local guard service, it is imperative that they have a working relationship with the local police officials.

The fire department will be more than happy to review fire protection measures and assist in improving them. In many cases, they will insist with inspecting such things as fire extinguishers and other fire suppression systems. It is most important that the fire department understand the facility layout and points of ingress and egress. They must also know about the fire suppression systems in use and the location of controls for those systems.

Acts of Nature

In most cases we cannot control the moods of Mother Nature or the results of her wrath. However, we can prepare for the most likely events and try to reduce their effects. Earthquake threats may require additional bracing and tie-down straps to prevent servers and peripheral devices from destruction due to tipping or falling. Flooding risks can be mitigated with the installation of sump pumps and locating equipment above the ground floor. Power outages resulting from tornadoes and thunderstorms may be addressed with uninterruptible power supply (UPS) systems and proper grounding of facilities.

The key point with natural disasters is that they cannot be eliminated in most cases. Remedies must be designed based on the likelihood that an event will occur and with provisions for proper response to it. In all cases, data backup with off-site storage or redundant systems are required to prepare for manmade or natural disasters.

Other External Risks

Until the events that occurred on September 11, 2001, physical security concerns related to riots, workplace violence, and local disruptions. The idea of terrorist acts within the country seemed remote but possible. Since that date, terrorism is not only possible, but also probable. Measures to protect facilities by use of cement barriers, no-parking zones, and guarded access gates have become understandable to both management and staff. The cost and inconvenience that these measures impose are suddenly more acceptable.

Many of our facilities are located in areas that are considered out of the target range that terrorists might attack. However, the Oklahoma City bombing occurred in a low-target area. The anthrax problems caused many unlikely facilities to be vacated. The risks of bioterrorism or attacks on nuclear power plants are now considered real and possible, and could occur in almost any city. Alternate site planning must be considered in business continuity and physical security plans.

Facility

The facilities that support our computing environments are critical to the organization in providing core business services and functions. There are few organizations today that do not rely on computing and telecommunications resources to operate their businesses and maintain services to their customers. This requires security over both the physical and logical aspects of the facility. The following discussion concentrates on the physical protective measures that should be considered for use in the computing center and the facilities that surround it.

Layers of Protection

For many computing facilities, the front door is the initial protection layer that is provided to control access and entry to the facility. This entry point will likely be one of many others such as back doors, loading docks, and other building access points. A guard or a receptionist usually controls front-door access. Beyond that, other security measures apply based on the value of contents within. However, physical security of facilities may begin outside the building.

External Protective Measures

Large organizations may have protective fences surrounding the entire campus with access controlled by a guard-activated or card-activated gate. The majority of organizations will not have perimeter fences around the campus but may have fences around portions of the building. In most of those cases, the front of the building is not fenced due to the need for entry by customers, visitors, and staff. These external protective measures may be augmented through the use of roving guards and closed-circuit television (CCTV) systems that provide a 360-degree view of the surrounding area.

Security practitioners must be aware of the risks and implement cost-effective measures that provide proper external protection. Measures to consider are:

- Campus perimeter fences with controlled access gates
- Building perimeter fences with controlled access gates
- Building perimeter fences controlling rear and side access to the building
- Cement barriers in the front of the building
- Restrict parking to areas away from the building
- CCTV viewing of building perimeters

External Walls

Facilities must be constructed to prevent penetration by accidental or unlawful means. Windows provide people comforts for office areas and natural light, but they can be a means for unauthorized entry. Ground floors may be equipped with windows; however, they could be eliminated if that floor were reserved for storage and equipment areas. Loading docks may provide a means of unauthorized entry and, if possible, should be located in unattached buildings or be equipped with secured doors to control entry. Doors that are not used for normal business purposes should be locked and alarmed with signs that prohibit their use except for emergencies.

Internal Structural Concerns

Critical rooms such as server and telecommunications areas should be constructed for fire prevention and access controls. Exterior walls for these rooms should not contain windows or other unnecessary entry points. They should also be extended above false ceilings and below raised floors to prevent unlawful entry and provide proper fire protection. Additional entry points may be required for emergency escape or equipment movement. These entrances should be locked when not in use and should be equipped with alarms to prevent unauthorized entry.

Ancillary Structures (Wiring Cabinets and Closets)

Wiring cabinets may be a source of unauthorized connectivity to computer networks and must be locked at all times unless needed by authorized personnel. Janitor closets should be reserved for that specific purpose and should not contain critical network or computing connections. They must be inspected on a regular basis to ensure that they do not contain flammable or other hazardous materials.

Facility Perils and Computer Room Locations

Computer rooms are subject to hazards that are created within the general facility. These hazards can be reduced through good facility design and consideration for critical equipment.

Floor Locations

Historically, computing equipment was added to facilities that were already in use for general business processes. Often, the only open area left for computing equipment was the basement. In many cases, buildings were not built to support heavy computers and disk storage devices on upper floors, so the computer room was

constructed on the ground floor. In fact, organizations were so proud of the flashy computer equipment that they installed observation windows for public viewing, with large signs to assist them in getting there.

Prudent practices along with a realization that computing resources were critical to the continued operation of the company have caused computing facilities to be relocated to more protected areas with minimum notification of their special status. Computer rooms have been moved to upper floors to mitigate flooding and access risks. Freight elevators have been installed to facilitate installation and removal of computing equipment and supplies. Windows have been eliminated and controlled doors have been added to ensure only authorized access.

Rest Rooms and Other Water Risks

Water hazards that are located above computer rooms could cause damage to critical computing equipment if flooding and leakage occurs. A malfunctioning toilet or sink that overflows in the middle of the night could be disastrous to computer operations. Water pipes that are installed in the flooring above the computer room could burst or begin to leak in the event of earthquakes or corrosion. A well-sealed floor will help, but the best prevention is to keep those areas clear of water hazards.

Adjacent Office Risks

Almost all computing facilities have office areas to support the technical staff or, in many cases, the rest of the business. These areas can provide risks to the computing facility from fire, unauthorized access, or chemical spills. Adjacent office areas should be equipped with appropriate fire suppression systems that are designed to control flammable material and chemical fires. Loading docks and janitor rooms can also be a source of risk from fire and chemical hazards. Motor-generated UPS systems should be located in a separate building due to their inherent risks of fire and carbon monoxide. The local fire department can provide assistance to reduce risks that may be contained in other offices as well as the computing center.

Protective Measures

Entrances to computing facilities must be controlled to protect critical computing resources, but they must also be controlled to protect employees and sensitive business information. As stated before, valuable information is often left on desks and in unlocked cabinets throughout the facility. Desktop computers are often left on overnight with valuable information stored locally. In some cases, these systems are left logged on to sensitive systems. Laptops with sensitive data can be stolen at night and even during business hours.

To protect valuable information resources, people, and systems, various methods and tools should be considered. Use of any of these tools must be justified according to the facility layout and the value of the resources contained within.

Guard Services

There are many considerations related to the use of guard services. The major consideration, other than whether to use them, is employee versus purchased services. The use of employee guards may be favored by organizations with the idea that employees are more loyal to the organization and will be trustworthy. However, there are training, company benefit, and insurance considerations that accompany that decision. Additionally, the location may not have an alternative guard source available. If the guards are to be armed, stringent controls and training must be considered.

There are high-quality guard services available in most areas that will furnish trained and bonded guards who are supervised by experienced managers. Although cost is a factor in the selection of a contract guard service, it should not be the major one. The selection process should include a request for proposal (RFP) that requires references and stringent performance criteria. Part of the final selection process must include discussions with customer references and a visit to at least two customer sites. Obviously, the guard service company should be properly licensed and provide standard business documentation.

The guard service will be operating existing and planned security systems that may include CCTV, card access systems, central control rooms, and fire suppression systems. Before contracting with an organization, that organization must demonstrate capabilities to operate existing and planned systems. It should also be able to provide documented operating procedures that can be modified to support the facility needs.

Intrusion Monitoring Systems

Closed-circuit television (CCTV) systems have been used for years to protect critical facilities. These systems have improved considerably over the years to provide digital images that take up less storage space and be transmitted over TCP/IP-based networks. Their images can be combined with other alarm events to provide a total picture for guard response as well as event history. Digital systems that are activated in conjunction with motion detection or other alarms may be more effective because their activation signals a change to the guard who is assigned to watch them.

CCTV systems allow guards to keep watch on areas that are located remotely, are normally unmanned, or require higher surveillance, such as critical access points. These systems can reduce the need for additional manpower to provide control over critical areas. In many cases, their mere presence serves as a deterrent to unwanted behavior.

They may also contribute to employee safety by providing surveillance over parking areas, low traffic areas, and high-value functions such as cashier offices. A single guard in a central control center can spot problems and dispatch roving manpower to quickly resolve threats. In addition to the above, stored images may be used to assist law enforcement in apprehending violators and as evidence in a court of law.

Security requirements will vary with different organizations; however, CCTV may be useful in the following areas:

- Parking lots for employee and property safety
- Emergency doors where access is restricted
- Office areas during nonworking hours
- Server and telecommunications equipment rooms during nonworking hours
- Loading docks and delivery gates
- Cashier and check-processing areas
- Remote facilities where roving guards would be too costly
- Executive office areas in support of executive protection programs
- Mantrap gates to ensure all entry cards have been entered

Alarms and motion-detection systems are designed to signal the organization that an unusual or prohibited event has occurred. Doors that should not be used during normal business activity may be equipped with local sound alarms or with electronic sensors that signal a guard or activate surveillance systems. Motion detectors are often installed in areas that are normally unmanned. In some systems, motion detection is activated during nonbusiness hours and can be disabled or changed to allow for activities that are properly scheduled in those areas.

Many systems can be IP addressable over the backbone TCP/IP network, and alarm signals can be transmitted from multiple remote areas. It is important to note that IP-based systems may be subject to attack. The vendor of these systems must ensure that these systems are hack-protected against covert activities by unauthorized people.

Physical Access Control Measures

Physical access controls are as important as logical access controls to protect critical information resources. Multiple methods are available, including manual and automated systems. Often, cost is the deciding factor in their selection despite the risks inherent in those tools.

Access Policies

All good security begins with policies. Policies are the drivers of written procedures that must be in place to provide consistent best practices in the protection of people and information resources. Policies are the method by which management communicates its wishes. Policies are also used to set standards and assign responsibility for their enforcement. Once policies are developed, they should be published for easy access and be part of the employee awareness training program.

Policies define the process of granting and removing access based on need-to-know. If badges are employed, policies define how they are to be designed, worn, and used. Policies define who is allowed into restricted areas or how visitors are to be processed. There is no magic to developing policies, but they are required as a basic tool to protect information resources.

Keys and Cipher Locks

Keys and cipher (keypad) locks are the simplest to use and hardest to control in providing access to critical areas. They do not provide a means of identifying who is accessing a given area, nor do they provide an audit trail. Keys provide a slightly better security control than keypad locks in that the physical device must be provided to allow use. While they can be copied, that requires extra effort to accomplish. If keys are used to control access, they should be inventoried and stamped with the words *Do Not Duplicate*.

Cipher locks require that a person know the cipher code to enter an area. Once given out, use of this code cannot be controlled and may be passed throughout an organization by word of mouth. There is no audit trail for entry, nor is there authentication that it is used by an authorized user. Control methods consist of periodic code changes and shielding to prevent other people from viewing the authorized user's code entry. Use of these methods of entry control could be better protected through the use of CCTV.

Card Access Controls

Card access controls are considerably better tools than keys and cipher locks if they are used for identification and contain a picture of the bearer. Without pictures, they are only slightly better than keys because they are more difficult to duplicate. If given to another person to gain entry, the card must be returned for use by the authorized cardholder. Different types of card readers can be employed to provide ease of use (proximity readers) and different card identification technology. Adding biometrics to the process would provide added control along with increased cost and inconvenience that might be justified to protect the contents within.

The most effective card systems use a central control computer that can be programmed to provide different access levels depending on need, time zone controls that limit access to certain hours of the day, and an audit trail of when the card was used and where it was entered. Some systems even provide positive in and out controls that require a card to be used for both entry and exit. If a corresponding entry/exit transaction is not in the system, future entry will be denied until management investigation actions are taken.

Smart card technology is being developed to provide added security and functionality. Smart cards can have multiple uses that expand beyond mere physical access. Additional uses for this type of card include computer access authentication, encryption using digital certificates, and debit cards for employee purchases in the cafeteria or employee store. There is some controversy about multiple-use cards because a single device can be used to gain access to many different resources. If the employee smart card provides multiple access functions as well as purchasing functions, the cardholder will be less likely to loan the badge to an unauthorized person for use and will be more likely to report its loss.

Mantraps and Turnstiles

Additional controls can be provided through the use of mantraps and turnstiles. These devices prevent unauthorized tailgating and can be used to require inspection of parcels when combined with guard stations. These devices also force the use of a badge to enter through a control point and overcome the tendency for guards to allow entry because the person looks familiar to them. Mantraps and turnstiles can control this weakness if the badge is confiscated upon termination of access privileges. The use of positive entry/exit controls can be added to prevent card users from passing their card back through the control point to let a friend enter.

Fire Controls

Different fire control mechanisms must be employed to match the risks that are present in protected areas. Fire control systems may be as simple as a hand-held fire extinguisher or be combined with various detection mechanisms to provide automated activation. Expert advice should be used to match the proper system to the existing threats. In some cases, multiple systems may be used to ensure that fires do not reignite and cause serious damage.

Detectors and Alarms

Smoke and water detectors can provide early warning and alarm the guards that something dangerous may be happening. Alarms may also trigger fire prevention systems to activate. To be effective, they must be carefully placed and tested by experts in fire prevention.

Water-Based Systems

Water-based systems control fires by reducing temperatures below the combustion point. They are usually activated through overhead sprinklers to extinguish fires before they can spread. The problem with water-based systems is that they cause a certain amount of damage to the contents of areas they are designed to protect. In addition, they may cause flooding in adjacent areas if they are not detected and shut off quickly following an event.

Water-based systems may be either dry pipe or wet pipe systems. Wet pipe systems are always ready to go and are activated when heat or accidental means open the sprinkler heads. There is no delay or shut-off mechanism that can be activated prior to the start of water flow. Water in the pipes that connect to the sprinkler heads may become corroded, causing failure of the sprinkler heads to activate in an emergency.

Dry pipe systems are designed to allow some preventive action before they activate. These types of systems employ a valve to prevent the flow of water into the overhead pipes until a fire alarm event triggers water release. Dry pipe systems will not activate and cause damage if a sprinkler head is accidentally broken off. They also allow human intervention to override water flow if the system is accidentally activated.

Gas-Based Fire Extinguishing Systems

Halon-type systems are different from water-based systems in that they control fires by interrupting the chemical reactions needed to continue combustion. They replaced older gas systems such as carbon dioxide that controlled fires by replacing the oxygen with a gas (CO_2) that did not support the combustion process. Oxygen replacement systems were effective, but they were toxic to humans who might be in the CO_2-activated room due to the need for oxygen to survive.

Throughout the 1970s and 1980s, halon systems were the preferred method to protect computer and telecommunication rooms from fire damage because they extinguished the fire without damaging sensitive electronic equipment. Those systems could extinguish fires and yet allow humans to breathe and survive in the flooded room. The problem with halon is that it proved unfriendly to the ozone layer and was banned from new implementations by an international agreement (Montreal Protocol). There are numerous Clean Air Act and EPA regulations now in effect to govern the use of existing halon systems and supplies. Current regulations and information can be obtained by logging onto www.epa.gov/docs/ozone/ title6/snap/hal.html. This site also lists manufacturers of halon substitute systems.

Today, halon replacement systems are available that continue to extinguish fires, do not harm the ozone layer, and, most important, do not harm humans who may be in the gas-flooded room. Although these systems will not kill human inhabitants, most system manufacturers warn that people should leave the gas-flooded area within one minute of system activation. Current regulations do not dictate the removal of halon systems that are in place; however, any new or replacement halon systems must employ the newer ozone-friendly gas (e.g., FM 200).

Utility and Telecommunications Backup Requirements

Emergency Lighting

As stated before, modern computer rooms are usually lacking in windows or other sources of natural light. Therefore, when a power outage occurs, these rooms become very dark and exits become difficult to find. Even in normal offices, power outages may occur in areas that are staffed at night. In all of these cases, emergency lighting with exit signs must be installed to allow people to evacuate in an orderly and safe manner. Emergency lighting is usually provided by battery-equipped lamps that are constantly charged until activated.

UPS Systems

Uninterruptible power supply (UPS) systems ensure that a computing system can continue to run, or at least shut down in an orderly manner, if normal power is lost. Lower cost systems rely on battery backup to provide an orderly shutdown; motor generator backup systems used in conjunction with battery backup can provide continuous power as long as the engines receive fuel (usually diesel). As usual, cost is the driver for choosing the proper UPS system. More enlightened management will insist on a business impact analysis prior to making that decision to ensure that critical business needs are met.

Regardless of the type of system employed, periodic testing is required to ensure that the system will work when needed. Diesel systems should be tested weekly to ensure they work and to keep the engines properly lubricated.

Redundant Connections

Redundancy should be considered for facility electrical power, air conditioning, telecommunications connections, and water supplies. Certain systems such as UPS can be employed to mitigate the need for electrical redundancy. Telecommunications connectivity should be ensured with redundant connections. In this E-commerce world, telecommunications redundancy should also include connections to the Internet. Water is important to the staff, but environmental systems (cooling towers) may also depend on a reliable supply. In most cases, this redundancy can be provided with separate connections to the water main that is provided by the supporting community.

Summary

Physical security must be considered to provide a safe working environment for the people who visit and work in a facility. Although physical access controls must be employed for safety reasons, they also should prevent unauthorized access to critical computing resources.

Many tools are available to provide physical security that continues to be enhanced with current technology. Backbone networks and central control computers can support the protection of geographically separated facilities and operations. IP-supported systems can support the collection of large amounts of data from various sensors and control mechanisms and provide enhanced physical security while keeping manpower at a minimum.

The information security practitioner must become aware of existing physical security issues and be involved. If a separate department provides physical security, coordination with them becomes important to a total security approach. If information security organizations are assigned to provide physical security, they must become aware of the tools that are available and determine where to employ them.

160

Computing Facility Physical Security

Alan Brusewitz, CISSP, CBCP

Most information security practitioners are experienced in and concentrate on logical issues of computer and telecommunications security while leaving physical security to another department. However, most of us would agree that a knowledgeable person with physical access to a console could bypass most of our logical protective measures by simply rebooting the system or accessing the system that is already turned on with root or administrator access in the computer room. Additionally, an unlocked wiring closet could provide hidden access to a network or a means to sabotage existing networks.

Physical access controls and protective measures for computing resources are key ingredients to a well-rounded security program. However, protection of the entire facility is even more important to the well-being of employees and visitors within the workplace. Also, valuable data is often available in hard copy on the desktop, by access to applications, and by using machines that are left unattended. Free access to the entire facility during or after work hours would be a tremendous asset to competitors or people conducting industrial espionage. There is also a great risk from disgruntled employees who might wish to do harm to the company or to their associates.

As demonstrated in the September 11, 2001 attack on the World Trade Center, greater dangers now exist than we may have realized. External dangers seem more probable than previously thought.

Physical access to facilities, lack of control over visitors, and lack of identification measures may place our workplaces and our employees in danger. Additionally, economic slowdowns that cause companies to downsize may create risks from displaced employees who may be upset about their loss of employment.

Physical security is more important than ever to protect valuable information and even more valuable employees. It must be incorporated into the total information security architecture. It must be developed with several factors in mind such as cost of remedies versus value of the assets, perceived threats in the environment, and protective measures that have already been implemented. The physical security plan must be developed and sold to employees as well as management to be successful. It must also be reviewed and audited periodically and updated with improvements developed to support the business of the organization.

Computing Centers

Computing centers have evolved over the years, but they still remain as the area where critical computing assets are enclosed and protected from random or unauthorized access. They have varying degrees of protection and protective measures, depending on the perceptions of management and the assets they contain.

Members of the technical staff often demand computing center access during off-hours, claiming that they might have to reboot systems. Members of management may also demand access because their position in the company requires that they have supervisory control over company assets. Additionally, computer room access is granted to nonemployees such as vendors and customer engineers to service the systems. Keeping track of authorized access and ensuring that it is kept to a minimum is a major task for the information security

department. Sometimes, the task is impossible when the control mechanisms consist of keys or combination locks.

Computing Center Evolution

In the days of large mainframes, computing centers often occupied whole buildings with some space left around for related staff. Those were the days of centralized computing centers where many people were required to perform a number of required tasks. Operators were required to run print operations, mount and dismount tapes, and manage the master console. Production control staffs were required to set up and schedule jobs. In addition, they required staffs of system programmers and, in some cases, system developers. Computer security was difficult to manage, but some controls were imposed with physical walls in place to keep the functions separate. Some of these large systems still remain; however, physical computer room tasks have been reduced through automation and departmental printing.

As distributed systems evolved, servers were installed and managed by system administrators who often performed all system tasks. Many of these systems were built to operate in office environments without the need for stringent environmental controls over heat and humidity. As a result, servers were located in offices where they might not be placed behind a locked door. That security was further eroded with the advent of desktop computing, when data became available throughout the office. In many cases, the servers were implemented and installed in the various departments that wanted control over their equipment and did not want control to go back to the computing staff with their bureaucratic change controls, charge-backs, and perceived slow response to end-user needs.

As the LANs and distributed systems grew in strategic importance, acquired larger user bases, needed software upgrades and interconnectivity, it became difficult for end-user departments to manage and control the systems. Moreover, the audit department realized that there were security requirements that were not fulfilled in support of these critical systems. This resulted in the migration of systems back to centralized control and centralized computer rooms.

Although these systems could withstand environmental fluctuations, the sheer number of servers required some infrastructure planning to keep the heat down and to provide uninterruptible power and network connectivity. In addition, the operating systems and user administration tasks became more burdensome and required an operations staff to support. However, these systems no longer required the multitudes of specialized staffs in the computer rooms to support them. Print operations disappeared for the most part, with data either displayed at the desktop or sent to a local printer for hard copy.

In many cases, computer centers still support large mainframes but they take up a much smaller footprint than the machines of old. Some of those facilities have been converted to support LANs and distributed UNIX-based systems. However, access controls, environmental protections, and backup support infrastructure must still be in place to provide stability, safety, and availability. The security practitioner must play a part ensuring that physical security measures are in place and effective.

As stated before, the computing center is usually part of a facility that supports other business functions. In many cases, that facility supports the entire business. Physical security must be developed to support the entire facility with special considerations for the computing center that is contained within. In fact, protective measures that are applied in and around the entire facility provide additional protection to the computing center.

Environmental Concerns

Most of us do not have the opportunity to determine where our facilities will be located because they probably existed prior to our appointment as an information security staff member. However, that does not prevent us from trying to determine what environmental risks exist and taking action to reduce them. If lucky, you will have some input regarding relocation of the facilities to areas with reduced exposure to threats such as airways, earthquake faults, and floodplains.

Community

The surrounding community may contribute to computer room safety as well as risks. Communities that have strong police and fire services will be able to provide rapid response to threats and incidents. Low crime rates

and strong economic factors provide safety for the computing facilities as well as a favorable climate for attracting top employees.

It is difficult to find the ideal community, and in most cases you will not have the opportunity to select one. Other businesses in that community may provide dangers such as explosive processes, chemical contaminants, and noise pollution. Community airports may have landing and takeoff flight paths that are near the facility. High crime rates could also threaten the computing facility and its inhabitants. Protective measures may have to be enhanced to account for these risks.

The security practitioner can enhance the value of community capabilities by cultivating a relationship with the local police and fire protection organizations. A good relationship with these organizations not only contributes to the safety of the facilities, but also will be key to safety of the staff in the event of an emergency. They should be invited to participate in emergency drills and to critique the process.

The local police should be invited to tour the facilities and understand the layout of the facilities and protective measures in place. In fact, they should be asked to provide suggested improvements to the existing measures that you have employed. If you have a local guard service, it is imperative that they have a working relationship with the local police officials.

The fire department will be more than happy to review fire protection measures and assist in improving them. In many cases, they will insist with inspecting such things as fire extinguishers and other fire suppression systems. It is most important that the fire department understand the facility layout and points of ingress and egress. They must also know about the fire suppression systems in use and the location of controls for those systems.

Acts of Nature

In most cases we cannot control the moods of Mother Nature or the results of her wrath. However, we can prepare for the most likely events and try to reduce their effects. Earthquake threats may require additional bracing and tie-down straps to prevent servers and peripheral devices from destruction due to tipping or falling. Flooding risks can be mitigated with the installation of sump pumps and locating equipment above the ground floor. Power outages resulting from tornadoes and thunderstorms may be addressed with uninterruptible power supply (UPS) systems and proper grounding of facilities.

The key point with natural disasters is that they cannot be eliminated in most cases. Remedies must be designed based on the likelihood that an event will occur and with provisions for proper response to it. In all cases, data backup with off-site storage or redundant systems are required to prepare for manmade or natural disasters.

Other External Risks

Until the events that occurred on September 11, 2001, physical security concerns related to riots, workplace violence, and local disruptions. The idea of terrorist acts within the country seemed remote but possible. Since that date, terrorism is not only possible, but also probable. Measures to protect facilities by use of cement barriers, no-parking zones, and guarded access gates have become understandable to both management and staff. The cost and inconvenience that these measures impose are suddenly more acceptable.

Many of our facilities are located in areas that are considered out of the target range that terrorists might attack. However, the Oklahoma City bombing occurred in a low-target area. The anthrax problems caused many unlikely facilities to be vacated. The risks of bioterrorism or attacks on nuclear power plants are now considered real and possible, and could occur in almost any city. Alternate site planning must be considered in business continuity and physical security plans.

Facility

The facilities that support our computing environments are critical to the organization in providing core business services and functions. There are few organizations today that do not rely on computing and telecommunications resources to operate their businesses and maintain services to their customers. This requires security over both the physical and logical aspects of the facility. The following discussion concentrates on the physical protective measures that should be considered for use in the computing center and the facilities that surround it.

Layers of Protection

For many computing facilities, the front door is the initial protection layer that is provided to control access and entry to the facility. This entry point will likely be one of many others such as back doors, loading docks, and other building access points. A guard or a receptionist usually controls front-door access. Beyond that, other security measures apply based on the value of contents within. However, physical security of facilities may begin outside the building.

External Protective Measures

Large organizations may have protective fences surrounding the entire campus with access controlled by a guard-activated or card-activated gate. The majority of organizations will not have perimeter fences around the campus but may have fences around portions of the building. In most of those cases, the front of the building is not fenced due to the need for entry by customers, visitors, and staff. These external protective measures may be augmented through the use of roving guards and closed-circuit television (CCTV) systems that provide a 360-degree view of the surrounding area.

Security practitioners must be aware of the risks and implement cost-effective measures that provide proper external protection. Measures to consider are:

- Campus perimeter fences with controlled access gates
- Building perimeter fences with controlled access gates
- Building perimeter fences controlling rear and side access to the building
- Cement barriers in the front of the building
- Restrict parking to areas away from the building
- CCTV viewing of building perimeters

External Walls

Facilities must be constructed to prevent penetration by accidental or unlawful means. Windows provide people comforts for office areas and natural light, but they can be a means for unauthorized entry. Ground floors may be equipped with windows; however, they could be eliminated if that floor were reserved for storage and equipment areas. Loading docks may provide a means of unauthorized entry and, if possible, should be located in unattached buildings or be equipped with secured doors to control entry. Doors that are not used for normal business purposes should be locked and alarmed with signs that prohibit their use except for emergencies.

Internal Structural Concerns

Critical rooms such as server and telecommunications areas should be constructed for fire prevention and access controls. Exterior walls for these rooms should not contain windows or other unnecessary entry points. They should also be extended above false ceilings and below raised floors to prevent unlawful entry and provide proper fire protection. Additional entry points may be required for emergency escape or equipment movement. These entrances should be locked when not in use and should be equipped with alarms to prevent unauthorized entry.

Ancillary Structures (Wiring Cabinets and Closets)

Wiring cabinets may be a source of unauthorized connectivity to computer networks and must be locked at all times unless needed by authorized personnel. Janitor closets should be reserved for that specific purpose and should not contain critical network or computing connections. They must be inspected on a regular basis to ensure that they do not contain flammable or other hazardous materials.

Facility Perils and Computer Room Locations

Computer rooms are subject to hazards that are created within the general facility. These hazards can be reduced through good facility design and consideration for critical equipment.

Floor Locations

Historically, computing equipment was added to facilities that were already in use for general business processes. Often, the only open area left for computing equipment was the basement. In many cases, buildings were not built to support heavy computers and disk storage devices on upper floors, so the computer room was

constructed on the ground floor. In fact, organizations were so proud of the flashy computer equipment that they installed observation windows for public viewing, with large signs to assist them in getting there.

Prudent practices along with a realization that computing resources were critical to the continued operation of the company have caused computing facilities to be relocated to more protected areas with minimum notification of their special status. Computer rooms have been moved to upper floors to mitigate flooding and access risks. Freight elevators have been installed to facilitate installation and removal of computing equipment and supplies. Windows have been eliminated and controlled doors have been added to ensure only authorized access.

Rest Rooms and Other Water Risks

Water hazards that are located above computer rooms could cause damage to critical computing equipment if flooding and leakage occurs. A malfunctioning toilet or sink that overflows in the middle of the night could be disastrous to computer operations. Water pipes that are installed in the flooring above the computer room could burst or begin to leak in the event of earthquakes or corrosion. A well-sealed floor will help, but the best prevention is to keep those areas clear of water hazards.

Adjacent Office Risks

Almost all computing facilities have office areas to support the technical staff or, in many cases, the rest of the business. These areas can provide risks to the computing facility from fire, unauthorized access, or chemical spills. Adjacent office areas should be equipped with appropriate fire suppression systems that are designed to control flammable material and chemical fires. Loading docks and janitor rooms can also be a source of risk from fire and chemical hazards. Motor-generated UPS systems should be located in a separate building due to their inherent risks of fire and carbon monoxide. The local fire department can provide assistance to reduce risks that may be contained in other offices as well as the computing center.

Protective Measures

Entrances to computing facilities must be controlled to protect critical computing resources, but they must also be controlled to protect employees and sensitive business information. As stated before, valuable information is often left on desks and in unlocked cabinets throughout the facility. Desktop computers are often left on overnight with valuable information stored locally. In some cases, these systems are left logged on to sensitive systems. Laptops with sensitive data can be stolen at night and even during business hours.

To protect valuable information resources, people, and systems, various methods and tools should be considered. Use of any of these tools must be justified according to the facility layout and the value of the resources contained within.

Guard Services

There are many considerations related to the use of guard services. The major consideration, other than whether to use them, is employee versus purchased services. The use of employee guards may be favored by organizations with the idea that employees are more loyal to the organization and will be trustworthy. However, there are training, company benefit, and insurance considerations that accompany that decision. Additionally, the location may not have an alternative guard source available. If the guards are to be armed, stringent controls and training must be considered.

There are high-quality guard services available in most areas that will furnish trained and bonded guards who are supervised by experienced managers. Although cost is a factor in the selection of a contract guard service, it should not be the major one. The selection process should include a request for proposal (RFP) that requires references and stringent performance criteria. Part of the final selection process must include discussions with customer references and a visit to at least two customer sites. Obviously, the guard service company should be properly licensed and provide standard business documentation.

The guard service will be operating existing and planned security systems that may include CCTV, card access systems, central control rooms, and fire suppression systems. Before contracting with an organization, that organization must demonstrate capabilities to operate existing and planned systems. It should also be able to provide documented operating procedures that can be modified to support the facility needs.

Intrusion Monitoring Systems

Closed-circuit television (CCTV) systems have been used for years to protect critical facilities. These systems have improved considerably over the years to provide digital images that take up less storage space and be transmitted over TCP/IP-based networks. Their images can be combined with other alarm events to provide a total picture for guard response as well as event history. Digital systems that are activated in conjunction with motion detection or other alarms may be more effective because their activation signals a change to the guard who is assigned to watch them.

CCTV systems allow guards to keep watch on areas that are located remotely, are normally unmanned, or require higher surveillance, such as critical access points. These systems can reduce the need for additional manpower to provide control over critical areas. In many cases, their mere presence serves as a deterrent to unwanted behavior.

They may also contribute to employee safety by providing surveillance over parking areas, low traffic areas, and high-value functions such as cashier offices. A single guard in a central control center can spot problems and dispatch roving manpower to quickly resolve threats. In addition to the above, stored images may be used to assist law enforcement in apprehending violators and as evidence in a court of law.

Security requirements will vary with different organizations; however, CCTV may be useful in the following areas:

- Parking lots for employee and property safety
- Emergency doors where access is restricted
- Office areas during nonworking hours
- Server and telecommunications equipment rooms during nonworking hours
- Loading docks and delivery gates
- Cashier and check-processing areas
- Remote facilities where roving guards would be too costly
- Executive office areas in support of executive protection programs
- Mantrap gates to ensure all entry cards have been entered

Alarms and motion-detection systems are designed to signal the organization that an unusual or prohibited event has occurred. Doors that should not be used during normal business activity may be equipped with local sound alarms or with electronic sensors that signal a guard or activate surveillance systems. Motion detectors are often installed in areas that are normally unmanned. In some systems, motion detection is activated during nonbusiness hours and can be disabled or changed to allow for activities that are properly scheduled in those areas.

Many systems can be IP addressable over the backbone TCP/IP network, and alarm signals can be transmitted from multiple remote areas. It is important to note that IP-based systems may be subject to attack. The vendor of these systems must ensure that these systems are hack-protected against covert activities by unauthorized people.

Physical Access Control Measures

Physical access controls are as important as logical access controls to protect critical information resources. Multiple methods are available, including manual and automated systems. Often, cost is the deciding factor in their selection despite the risks inherent in those tools.

Access Policies

All good security begins with policies. Policies are the drivers of written procedures that must be in place to provide consistent best practices in the protection of people and information resources. Policies are the method by which management communicates its wishes. Policies are also used to set standards and assign responsibility for their enforcement. Once policies are developed, they should be published for easy access and be part of the employee awareness training program.

Policies define the process of granting and removing access based on need-to-know. If badges are employed, policies define how they are to be designed, worn, and used. Policies define who is allowed into restricted areas or how visitors are to be processed. There is no magic to developing policies, but they are required as a basic tool to protect information resources.

Keys and Cipher Locks

Keys and cipher (keypad) locks are the simplest to use and hardest to control in providing access to critical areas. They do not provide a means of identifying who is accessing a given area, nor do they provide an audit trail. Keys provide a slightly better security control than keypad locks in that the physical device must be provided to allow use. While they can be copied, that requires extra effort to accomplish. If keys are used to control access, they should be inventoried and stamped with the words *Do Not Duplicate*.

Cipher locks require that a person know the cipher code to enter an area. Once given out, use of this code cannot be controlled and may be passed throughout an organization by word of mouth. There is no audit trail for entry, nor is there authentication that it is used by an authorized user. Control methods consist of periodic code changes and shielding to prevent other people from viewing the authorized user's code entry. Use of these methods of entry control could be better protected through the use of CCTV.

Card Access Controls

Card access controls are considerably better tools than keys and cipher locks if they are used for identification and contain a picture of the bearer. Without pictures, they are only slightly better than keys because they are more difficult to duplicate. If given to another person to gain entry, the card must be returned for use by the authorized cardholder. Different types of card readers can be employed to provide ease of use (proximity readers) and different card identification technology. Adding biometrics to the process would provide added control along with increased cost and inconvenience that might be justified to protect the contents within.

The most effective card systems use a central control computer that can be programmed to provide different access levels depending on need, time zone controls that limit access to certain hours of the day, and an audit trail of when the card was used and where it was entered. Some systems even provide positive in and out controls that require a card to be used for both entry and exit. If a corresponding entry/exit transaction is not in the system, future entry will be denied until management investigation actions are taken.

Smart card technology is being developed to provide added security and functionality. Smart cards can have multiple uses that expand beyond mere physical access. Additional uses for this type of card include computer access authentication, encryption using digital certificates, and debit cards for employee purchases in the cafeteria or employee store. There is some controversy about multiple-use cards because a single device can be used to gain access to many different resources. If the employee smart card provides multiple access functions as well as purchasing functions, the cardholder will be less likely to loan the badge to an unauthorized person for use and will be more likely to report its loss.

Mantraps and Turnstiles

Additional controls can be provided through the use of mantraps and turnstiles. These devices prevent unauthorized tailgating and can be used to require inspection of parcels when combined with guard stations. These devices also force the use of a badge to enter through a control point and overcome the tendency for guards to allow entry because the person looks familiar to them. Mantraps and turnstiles can control this weakness if the badge is confiscated upon termination of access privileges. The use of positive entry/exit controls can be added to prevent card users from passing their card back through the control point to let a friend enter.

Fire Controls

Different fire control mechanisms must be employed to match the risks that are present in protected areas. Fire control systems may be as simple as a hand-held fire extinguisher or be combined with various detection mechanisms to provide automated activation. Expert advice should be used to match the proper system to the existing threats. In some cases, multiple systems may be used to ensure that fires do not reignite and cause serious damage.

Detectors and Alarms

Smoke and water detectors can provide early warning and alarm the guards that something dangerous may be happening. Alarms may also trigger fire prevention systems to activate. To be effective, they must be carefully placed and tested by experts in fire prevention.

Water-Based Systems

Water-based systems control fires by reducing temperatures below the combustion point. They are usually activated through overhead sprinklers to extinguish fires before they can spread. The problem with water-based systems is that they cause a certain amount of damage to the contents of areas they are designed to protect. In addition, they may cause flooding in adjacent areas if they are not detected and shut off quickly following an event.

Water-based systems may be either dry pipe or wet pipe systems. Wet pipe systems are always ready to go and are activated when heat or accidental means open the sprinkler heads. There is no delay or shut-off mechanism that can be activated prior to the start of water flow. Water in the pipes that connect to the sprinkler heads may become corroded, causing failure of the sprinkler heads to activate in an emergency.

Dry pipe systems are designed to allow some preventive action before they activate. These types of systems employ a valve to prevent the flow of water into the overhead pipes until a fire alarm event triggers water release. Dry pipe systems will not activate and cause damage if a sprinkler head is accidentally broken off. They also allow human intervention to override water flow if the system is accidentally activated.

Gas-Based Fire Extinguishing Systems

Halon-type systems are different from water-based systems in that they control fires by interrupting the chemical reactions needed to continue combustion. They replaced older gas systems such as carbon dioxide that controlled fires by replacing the oxygen with a gas (CO_2) that did not support the combustion process. Oxygen replacement systems were effective, but they were toxic to humans who might be in the CO_2-activated room due to the need for oxygen to survive.

Throughout the 1970s and 1980s, halon systems were the preferred method to protect computer and telecommunication rooms from fire damage because they extinguished the fire without damaging sensitive electronic equipment. Those systems could extinguish fires and yet allow humans to breathe and survive in the flooded room. The problem with halon is that it proved unfriendly to the ozone layer and was banned from new implementations by an international agreement (Montreal Protocol). There are numerous Clean Air Act and EPA regulations now in effect to govern the use of existing halon systems and supplies. Current regulations and information can be obtained by logging onto www.epa.gov/docs/ozone/ title6/snap/hal.html. This site also lists manufacturers of halon substitute systems.

Today, halon replacement systems are available that continue to extinguish fires, do not harm the ozone layer, and, most important, do not harm humans who may be in the gas-flooded room. Although these systems will not kill human inhabitants, most system manufacturers warn that people should leave the gas-flooded area within one minute of system activation. Current regulations do not dictate the removal of halon systems that are in place; however, any new or replacement halon systems must employ the newer ozone-friendly gas (e.g., FM 200).

Utility and Telecommunications Backup Requirements

Emergency Lighting

As stated before, modern computer rooms are usually lacking in windows or other sources of natural light. Therefore, when a power outage occurs, these rooms become very dark and exits become difficult to find. Even in normal offices, power outages may occur in areas that are staffed at night. In all of these cases, emergency lighting with exit signs must be installed to allow people to evacuate in an orderly and safe manner. Emergency lighting is usually provided by battery-equipped lamps that are constantly charged until activated.

UPS Systems

Uninterruptible power supply (UPS) systems ensure that a computing system can continue to run, or at least shut down in an orderly manner, if normal power is lost. Lower cost systems rely on battery backup to provide an orderly shutdown; motor generator backup systems used in conjunction with battery backup can provide continuous power as long as the engines receive fuel (usually diesel). As usual, cost is the driver for choosing the proper UPS system. More enlightened management will insist on a business impact analysis prior to making that decision to ensure that critical business needs are met.

Regardless of the type of system employed, periodic testing is required to ensure that the system will work when needed. Diesel systems should be tested weekly to ensure they work and to keep the engines properly lubricated.

Redundant Connections

Redundancy should be considered for facility electrical power, air conditioning, telecommunications connections, and water supplies. Certain systems such as UPS can be employed to mitigate the need for electrical redundancy. Telecommunications connectivity should be ensured with redundant connections. In this E-commerce world, telecommunications redundancy should also include connections to the Internet. Water is important to the staff, but environmental systems (cooling towers) may also depend on a reliable supply. In most cases, this redundancy can be provided with separate connections to the water main that is provided by the supporting community.

Summary

Physical security must be considered to provide a safe working environment for the people who visit and work in a facility. Although physical access controls must be employed for safety reasons, they also should prevent unauthorized access to critical computing resources.

Many tools are available to provide physical security that continues to be enhanced with current technology. Backbone networks and central control computers can support the protection of geographically separated facilities and operations. IP-supported systems can support the collection of large amounts of data from various sensors and control mechanisms and provide enhanced physical security while keeping manpower at a minimum.

The information security practitioner must become aware of existing physical security issues and be involved. If a separate department provides physical security, coordination with them becomes important to a total security approach. If information security organizations are assigned to provide physical security, they must become aware of the tools that are available and determine where to employ them.

161

Closed-Circuit Television and Video Surveillance

David Litzau, CISSP

In June 1925, Charles Francis Jenkins successfully transmitted a series of motion pictures of a small windmill to a receiving facility over five miles away. The image included 48 lines of resolution and lasted ten minutes. This demonstration would move the television from an engineer's lark to reality. By 1935, *Broadcast* magazine listed 27 different television broadcast facilities across the nation, some with as many as 45 hours of broadcast a week. Although the television set was still a toy for the prosperous, the number of broadcast facilities began to multiply rapidly.

On August 10, 1948, the American Broadcasting Company (ABC) debuted the television show *Candid Camera*. The basis of the show was to observe the behavior of people in awkward circumstances — much to the amusement of the viewing audience — by a hidden camera. This human behavior by surreptitious observation did not go unnoticed by psychologists and security experts of the time. Psychologists recognized the hidden camera as a way to study human behavior, and for security experts it became a tool of observation. Of particular note to both was the profound effect on behavior that the presence of a camera had on people once they became aware that they were being observed.

Security experts would have to wait for advances in technology before the emerging technology could be used. Television was based on vacuum tube technology and the use of extensive broadcast facilities. It would be the space race of the late 1950s and 1960s that would bring the television and its cameras into the realm of security. Two such advances that contributed were the mass production of transistors and the addition of another new technology known as videotape. The transistor replaced bulky, failure-prone vacuum tubes and resulted in television cameras becoming smaller and more affordable. The videotape machine meant that the images no longer had to be broadcast; the images could be collected through one or more video cameras and the data transmitted via a closed circuit of wiring to be viewed on a video monitor or recorded on tape. This technology became known as closed-circuit television, or CCTV.

In the early 1960s, CCTV would be embraced by the Department of Defense as an aid for perimeter security. In the private sector, security experts for merchants were quick to see the value of such technology as an aid in the prevention of theft by customers and employees. Today, unimagined advances in the technology in cameras and recording devices have brought CCTV into the home and workplace in miniature form.

Why CCTV?

Information security is a multifaceted process, and the goal is to maintain security of the data processing facility and the assets within. Typically, those assets can be categorized as hardware, software, data, and people, which also involves the policies and procedures that govern the behavior of those people. With the possible exception of software, CCTV has the ability to provide defense of these assets on several fronts.

To Deter

The presence of cameras both internally and externally has a controlling effect on those who step into the field of view. In much the same way that a small padlock on a storage shed will keep neighbors from helping themselves to garden tools when the owner is not at home, the camera's lens tends to keep personnel from behaving outside of right and proper conduct. In the case of the storage shed, the lock sends the message that the contents are for the use of those with the key to access it, but it would offer little resistance to a determined thief. Likewise, the CCTV camera sends a similar message and will deter an otherwise honest employee from stepping out of line, but it will not stop someone determined to steal valuable assets. It becomes a conscious act to violate policies and procedures because the act itself will likely be observed and recorded.

With the cameras at the perimeter, those looking for easy targets will likely move on, just as employees within the facility will tend to conduct themselves in a manner that complies with corporate policies and procedures. With cameras trained on data storage devices, it becomes difficult to physically access the device unobserved, thereby deterring the theft of the data contained within. The unauthorized installation or removal of hardware can be greatly deterred by placing cameras in a manner that permits the observation of portals such as windows or doors. Overall, the statistics of crimes in the presence of CCTV cameras is dramatically reduced.

To Detect

Of particular value to the security professional is the ability of a CCTV system to provide detection. The eyes of a security guard can only observe a single location at a time, but CCTV systems can be configured in such a manner that a single pair of eyes can observe a bank of monitors. Further, each monitor can display the output of multiple cameras. The net effect is that the guard in turn can observe dozens of locations from a single observation point. During periods of little or no traffic, a person walking into the view of a camera is easily detected. Placing the camera input from high-security and high-traffic locations in the center of the displays can further enhance the coverage, because an intruder entering the field of view on a surrounding monitor would be easily detected even though the focus of attention is at the center of the monitors. Technology is in use that will evaluate the image field; and if the content of the image changes, an alarm can be sounded or the mode of recording changed to capture more detail of the image. Further, with the aid of recording equipment, videotape recordings can be reviewed in fast-forward or rewind to quickly identify the presence of intruders or other suspicious activities.

To Enforce

The human eyewitness has been challenged in the court of law more often in recent history. The lack of sleep, age of the witness, emotional state, etc., can all come to bear on the validity of an eyewitness statement. On the other hand, the camera does not get tired; video recording equipment is not susceptible to such human frailties. A video surveillance recording can vastly alter the outcome of legal proceedings and has an excellent track record in swaying juries as to the guilt or innocence of the accused. Often, disciplinary action is not even required once the alleged act is viewed on video by the accused, thereby circumventing the expense of a trial or arbitration. If an act is caught on tape that requires legal or disciplinary action, the tape ensures that there is additional evidence to support the allegations.

With the combined abilities of deterrence, detection, and enforcement of policies and procedures over several categories of assets, the CCTV becomes a very effective aid in the process of information security, clearly an aid that should be carefully considered when selecting countermeasures and defenses.

CCTV Components

One of the many appealing aspects of CCTV is the relative simplicity of its component parts. As in any system, the configuration can only be as good as the weakest link. Inexpensive speakers on the highest quality sound system will result in inexpensive quality sound. Likewise, a poor quality component in a CCTV system produces poor results. There are basically four groups of components:

1. Cameras
2. Transmission media

3. Monitors
4. Peripherals

The Camera

The job of the camera is to collect images of the desired viewing area and is by far the component that requires the most consideration when configuring a CCTV system. In a typical installation, the camera relies on visible light to illuminate the target; the reflected light is then collected through the camera lens and converted into an electronic signal that is transmitted back through the system to be processed.

The camera body contains the components to convert visible light to electronic signals. There are still good-quality, vacuum-tube cameras that produce an analog signal, but most cameras in use today are solid-state devices producing digital signal output. Primary considerations when selecting a camera are the security objectives. The sensitivity of a camera refers to the number of receptors on the imaging surface and will determine the resolution of the output; the greater the number of receptors, the greater the resolution. If there is a need to identify humans with a high level of certainty, one should consider a color camera with a high level of sensitivity. On the other hand, if the purpose of the system is primarily to observe traffic, a simple black-and-white camera with a lower sensitivity will suffice.

The size of cameras can range from the outwardly overt size of a large shoebox to the very covert size of a matchbox. Although the miniaturized cameras are capable of producing a respectable enough image to detect the presence of a human, most do not collect enough reflective light to produce an image quality that could be used for positive identification. This is an area of the technology that is seeing rapid improvement.

There are so many considerations in the placement of cameras that an expert should be consulted for the task. Some of those considerations include whether the targeted coverage is internal or external to the facility. External cameras need to be positioned so that all approaches to the facility can be observed, thereby eliminating blind spots. The camera should be placed high enough off the ground so that it cannot be easily disabled, but not so high that the images from the scene only produce the tops of people's heads and the camera is difficult to service. The camera mount can have motor drives that will permit aiming left and right (panning) or up and down (tilting), commonly referred to as a *pan/tilt drive*. Additionally, if the camera is on the exterior of the facility, it may require the use of a sunshade to prevent the internal temperature from reaching damaging levels. A mount that can provide heating to permit de-icing should be considered in regions of extreme cold so that snow and ice will not damage the pan/tilt drive. Internal cameras require an equal amount of consideration; and, again, the area to be covered and ambient light will play a large part in the placement. Cameras may be overt or covert and will need to be positioned such that people coming or going from highly valued assets or portals can be observed.

Because the quality of the image relies in large part on the reflective light, the lens on the camera must be carefully selected to make good use of available light. The cameras should be placed in a manner that will allow the evening lighting to work with the camera to provide front lighting (lights that shine in the same direction that the camera is aimed) to prevent shadowing of approaching people or objects. Constant adjustments must be made to lenses to accommodate the effects of a constantly changing angle of sunlight, changing atmospheric conditions, highly reflective rain or snowfall, and the transition to artificial lighting in the evening; all affect ambient light. This is best accomplished with the use of an automatic iris. The iris in a camera, just as in the human eye, opens and closes to adjust the amount of light that reaches the imaging surface. Direct exposure to an intense light source will result in blossoming of the image — where the image becomes all white and washes out the picture to the point where nothing is seen — and can also result in serious damage to the imaging surface within the camera.

The single most-important element of the camera is the lens. There are basically four types of lenses: standard, wide-angle, telephoto, and zoom. When compared to human eyesight, the standard lens is the rough equivalent; the wide-angle takes in a scene wider than what humans can see; and the telephoto is magnified and roughly equivalent to looking through a telescope. All are fixed focal length lenses. The characteristics of these three combined are a zoom lens.

The Transmission Media

Transmission media refer to how the video signal from the cameras will be transported to the multiplexer or monitor. This is typically some type of wiring.

Coaxial Cable

By far the most commonly used media are coaxial cables. There are varying grades of coaxial cable, and the quality of the cable will have a profound effect on the quality of the video. Coaxial cable consists of a single center conductor with a piezoelectric insulator surrounding it. The insulation is then encased in a foil wrap and further surrounded by a wire mesh. A final coating of weather-resistant insulation is placed around the entire bundle to produce a durable wire that provides strong protection for the signal as it transits through the center conductor. The center conductor can be a single solid wire or a single conductor made up of multiple strands of wire. Engineers agree that the best conductor for a video signal is pure copper. The amount of shielding will determine the level of protection for the center conductor. The shielding is grounded at both ends of the connection and thereby shunts extraneous noise from electromagnetic radiation to ground.

Although 100 percent pure copper is an excellent conductor of the electronic signal, there is still a level of internal resistance that will eventually degrade the signal's strength. To overcome the loss of signal strength, the diameter of the center conductor and the amount of shielding can be increased to obtain greater transmission lengths before an in-line repeater/amplifier will be required. This aspect of the cable is expressed in an industry rating. The farther the distance the signal must traverse, the higher the rating of the coaxial cable that should be used or noticeable signal degradation will occur.

Some examples are:

- RG59/U rated to carry the signal up to distances of 1000 feet
- RG6/U rated to carry a signal up to 1500 feet
- RG11/U rated to carry a signal up to 3000 feet

One of the benefits of coaxial cable is that it is easy to troubleshoot the media should there be a failure. A device that sends a square-wave signal down the wire (time domain reflectometer) can pinpoint the location of excessive resistance or a broken wire. Avoid using a solid center conductor wire on cameras mounted on a pan/tilt drive because the motion of the camera can fatigue the wire and cause a failure; thus, multi-strand wire should be used.

Fiber-Optic Cable

Fiber-optic cable is designed to transmit data in the form of light pulses. It typically consists of a single strand of highly purified silica (glass), smaller than a human hair, surrounded by another jacket of lower grade glass. This bundle is then clad in a protective layer to prevent physical damage to the core. The properties of the fiber-optic core are such that the outer surface of the center fiber has a mirror effect, thereby reflecting the light back into itself. This means that the cable can be curved, and it has almost no effect on the light pulses within. This effect, along with the fact that the frequency spectrum that spans the range of light is quite broad, produces an outstanding medium for the transfer of a signal. There is very little resistance or degradation of the signal as it traverses the cable, and the end result is much greater transmission lengths and available communication channels when compared to a metallic medium.

The reason that fiber-optics has not entirely replaced its coaxial counterpart is that the cost is substantially higher. Because the fiber does not conduct any electrical energy, the output signal must be converted to light pulses. This conversion is known as modulation and is accomplished using a laser. Once converted to light pulses, the signal is transferred into the fiber-optic cable. Because the fiber of the cable is so small, establishing good connections and splices is critical. Any misalignment or damage to the fiber will result in reflective energy or complete termination of the signal. Therefore, a skilled technician with precision splicing and connection tools is required. This cost, along with modulators/demodulators and the price of the medium, adds substantial cost to the typical CCTV installation.

For the additional cost, some of the benefits include generous gains in bandwidth. This means that more signals carrying a greater amount of data can be realized. Adding audio from microphones, adjustment signals to control zoom lenses and automatic irises, and additional cameras can be accommodated. The medium is smaller and lighter and can carry a signal measured in miles instead of feet. Because there is no electromagnetic energy to create compromising emanations, and a splice to tap the connection usually creates an easily detected interruption of the signal, there is the additional benefit of a high level of assurance of data integrity and security. In an environment of remote locations or a site containing highly valued assets, these benefits easily offset the additional cost of fiber-optic transmission.

Wireless Transmission

The option of not using wiring at all is available for CCTV. The output signals from cameras can be converted to radio frequency, light waves, or microwave signals for transmission. This may be the only viable option for some remote sites and can range from neighboring buildings using infrared transceivers to a satellite link for centralized monitoring of remote sites throughout the globe. Infrared technology must be configured in a line-of-sight manner and has a limited range of distance. Radio frequency and microwaves can get substantial improvements in distance but will require the use of repeaters and substations to traverse distances measured in miles. The more obstacles that must be negotiated (i.e., buildings, mountains, etc.), the greater the degradation of the signal that takes place.

Two of the biggest drawbacks of utilizing wireless are that the signal is vulnerable to atmospheric conditions and, as in any wireless transmission, easily intercepted and inherently insecure. Everything from the local weather to solar activity can affect the quality of the signal. From a security standpoint, the transmission is vulnerable to interception, which could reveal to the viewer the activity within a facility and compromise other internal defenses. Further, the signal could be jammed or modified to render the system useless or to provide false images. If wireless transmission is to be utilized, some type of signal scrambling or channel-hopping technology should be utilized to enhance the signal confidentiality and integrity.

Some of the more recent trends in transmission media have been the use of existing telephone lines and computer networking media. The dial-up modem has been implemented in some installations with success, but the limited amount of data that can be transmitted results in slow image refreshing; and control commands to the camera (focus, pan, tilt, etc.) are slow to respond. The response times and refresh rates can be substantially increased through the use of ISDN phone line technology. Some recent advances in data compression, and protocols that allow video over IP, have moved the transmission possibilities into existing computer network cabling.

The Monitor

The monitor is used to convert the signal from cameras into a visible image. The monitor can be used for real-time observation or the playback of previously recorded data.

Color or black-and-white video monitors are available but differ somewhat from a standard television set. A television set will come with the electronics to convert signals broadcast on the UHF and VHF frequency spectrums and demodulate those signals into a visible display of the images. The CCTV monitor does not come with such electronics and is designed to process the signals of a standard 75-ohm impedance video signal into visible images. This does not mean that a television set cannot be used as a video monitor, but proper attenuation equipment will be needed to convert the video into a signal that the television can process.

The lines of resolution determine detail and the overall sharpness of the image. The key to reproducing a quality image is matching as closely as possible the resolution of the monitor to the camera; but it is generally accepted that, if a close match is not made, then it is better to have a monitor with a greater resolution. The reason for this is that a 900-line monitor displaying an image of 300 lines of resolution will provide three available lines for each line of image. The image will be large and appear less crisp; but if at a later date the monitor is used in a split-screen fashion to display the output from several cameras on the screen at the same time, there will be enough resolution for each image. On the other hand, if the resolution of the monitor is lower than that of the camera, detail will be lost because the entire image cannot be displayed.

The size of the monitor to be used is based on several factors. The more images to be viewed, the greater the number of monitors. A single monitor is capable of displaying the output from several cameras on the same screen (see multiplexers), but this still requires a comfortable distance between the viewer and monitor. Although not exactly scientific, a general rule of thumb is that the viewer's fist at the end of an extended arm should just cover the image. This would place the viewer farther away from the monitor for a single image and closer if several images were displayed.

The Peripherals

A multiplexer is a hardware device that is capable of receiving the output signal from multiple cameras and processing those signals in several ways. The most common use is to combine the inputs from selected cameras into a single output such that the group of inputs is displayed on a single monitor. A multiplexer is capable

of accepting from four to 32 separate signals and provides video enhancement, data compression, and storing or output to a storage device. Some of the additional features available from a multiplexer include alarm modes that will detect a change to an image scene to alert motion and the ability to convert analog video signals into digital format. Some multiplexers have video storage capabilities, but most provide output that is sent to a separate storage device.

A CCTV system can be as simple as a camera, transmission medium, and a monitor. This may be fine if observation is the goal of the system; but if the intent is part of a security system, storage of captured images should be a serious consideration. The output from cameras can be stored and retrieved to provide nearly irrefutable evidence for legal proceedings.

There are several considerations in making a video storage decision. Foremost is the desired quality of the retrieved video. The quality of the data always equates to quantity of storage space required.

The primary difference in storage devices is whether the data will be stored in analog or digital format. The options for analog primarily consist of standard three-quarter-inch VHS tape or higher quality one-inch tape. The measure of quantity for analog is time, where the speed of recording and tape length will determine the amount of time that can be recorded. To increase the amount of time that a recording spans, one of the best features available in tape is time-lapse recording. Time-lapse videocassette recorders (VCRs) reduce the number of frames per second (fps) that are recorded. This equates to greater spans of time on less tape, but the images will appear as a series of sequential still images when played back. There is the potential of a critical event taking place between pictures and thereby losing its evidentiary value. This risk can be offset if the VCR is working in conjunction with a multiplexer that incorporates motion detection. Then the FPS can be increased to record more data from the channel with the activity. Another consideration of analog storage medium is that the shelf life is limited. Usually if there is no event of significance, then tapes can be recorded over existing data; but if there is a need for long-term storage, the quality of the video will degrade with time.

Another option for the storage of data is digital format. There are many advantages to utilizing digital storage media. The beauty of digital is that the signal is converted to binary 1s and 0s, and once converted the data is ageless. The data can then be stored on any data processing hardware, including hard disk drives, tapes, DVDs, magneto-optical disks, etc. By far the best-suited hardware is the digital video recorder (DVR). Some of the capabilities of DVRs may include triplex functions (simultaneous video observation, playback, and recording), multiple camera inputs, multi-screen display outputs, unlimited recording time by adding multiple hard disk drives, hot-swappable RAID, multiple trigger events for alarms, and tape archiving of trigger events. Because the data can be indexed on events such as time, dates, and alarms, the video can be retrieved for playback almost instantly.

Whether analog or digital, the sensitivity of the cameras used, frames recorded each second, whether the signal is in black and white or color, and the length of time to store will impact the amount of storage space required.

Putting It All Together

By understanding the stages of implementation and how hardware components are integrated, the security professional will have a much higher likelihood of successfully integrating a CCTV system. There is no typical installation, and every site will have its unique characteristics to accommodate; but there is a typical progression of events from design to completion.

- *Define the purpose.* If observation of an entrance is the only goal, there will be little planning to consider. Will the quality of images be sufficient to positively identify an individual? Will there be a requirement to store image data, and what will be the retention period? Should the presence of a CCTV system be obvious with the presence of cameras, or will they be hidden? Ultimately, the question becomes: What is the purpose of implementing and what is to be gained?
- *Define the surveyed area.* Complete coverage for the exterior and interior of a large facility or multiple facilities will require a substantial budget. If there are financial restraints, then decisions will have to be made concerning what areas will be observed. Some of the factors that will influence that decision may be the value of the assets under scrutiny and the security requirements in a particular location.

- *Select appropriate cameras.* At this point in the planning, a professional consultation should be considered. Internal surveillance is comparatively simpler than external because the light levels are consistent; but external surveillance requires an in-depth understanding of how light, lenses, weather, and other considerations will affect the quality of the images. Placement of cameras can make a substantial difference in the efficiency of coverage and the effectiveness of the images that will be captured.
- *Selection and placement of monitors.* Considerations that need to be addressed when planning the purchase of monitors include the question of how many camera inputs will have to be observed at the same time. How many people will be doing the observation simultaneously? How much room space is available in the monitoring room? Is there sufficient air conditioning to accommodate the heat generated by large banks of monitors?
- *Installation of transmission media.* Once the camera locations and the monitoring location have been determined, the installation of the transmission media can then begin. A decision should have already been made on the type of media that will be utilized and sufficient quantities ordered. Technicians skilled in installation, splicing, and testing will be required.
- *Peripherals.* If the security requirements are such that image data must be recorded and retained, then storage equipment will have to be installed. Placement of multiplexers, switches, universal power supplies, and other supporting equipment will have to be planned in advance. Personnel access controls are critical to areas containing such equipment.

Summary

CCTV systems are by no means a guarantee of security, but the controlling effect they have on human behavior cannot be dismissed easily. The mere presence of a camera, regardless of whether it works, has proven to be invaluable in the security industry as a deterrent.

Defense-in-depth is the mantra of the information security industry. It is the convergence of many layers of protection that will ultimately provide the highest level of assurance, and the physical security of a data processing facility is often the weakest layer. There is little else that can compare to a properly implemented CCTV system to provide security of the facility, data, and people, as well as enforcement of policies and procedures.

Works Cited

1. Kruegle, Herman, *CCTV Surveillance: Video Technologies and Practices*, 3rd ed., Butterworth-Heinemann, 1999.
2. Axiom Engineering, CCTV Video Surveillance Systems, http://www.axiomca.com/services/ cctv.htm.
3. Kriton Electronics, Design Basics, http://shop.store.yahoo.com/kriton/secsysselrul.html.
4. Video Surveillance Cameras and CCTV Monitors, http://www.pelikanind.com/.
5. CCTV — Video Surveillance Cameras Monitors Switching Units, http://www.infosyssec.org/infosyssec/cctv.htm.

162

Types of Information Security Controls

Harold F. Tipton, CISSP

Security is generally defined as the freedom from danger or as the condition of safety. Computer security, specifically, is the protection of data in a system against unauthorized disclosure, modification, or destruction and protection of the computer system itself against unauthorized use, modification, or denial of service. Because certain computer security controls inhibit productivity, security is typically a compromise toward which security practitioners, system users, and system operations and administrative personnel work to achieve a satisfactory balance between security and productivity.

Controls for providing information security can be physical, technical, or administrative. These three categories of controls can be further classified as either preventive or detective. Preventive controls attempt to avoid the occurrence of unwanted events, whereas detective controls attempt to identify unwanted events after they have occurred. Preventive controls inhibit the free use of computing resources and therefore can be applied only to the degree that the users are willing to accept. Effective security awareness programs can help increase users' level of tolerance for preventive controls by helping them understand how such controls enable them to trust their computing systems. Common detective controls include audit trails, intrusion detection methods, and checksums.

Three other types of controls supplement preventive and detective controls. They are usually described as deterrent, corrective, and recovery. Deterrent controls are intended to discourage individuals from intentionally violating information security policies or procedures. These usually take the form of constraints that make it difficult or undesirable to perform unauthorized activities or threats of consequences that influence a potential intruder to not violate security (e.g., threats ranging from embarrassment to severe punishment).

Corrective controls either remedy the circumstances that allowed the unauthorized activity or return conditions to what they were before the violation. Execution of corrective controls could result in changes to existing physical, technical, and administrative controls. Recovery controls restore lost computing resources or capabilities and help the organization recover monetary losses caused by a security violation.

Deterrent, corrective, and recovery controls are considered to be special cases within the major categories of physical, technical, and administrative controls; they do not clearly belong in either preventive or detective categories. For example, it could be argued that deterrence is a form of prevention because it can cause an intruder to turn away; however, deterrence also involves detecting violations, which may be what the intruder fears most. Corrective controls, on the other hand, are not preventive or detective, but they are clearly linked with technical controls when anti-viral software eradicates a virus or with administrative controls when backup procedures enable restoring a damaged database. Finally, recovery controls are neither preventive nor detective but are included in administrative controls as disaster recovery or contingency plans.

Because of these overlaps with physical, technical, and administrative controls, the deterrent, corrective, and recovery controls are not discussed further in this chapter. Instead, the preventive and detective controls within the three major categories are examined.

Physical Controls

Physical security is the use of locks, security guards, badges, alarms, and similar measures to control access to computers, related equipment (including utilities), and the processing facility itself. In addition, measures are required for protecting computers, related equipment, and their contents from espionage, theft, and destruction or damage by accident, fire, or natural disaster (e.g., floods and earthquakes).

Preventive Physical Controls

Preventive physical controls are employed to prevent unauthorized personnel from entering computing facilities (i.e., locations housing computing resources, supporting utilities, computer hard copy, and input data media) and to help protect against natural disasters. Examples of these controls include:

- Backup files and documentation
- Fences
- Security guards
- Badge systems
- Double door systems
- Locks and keys
- Backup power
- Biometric access controls
- Site selection
- Fire extinguishers

Backup Files and Documentation

Should an accident or intruder destroy active data files or documentation, it is essential that backup copies be readily available. Backup files should be stored far enough away from the active data or documentation to avoid destruction by the same incident that destroyed the original. Backup material should be stored in a secure location constructed of noncombustible materials, including two-hour-rated fire walls. Backups of sensitive information should have the same level of protection as the active files of this information; it is senseless to provide tight security for data on the system but lax security for the same data in a backup location.

Fences

Although fences around the perimeter of the building do not provide much protection against a determined intruder, they do establish a formal no-trespassing line and can dissuade the simply curious person. Fences should have alarms or should be under continuous surveillance by guards, dogs, or TV monitors.

Security Guards

Security guards are often stationed at the entrances of facilities to intercept intruders and ensure that only authorized persons are allowed to enter. Guards are effective in inspecting packages or other hand-carried items to ensure that only authorized, properly described articles are taken into or out of the facility. The effectiveness of stationary guards can be greatly enhanced if the building is wired with appropriate electronic detectors with alarms or other warning indicators terminating at the guard station. In addition, guards are often used to patrol unattended spaces inside buildings after normal working hours to deter intruders from obtaining or profiting from unauthorized access.

Badge Systems

Physical access to computing areas can be effectively controlled using a badge system. With this method of control, employees and visitors must wear appropriate badges whenever they are in access-controlled areas. Badge-reading systems programmed to allow entrance only to authorized persons can then easily identify intruders.

Double Door Systems

Double door systems can be used at entrances to restricted areas (e.g., computing facilities) to force people to identify themselves to the guard before they can be released into the secured area. Double doors are an excellent way to prevent intruders from following closely behind authorized persons and slipping into restricted areas.

Locks and Keys

Locks and keys are commonly used for controlling access to restricted areas. Because it is difficult to control copying of keys, many installations use cipher locks (i.e., combination locks containing buttons that open the lock when pushed in the proper sequence). With cipher locks, care must be taken to conceal which buttons are being pushed to avoid a compromise of the combination.

Backup Power

Backup power is necessary to ensure that computer services are in a constant state of readiness and to help avoid damage to equipment if normal power is lost. For short periods of power loss, backup power is usually provided by batteries. In areas susceptible to outages of more than 15 to 30 minutes, diesel generators are usually recommended.

Biometric Access Controls

Biometric identification is a more-sophisticated method of controlling access to computing facilities than badge readers, but the two methods operate in much the same way. Biometrics used for identification include fingerprints, handprints, voice patterns, signature samples, and retinal scans. Because biometrics cannot be lost, stolen, or shared, they provide a higher level of security than badges. Biometric identification is recommended for high-security, low-traffic entrance control.

Site Selection

The site for the building that houses the computing facilities should be carefully chosen to avoid obvious risks. For example, wooded areas can pose a fire hazard, areas on or adjacent to an earthquake fault can be dangerous and sites located in a flood plain are susceptible to water damage. In addition, locations under an aircraft approach or departure route are risky, and locations adjacent to railroad tracks can be susceptible to vibrations that can precipitate equipment problems.

Fire Extinguishers

The control of fire is important to prevent an emergency from turning into a disaster that seriously interrupts data processing. Computing facilities should be located far from potential fire sources (e.g., kitchens or cafeterias) and should be constructed of noncombustible materials. Furnishings should also be noncombustible. It is important that appropriate types of fire extinguishers be conveniently located for easy access. Employees must be trained in the proper use of fire extinguishers and in the procedures to follow should a fire break out.

Automatic sprinklers are essential in computer rooms and surrounding spaces and when expensive equipment is located on raised floors. Sprinklers are usually specified by insurance companies for the protection of any computer room that contains combustible materials. However, the risk of water damage to computing equipment is often greater than the risk of fire damage. Therefore, carbon dioxide extinguishing systems were developed; these systems flood an area threatened by fire with carbon dioxide, which suppresses fire by removing oxygen from the air. Although carbon dioxide does not cause water damage, it is potentially lethal to people in the area and is now used only in unattended areas.

Current extinguishing systems flood the area with halon, which is usually harmless to equipment and less dangerous to personnel than carbon dioxide. At a concentration of about 10percent, halon extinguishes fire and can be safely breathed by humans. However, higher concentrations can eventually be a health hazard. In addition, the blast from releasing halon under pressure can blow loose objects around and can be a danger to equipment and personnel. For these reasons and because of the high cost of halon, it is typically used only under raised floors in computer rooms. Because it contains chlorofluorocarbons, it will soon be phased out in favor of a gas that is less hazardous to the environment.

Detective Physical Controls

Detective physical controls warn protective services personnel that physical security measures are being violated. Examples of these controls include:

- Motion detectors
- Smoke and fire detectors
- Closed-circuit television monitors
- Sensors and alarms

Motion Detectors

In computing facilities that usually do not have people in them, motion detectors are useful for calling attention to potential intrusions. Motion detectors must be constantly monitored by guards.

Fire and Smoke Detectors

Fire and smoke detectors should be strategically located to provide early warning of a fire. All fire detection equipment should be tested periodically to ensure that it is in working condition.

Closed-Circuit Television Monitors

Closed-circuit televisions can be used to monitor the activities in computing areas where users or operators are frequently absent. This method helps detect individuals behaving suspiciously.

Sensors and Alarms

Sensors and alarms monitor the environment surrounding the equipment to ensure that air and cooling water temperatures remain within the levels specified by equipment design. If proper conditions are not maintained, the alarms summon operations and maintenance personnel to correct the situation before a business interruption occurs.

Technical Controls

Technical security involves the use of safeguards incorporated in computer hardware, operations or applications software, communications hardware and software, and related devices. Technical controls are sometimes referred to as logical controls.

Preventive Technical Controls

Preventive technical controls are used to prevent unauthorized personnel or programs from gaining remote access to computing resources. Examples of these controls include:

- Access control software
- Antivirus software
- Library control systems
- Passwords
- Smart cards
- Encryption
- Dial-up access control and callback systems

Access Control Software

The purpose of access control software is to control sharing of data and programs between users. In many computer systems, access to data and programs is implemented by access control lists that designate which users are allowed access. Access control software provides the ability to control access to the system by establishing that only registered users with an authorized log-on ID and password can gain access to the computer system.

After access to the system has been granted, the next step is to control access to the data and programs residing in the system. The data or program owner can establish rules that designate who is authorized to use the data or program.

Anti-Virus Software

Viruses have reached epidemic proportions throughout the microcomputing world and can cause processing disruptions and loss of data as well as significant loss of productivity while cleanup is conducted. In addition, new viruses are emerging at an ever-increasing rate — currently about one every 48 hours. It is recommended that anti-virus software be installed on all microcomputers to detect, identify, isolate, and eradicate viruses. This software must be updated frequently to help fight new viruses. In addition, to help ensure that viruses are intercepted as early as possible, anti-virus software should be kept active on a system, not used intermittently at the discretion of users.

Library Control Systems

These systems require that all changes to production programs be implemented by library control personnel instead of the programmers who created the changes. This practice ensures separation of duties, which helps prevent unauthorized changes to production programs.

Passwords

Passwords are used to verify that the user of an ID is the owner of the ID. The ID–password combination is unique to each user and therefore provides a means of holding users accountable for their activity on the system.

Fixed passwords that are used for a defined period of time are often easy for hackers to compromise; therefore, great care must be exercised to ensure that these passwords do not appear in any dictionary. Fixed passwords are often used to control access to specific databases. In this use, however, all persons who have authorized access to the database use the same password; therefore, no accountability can be achieved.

Currently, dynamic or one-time passwords, which are different for each log-on, are preferred over fixed passwords. Dynamic passwords are created by a token that is programmed to generate passwords randomly.

Smart Cards

Smart cards are usually about the size of a credit card and contain a chip with logic functions and information that can be read at a remote terminal to identify a specific user's privileges. Smart cards now carry prerecorded, usually encrypted access control information that is compared with data that the user provides (e.g., a personal ID number or biometric data) to verify authorization to access the computer or network.

Encryption

Encryption is defined as the transformation of plaintext (i.e., readable data) into ciphertext (i.e., unreadable data) by cryptographic techniques. Encryption is currently considered to be the only sure way of protecting data from disclosure during network transmissions.

Encryption can be implemented with either hardware or software. Software-based encryption is the least expensive method and is suitable for applications involving low-volume transmissions; the use of software for large volumes of data results in an unacceptable increase in processing costs. Because there is no overhead associated with hardware encryption, this method is preferred when large volumes of data are involved.

Dial-Up Access Control and Callback Systems

Dial-up access to a computer system increases the risk of intrusion by hackers. In networks that contain personal computers or are connected to other networks, it is difficult to determine whether dial-up access is available or not because of the ease with which a modem can be added to a personal computer to turn it into a dial-up access point. Known dial-up access points should be controlled so that only authorized dial-up users can get through.

Currently, the best dial-up access controls use a microcomputer to intercept calls, verify the identity of the caller (using a dynamic password mechanism), and switch the user to authorized computing resources as requested. Previously, call-back systems intercepted dial-up callers, verified their authorization and called them back at their registered number, which at first proved effective; however, sophisticated hackers have learned how to defeat this control using call-forwarding techniques.

Detective Technical Controls

Detective technical controls warn personnel of violations or attempted violations of preventive technical controls. Examples of these include audit trails and intrusion detection expert systems, which are discussed in the following sections.

Audit Trails

An audit trail is a record of system activities that enables the reconstruction and examination of the sequence of events of a transaction, from its inception to output of final results. Violation reports present significant, security-oriented events that may indicate either actual or attempted policy transgressions reflected in the audit trail. Violation reports should be frequently and regularly reviewed by security officers and database owners to identify and investigate successful or unsuccessful unauthorized accesses.

Intrusion Detection Systems

These expert systems track users (on the basis of their personal profiles) while they are using the system to determine whether their current activities are consistent with an established norm. If not, the user's session can be terminated or a security officer can be called to investigate. Intrusion detection can be especially effective in cases in which intruders are pretending to be authorized users or when authorized users are involved in unauthorized activities.

Administrative Controls

Administrative, or personnel, security consists of management constraints, operational procedures, accountability procedures, and supplemental administrative controls established to provide an acceptable level of protection for computing resources. In addition, administrative controls include procedures established to ensure that all personnel who have access to computing resources have the required authorizations and appropriate security clearances.

Preventive Administrative Controls

Preventive administrative controls are personnel-oriented techniques for controlling people's behavior to ensure the confidentiality, integrity, and availability of computing data and programs. Examples of preventive administrative controls include:

- Security awareness and technical training
- Separation of duties
- Procedures for recruiting and terminating employees
- Security policies and procedures
- Supervision
- Disaster recovery, contingency, and emergency plans
- User registration for computer access

Security Awareness and Technical Training

Security awareness training is a preventive measure that helps users to understand the benefits of security practices. If employees do not understand the need for the controls being imposed, they may eventually circumvent them and thereby weaken the security program or render it ineffective.

Technical training can help users prevent the most common security problem — errors and omissions — as well as ensure that they understand how to make appropriate backup files and detect and control viruses. Technical training in the form of emergency and fire drills for operations personnel can ensure that proper action will be taken to prevent such events from escalating into disasters.

Separation of Duties

This administrative control separates a process into component parts, with different users responsible for different parts of the process. Judicious separation of duties prevents one individual from obtaining control of an entire process and forces collusion with others in order to manipulate the process for personal gain.

Recruitment and Termination Procedures

Appropriate recruitment procedures can prevent the hiring of people who are likely to violate security policies. A thorough background investigation should be conducted, including checking on the applicant's criminal

history and references. Although this does not necessarily screen individuals for honesty and integrity, it can help identify areas that should be investigated further.

Three types of references should be obtained: (1) employment, (2) character, and (3) credit. Employment references can help estimate an individual's competence to perform, or be trained to perform, the tasks required on the job. Character references can help determine such qualities as trustworthiness, reliability, and ability to get along with others. Credit references can indicate a person's financial habits, which in turn can be an indication of maturity and willingness to assume responsibility for one's own actions.

In addition, certain procedures should be followed when any employee leaves the company, regardless of the conditions of termination. Any employee being involuntarily terminated should be asked to leave the premises immediately upon notification, to prevent further access to computing resources. Voluntary terminations may be handled differently, depending on the judgment of the employee's supervisors, to enable the employee to complete work in process or train a replacement.

All authorizations that have been granted to an employee should be revoked upon departure. If the departing employee has the authority to grant authorizations to others, these other authorizations should also be reviewed. All keys, badges, and other devices used to gain access to premises, information, or equipment should be retrieved from the departing employee. The combinations of all locks known to a departing employee should be changed immediately. In addition, the employee's log-on IDs and passwords should be canceled, and the related active and backup files should be either deleted or reassigned to a replacement employee.

Any special conditions to the termination (e.g., denial of the right to use certain information) should be reviewed with the departing employee; in addition, a document stating these conditions should be signed by the employee. All terminations should be routed through the computer security representative for the facility where the terminated employee works to ensure that all information system access authority has been revoked.

Security Policies and Procedures

Appropriate policies and procedures are key to the establishment of an effective information security program. Policies and procedures should reflect the general policies of the organization as regards the protection of information and computing resources. Policies should cover the use of computing resources, marking of sensitive information, movement of computing resources outside the facility, introduction of personal computing equipment and media into the facility, disposal of sensitive waste, and computer and data security incident reporting. Enforcement of these policies is essential to their effectiveness.

Supervision

Often, an alert supervisor is the first person to notice a change in an employee's attitude. Early signs of job dissatisfaction or personal distress should prompt supervisors to consider subtly moving the employee out of a critical or sensitive position.

Supervisors must be thoroughly familiar with the policies and procedures related to the responsibilities of their department. Supervisors should require that their staff members comply with pertinent policies and procedures and should observe the effectiveness of these guidelines. If the objectives of the policies and procedures can be accomplished more effectively, the supervisor should recommend appropriate improvements. Job assignments should be reviewed regularly to ensure that an appropriate separation of duties is maintained, that employees in sensitive positions are occasionally removed from a complete processing cycle without prior announcement, and that critical or sensitive jobs are rotated periodically among qualified personnel.

Disaster Recovery, Contingency, and Emergency Plans

The disaster recovery plan is a document containing procedures for emergency response, extended backup operations, and recovery should a computer installation experience a partial or total loss of computing resources or physical facilities (or of access to such facilities). The primary objective of this plan, used in conjunction with the contingency plans, is to provide reasonable assurance that a computing installation can recover from disasters, continue to process critical applications in a degraded mode, and return to a normal mode of operation within a reasonable time. A key part of disaster recovery planning is to provide for processing at an alternative site during the time that the original facility is unavailable.

Contingency and emergency plans establish recovery procedures that address specific threats. These plans help prevent minor incidents from escalating into disasters. For example, a contingency plan might

provide a set of procedures that defines the condition and response required to return a computing capability to nominal operation; an emergency plan might be a specific procedure for shutting down equipment in the event of a fire or for evacuating a facility in the event of an earthquake.

User Registration for Computer Access

Formal user registration ensures that all users are properly authorized for system and service access. In addition, it provides the opportunity to acquaint users with their responsibilities for the security of computing resources and to obtain their agreement to comply with related policies and procedures.

Detective Administrative Controls

Detective administrative controls are used to determine how well security policies and procedures are complied with, to detect fraud, and to avoid employing persons that represent an unacceptable security risk. This type of control includes:

- Security reviews and audits
- Performance evaluations
- Required vacations
- Background investigations
- Rotation of duties

Security Reviews and Audits

Reviews and audits can identify instances in which policies and procedures are not being followed satisfactorily. Management involvement in correcting deficiencies can be a significant factor in obtaining user support for the computer security program.

Performance Evaluations

Regularly conducted performance evaluations are an important element in encouraging quality performance. In addition, they can be an effective forum for reinforcing management's support of information security principles.

Required Vacations

Tense employees are more likely to have accidents or make errors and omissions while performing their duties. Vacations contribute to the health of employees by relieving the tensions and anxieties that typically develop from long periods of work. In addition, if all employees in critical or sensitive positions are forced to take vacations, there will be less opportunity for an employee to set up a fraudulent scheme that depends on the employee's presence (e.g., to maintain the fraud's continuity or secrecy). Even if the employee's presence is not necessary to the scheme, required vacations can be a deterrent to embezzlement because the employee may fear discovery during his or her absence.

Background Investigations

Background investigations may disclose past performances that might indicate the potential risks of future performance. Background investigations should be conducted on all employees being considered for promotion or transfer into a position of trust; such investigations should be completed before the employee is actually placed in a sensitive position. Job applicants being considered for sensitive positions should also be investigated for potential problems. Companies involved in government-classified projects should conduct these investigations while obtaining the required security clearance for the employee.

Rotation of Duties

Like required vacations, rotation of duties (i.e., moving employees from one job to another at random intervals) helps deter fraud. An additional benefit is that as a result of rotating duties, employees are cross-trained to perform each other's functions in case of illness, vacation, or termination.

PHYSICAL CONTROLS

Preventive

- Backup files and documentation
- Fences
- Security guards
- Badge systems
- Locks and keys
- Backup power
- Biometric access controls
- Site selection
- Fire extinguishers

Detective

- Motion detectors
- Smoke and fire detectors
- Closed-circuit television monitoring
- Sensors and alarms

TECHNICAL CONTROLS

Preventive

- Access control software
- Anti-virus software
- Library control systems
- Passwords
- Smart cards
- Encryption
- Dial-up access control and callback systems

Detective

- Audit trails
- Intrusion-detection expert systems

ADMINISTRATIVE CONTROLS

Preventive

- Security awareness and technical training
- Separation of duties
- Procedures for recruiting and terminating employees
- Security policies and procedures
- Supervision
- Disaster recovery and contingency plans
- User registration for computer access

Detective

- Security reviews and audits
- Performance evaluations
- Required vacations
- Background investigations
- Rotation of duties

EXHIBIT 162.1 Information security controls.

Summary

Information security controls can be classified as physical, technical, or administrative. These are further divided into preventive and detective controls. Exhibit 162.1 lists the controls discussed in this chapter.

The organization's security policy should be reviewed to determine the confidentiality, integrity, and availability needs of the organization. The appropriate physical, technical, and administrative controls can then be selected to provide the required level of information protection, as stated in the security policy.

A careful balance between preventive and detective control measures is needed to ensure that users consider the security controls reasonable and to ensure that the controls do not overly inhibit productivity. The combination of physical, technical, and administrative controls best suited for a specific computing environment can be identified by completing a quantitative risk analysis. Because this is usually an expensive, tedious, and subjective process, however, an alternative approach — referred to as meeting the standard of due care — is often used. Controls that meet a standard of due care are those that would be considered prudent by most organizations in similar circumstances or environments. Controls that meet the standard of due care generally are readily available for a reasonable cost and support the security policy of the organization; they include, at the least, controls that provide individual accountability, auditability, and separation of duties.

Physical Security

Tom Peltier

Before any controls can be implemented into the workplace, it is necessary to assess the current level of security. This can be accomplished in a number of ways. The easiest one is a "walk-about." After hours, walk through the facility and check for five key controls:

1. Office doors are locked.
2. Desks and cabinets are locked.
3. Workstations are secured.
4. Diskettes are secured.
5. Company information is secured.

Checking for these five key control elements will give you a basic understanding of the level of controls already in place and a benchmark for measuring improvements once a security control system is implemented. Typically, this review will nearly show a 90% control deficiency rate. A second review is recommended six to nine months after the new security controls are in place.

This chapter examines two key elements of basic computer security: physical security and biometrics. Physical security protects your organization's physical computer facilities. It includes access to the building, to the computer room(s), to the computers (mainframe, mini, and micros), to the magnetic media, and to other media. Biometrics devices record physical traits (i.e., fingerprint, palm print, facial features, etc.) or behavioral traits (signature, typing habits, etc.).

A BRIEF HISTORY

In the beginning of the computer age, it was easy to protect the systems; they were locked away in a lab and only a select few "wizards" were granted access. Today, computers are cheaper, smaller, and more accessible to almost everyone.

During the mid-twentieth century, the worldwide market for mainframe computer systems exploded. As the third-generation systems became available in the 1960s, companies began to understand their dependence on these systems. By the mid to late 1970s, the security industry began to

catch up: with Halon fire suppression systems, card access, and RACF and ACF2. In the final quarter of the century, mainframe-centered computing was at its zenith.

By 1983, the affordable portable computer began to change the working landscape for information security professionals. An exodus from the mainframe to the desktop began. The controls that had been so hard won in the previous two decades were now considered the cause of much bureaucracy. Physical security is now needed in desktops. For years, conventional thinking was that a computer is a computer is a computer is a computer. Controls are even more important in the desktop or workstation environment than in the mainframe environment.

The computing environment is now moving from the desktop to the user. With the acceptance of telecommuting, the next challenge will be to apply physical security solutions to the user-centered computing environment.

With computers on every desk connected via networks to other local and remote systems, physical security needs must be reviewed and upgraded wherever necessary. Advances in computer and communications security are not enough; physical security remains a vitally important component of an overall information security plan.

WHERE TO FOCUS ATTENTION

Before implementing any form of physical security, it may be helpful to conduct a limited business impact analysis (BIA) to focus on existing threats to the computer systems and determine where resources can best be spent. It is very important to consider all potential threats, even unlikely ones. Ignore those with a zero likelihood, such as a tsunami in Phoenix or a sandstorm in Maui. A very simple BIA could be diagrammed as shown in Exhibit 1.

An unlimited number of threats can be of concern to your organization. Any number of high-likelihood threats can be identified. First consider those threats that might actually affect your organization (e.g., fire, flood, or fraud). Three elements are generally associated with each threat:

- The agent: the destructive agent can be a human, a machine, or nature.
- The motive: the only agent that can threaten accidentally and intentionally is the human.
- The results: for the information systems community, this would be a loss of access or unauthorized access, modification, or disclosure or destruction of data or information.

TYPE OF THREAT	Probability	Human Impact	Property Impact	Business Impact	Internal Resource	External Resource	TOTAL
	4 ←				→	1	
Fire	3	3	4	4	2	2	16

Exhibit 1. Business Impact Analysis Example.

Note: Rank each impact based on 4 = high to 1 = low. Rank each resource based on 4 = weak resources available to 1 = strong resources available.

The focus of physical security has often been on human-made disasters, such as sabotage, hacking, and human error. Don't forget that the same kinds of threats can also occur from natural disasters.

NATURAL DISASTERS AND CONTROLS

Fire — A conflagration affects information systems through heat, smoke, or suppression agent (e.g., fire extinguishers and water) damage. This threat category can be minor, major, or catastrophic. *Controls:* install smoke detectors near equipment; keep fire extinguishers near equipment and train employees in their proper use; conduct regular fire evacuation exercises.

Environmental failure — This type of disaster includes any interruption in the supply of controlled environmental support provided to the operations center. Environmental controls include clean air, air conditioning, humidity, and water. *Controls:* since humans and computers don't coexist well, try to keep them separate. Many companies are establishing command centers for employees and a "lights-out" environment for the machines. Keep all rooms containing computers at reasonable temperatures (60 to 75°F or 10 to 25°C). Keep humidity levels at 20 to 70% and monitor environmental settings.

Earthquake — A violent ground motion results from stresses and movements of the earth's surface. *Controls:* keep computer systems away from

glass and elevated surfaces; in high-risk areas secure the computers with antivibration devices.

Liquid Leakage — A liquid inundation includes burst or leaking pipes and accidental discharge of sprinklers. *Controls:* keep liquid-proof covers near the equipment and install water detectors on the structural floor near the computer systems.

Lightning — An electrical charge of air can cause either direct lightning strikes to the facility or surges due to strikes to electrical power transmission lines, transformers, and substations. *Controls:* install surge suppressors, store backups in grounded storage media, install and test Uninterruptible Power Supply (UPS) and diesel generators.

Electrical Interruption — A disruption in the electrical power supply, usually lasting longer than one-half hour, can have serious business impact. *Controls:* install and test UPS, install line filters to control voltage spikes, and install antistatic carpeting.

THE HUMAN FACTOR

Recent FBI statistics indicate that 72% of all thefts, fraud, sabotage, and accidents are caused by a company's own employees. Another 15 to 20% comes from contractors and consultants who are given access to buildings, systems, and information. Only about 5 to 8% is done by external people, yet the press and management focus mostly on them. The typical computer criminal is a nontechnical authorized user of the system who has been around long enough to locate the control deficiencies.

When implementing control devices, make certain that the controls meet the organization's needs. Include a review of internal access, and be certain that employees meet the standards of due care imposed on external sources. "Intruders" can include anybody who is not authorized to enter a building, system, or data.

The first defense against instruders is to keep them out of the building or computer room. However, because of cost-cutting measures in the past two decades, very few computer facilities are guarded anymore. With computers everywhere, determining where to install locks is a significant problem.

To gain access to any business environment, everybody should have to pass an authentication and/or authorization test. The three ways of authenticating users involve something:

- That the user knows (a password).
- That the user has (a badge, key, card, or token).
- Of their physiognomy (fingerprint, retinal image, voice).

LOCKS

In addition to securing the campus, it may be necessary to secure the computers, networks, disk drives, and electronic media. One method of securing a workstation is with an anchor pad, a metal pad with locking rods secured to the surface of the workstation. The mechanism is installed to the shell of the computer. These are available from many vendors.

Many organizations use cables and locks. Security cables are multi-strand, aircraft-type steel cables affixed to the workstation with a permanently attached plate that anchors the security cable to the desk or other fixture.

Disk locks are another way to secure the workstation. These small devices are quickly inserted into the diskette slot and lock out any other diskette from the unit. They can prevent unauthorized booting from diskettes and infection from viruses.

Cryptographic locks also prevent unauthorized access by rendering information unreadable to unauthorized personnel. Encryption software does not impact day-to-day operations while ensuring the confidentiality of sensitive business information. Crypographic locks are cost-effective and easily available.

TOKENS

As human security forces shrink, there is more need to ensure that only authorized personnel can get into the computer room. A token is an object the user carries to authenticate his or her identity. These devices can be token cards, card readers, or biometric devices. They have the same purpose: to validate the user to the system. The most prevalent form is the card, an electric device that normally contains encoded information about the individual who is authorized to carry it. Tokens are typically used with another type of authentication. Many cipher locks have been replaced with token card access systems.

Challenge-Response Tokens

Challenge-response tokens supply passcodes that are generated using a challenge from the process requesting authentication (such as the Security Dynamics' SecurID). Users enter their assigned user IDs and passwords plus a password supplied by the token card. This process requires that the user supply something they possess (the token) and something that they know (the challenge/response process). This process makes passcode sniffing and brute force attacks futile.

Challenge-response is an asynchronous process. An alternative to challenge-response is the synchronous token that generates the password without the input of a challenge from the system. It is synchronized with

the authenticating computer when the user and token combination is registered on the system.

Dumb Cards

For many years, photo identification badges have sufficed as a credential for most people. With drivers' licenses, passports, and employee ID badges, the picture — along with the individual's statistics — supplies enough information for the authentication process to be completed. Most people flash the badge to the security guard or give a license to a bank teller. Someone visually matches the ID holder's face to the information on the card.

Smart Cards

The automatic teller machine (ATM) card is an improvement on the "dumb card"; these "smart" cards require the user to enter a personal ID number (PIN) along with the card to gain access. The ATM compares the information encoded on the magnetic stripe with the information entered at the ATM machine.

The smart card contains microchips that consist of a processor, memory used to store programs and data, and some kind of user interface. Sensitive information is kept in a secret read-only area in its memory, which is encoded during manufacturing and is inaccessible to the card's owner. Typically, these cards use some form of cryptography that protects the information. Not all smart cards work with card readers. A user inserts the card into the reader, the system displays a message, and if there is a match, then the user is granted access.

Types of Access Cards

Access cards employ different types of technology to ensure authenticity:

- Photo ID cards contain a photograph of the user's face and are checked visually.
- Optical-coded cards contain tiny, photographically etched or laser-burned dots representing binary zeros and ones that contain the individual's encoded ID number. The card's protective lamination cannot be removed without destroying the data and invalidating the card.
- Electric circuit cards contain a printed circuit pattern. When inserted into a reader, the card closes certain electrical circuits.
- Magnetic cards, the most common form of access control card, contain magnetic particles that contain, in encoded form, the user's permanent ID number. Data can be encoded on the card, but the tape itself cannot be altered or copied.
- Metallic stripe cards contain rows of copper strips. The presence or absence of strips determines the code.

BIOMETRIC DEVICES

Every person has unique physiological, behavioral, and morphological characteristics that can be examined and quantified. Biometrics is the use of these characteristics to provide positive personal identification. Fingerprints and signatures have been used for years to prove an individual's identity, but individuals can be identified in many other ways. Computerized biometrics identification systems examine a particular trait and use that information to decide whether the user may enter a building, unlock a computer, or access system information.

Biometric devices use some type of data input device, such as a video camera, retinal scanner, or microphone, to collect information that is unique to the individual. A digitized representation of a user's biometric characteristic (fingerprint, voice, etc.) is used in the authentication process. This type of authentication is virtually spoof-proof and is never misplaced. The data are relatively static but not necessarily secret. The advantage of this authentication process is that it provides the correct data to the input devices.

Fingerprint Scan

The individual places a finger in or on a reader that scans the finger, digitizes the fingerprint, and compares it against a stored fingerprint image in the file. This method can be used to verify the identity of individuals or compare information against a data base covering many individuals for recognition. Performance:

- False rejection rate = 9.4%
- False acceptance rate = 0
- Average processing time = 7 seconds

Retinal Scan

This device requires that the user look into an eyepiece that laser-scans the pattern of the blood vessels. The patterns are compared to provide positive identification. It costs about $2,650. Performance:

- False rejection rate = 1.5%
- False acceptance rate = 1.5%
- Average processing time = 7 seconds

Palm Scan

The system scans 10,000 points of information from a 2-inch-square area of the human palm. With the information, the system identifies the person as an impostor or authentic. The typical price is $2,500. Performance:

- False rejection rate = 0
- False acceptance rate = 0.00025%
- Average processing time = 2-3 seconds

Hand Geometry

This device uses three-dimensional hand geometry measurements to provide identification. The typical price is $2,150. Performance:

- False rejection rate = 0.1%
- False acceptance rate = 0.1%
- Average processing time = 2 to 3 seconds

Facial Recognition

Using a camera mounted at the authentication place (gate, monitor, etc.) the device compares the image of the person seeking entry with the stored image of the authorized user indexed to the system. The typical price is $2,500. Performance:

- Average processing time = 2 seconds

Voice Verification

When a person speaks a specified phrase into a microphone, this device analyzes the voice pattern and compares it against a stored data base. The price can run as high as $12,000 for 3,000 users. Performance:

- False rejection rate = 8.2%
- False acceptance rate = 0.4%
- Average processing time = 2 to 3 seconds (response time is calculated after the password or phrase is actually spoken into the voice verification system).

TESTING

Security systems, passwords, locks, token cards, biometrics, and other authentication devices are expected to function accurately from the moment they are installed, but it is the management and testing that makes them work. There is little point in installing an elaborate access control system for the computer room if the employees routinely use the emergency fire exits. Employees must be trained in the proper use of physical security systems. Access logs must be monitored and reconciled in a timely manner.

Training and awareness demands time, money, and personnel, but it is essential for organizations to meet the challenges brought about by increased competition and reduced resources. There must be a partnership between the technology and the employees. Exhibit on spending at

least as much time and resources on training employees on how to use the technology as on procuring and installing it. Employees must understand why the control mechanisms were selected and what their roles are in the security process.

SUMMARY

Companies where employees hold open the door for others to walk through may need to review their level of security awareness. The first step in implementing a physical security program is determining the level of need and the current level of awareness. To implement a cost-effective security program (1) analyze the problems, (2) design or procure controls, (3) implement those controls, (4) test and exercise those controls, and (5) monitor the controls. Implement only controls needed to meet the current needs, but make sure that additional control can be added later if required. Physical security is an organization's first line of defense against theft, sabotage, and natural disasters.

Recommended Readings

Russell, D. and Gangemi, G.T., *Computer Security Basics,* O' Reilly & Associates, Inc., Sebastopol, CA, 1991.

Jackson, K. and Hruska, J., *Computer Security Reference Book,* CRC Press, Inc., Boca Raton, FL, 1992.

Ashborn, J., "Baubles, Bangles and Biometrics," Association for Biometrics (1995).

Davies, S. G., "Touching Big Brother: How biometric technology will fuse flesh and machine," *Information Technology & People,* Vol. 7, No. 4, 1994.

Lawrence, S. et al., "Face Recognition: A hybrid neural network approach," Technical Report UMIACS-TR-96 and CS-TR-3608, Institute for Advanced Computer Studies, University of Maryland, College Park, MD, 1996.

163

Physical Security: The Threat after September 11, 2001

Jaymes Williams, CISSP

The day that changed everything began for me at 5:50 a.m. I woke up and turned on the television to watch some news. This was early Tuesday morning, September 11, 2001. My local news station had just interrupted its regular broadcast and switched over to CNN, so right away I knew something important had happened. I learned an airliner had crashed into one of the towers of the World Trade Center in New York.

In disbelief, I made my way to the kitchen and poured myself a cup of coffee. I returned to the television and listened to journalists and airline experts debate the likely cause of this event. I thought to myself, "there isn't a cloud in the sky; how could an aircraft accidentally hit such a large structure?" Knowing, but not wanting to accept the answer, I listened while hoping the television would give me a better one.

While waiting for the answer that never came, I noticed an aircraft come from the right side of the screen. It appeared to be going behind the towers of the Trade Center, or perhaps I was only hoping it would. This was one of those instances where time appeared to dramatically slow down. In the split second it took to realize the plane should have already come out from behind the towers, the fireball burst out the side of the tower instead. It was now undeniable. This was no accident.

Later, after getting another cup of coffee, I returned to the television to see only smoke; the kind of smoke you only see when a building is imploded to make way for new construction. To my horror, I knew a tower had collapsed. Then, while the journalists were recovering from the shock and trying to maintain their on-air composure, they showed the top of the remaining tower. For some reason, it appeared that the camera had started to pan up. I started to feel a bit of vertigo. Then, once again, a horrible realization struck. The camera was not going up; the building was going down. Within the span of minutes, the World Trade Center was no more; and Manhattan was totally obscured by smoke. I was in total disbelief. This had to be a movie; but it was not. The mind's self-defenses take over when things occur that it cannot fathom, and I felt completely numb. I had witnessed the deaths of untold thousands of people on live TV. Although I live 3000 miles away, it might as well have happened down the street. The impact was the same. Then the news of the crash at the Pentagon came, followed by the crash of the aircraft in Pennsylvania.

I tried to compose myself to go to work, although work seemed quite unimportant at the moment. Somehow, I put myself together and made my way out the door. On the way to work, I thought to myself that this must be the Pearl Harbor of my generation. And, I realized, my country was probably at war — but with whom?

The preceding is my recollection of the morning of September 11. This day has since become one of those days in history where we all remember where we were and what we were doing. Although we all have our own individual experiences from that horrible day, some people more affected than others, these individual experiences all form a collective experience that surprised and shocked us all.

Security practitioners around the world, and especially in the United States, have to ask themselves some questions. Can this happen here? Is my organization a potential target? Now that a War on Terrorism has begun as a result of the September 11 attacks, the answer to both of these questions, unfortunately, is "yes." However, there are some things that can be done to lessen the risk. This chapter examines why the risk of terrorism has increased, what types of organizations or facilities are at higher risk, and what can be done to lessen that risk.

Why Is America a Target?

> Just because you're not paranoid doesn't mean they're not out to get you!
>
> **— From the U.S. Air Force Special Operations Creed**

There are many reasons terrorist groups target America. One reason is ideological differences. There are nations or cultures that do not appreciate the freedom and tolerance espoused by Americans. America is inarguably the world's leading industrial power and capitalist state. There are people in the world who may view America as a robber baron nation and hate Americans because of our perceived wealth. Another reason is religious differences. There are religiously motivated groups that may despise America and the West because of perceived nonconformance with their religious values and faith. A further reason is the perception that the U.S. government has too much influence over the actions of other governments. Terrorists may think that, through acts of terror, the U.S. government will negotiate and ultimately comply with their demands. However, our government has repeatedly stated it will not negotiate with terrorists.

A final reason is that Americans are perceived as easy targets. The "open society" in America and many Western countries makes for easy movement and activities by terrorists. Whether performing in charitable organizations, businesses, in governmental capacities, or as tourists, Americans are all over the world. This makes targeting Americans quite easy for even relatively poorly trained terrorist groups. U.S. military forces stationed around the world are seen as visible symbols of U.S. power and, as such, are also appealing targets to terrorists.

Why be Concerned?

Terrorism can be defined as the calculated use of violence, or threat of violence, to inculcate fear; intended to coerce or intimidate governments or societies in the pursuit of goals that are generally political, religious, or ideological. Some examples of terrorist objectives and tactics can be seen in Exhibit 163.1.

The increased threat of terrorism and cyber-terrorism is a new and important consideration for information security practitioners. Previously, physical security threats included such things as unauthorized access, crime, environmental conditions, inclement weather, earthquakes, etc. The events of September 11 have shown us exactly how vulnerable we are. One of the most important lessons we security practitioners can take from that day is to recognize the need to reevaluate our physical security practices to include terrorism. Adding terrorism to the mix necessitates some fundamental changes in the way we view traditional physical security. These changes need to include protective measures from terrorism.

Depending on the type of organization, it is quite possible that terrorists may target it. Whether they target facilities or offices for physical destruction or they select an organization for a cyber-strike, prudent information security practitioners will assume they have been targeted and plan accordingly.

Is Your Organization a Potential Target?

Many organizations may be potential targets of terrorists and have no idea they are even vulnerable. Government agencies, including federal, state, and local, and infrastructure companies may be primary targets. Other vulnerable organizations may be large multinational companies that market American products around the world and organizations located in well-known skyscrapers. Specific examples of these types of potential targets

EXHIBIT 163.1 Terrorist Objectives and Tactics

Examples of Terrorist Objectives
Attract publicity for the group's cause
Demonstrate the group's power
Show the existing government's lack of power
Extract revenge
Obtain logistic support
Cause a government to overreact
Common Terrorist Tactics
Assassination
Arson
Bombing
Hostage taking
Kidnapping
Hijacking or skyjacking
Seizure
Raids or attacks on facilities
Sabotage
Hoaxes
Use of special weapons
Environmental destruction
Use of technology

will not be named to avoid the possibility of placing them at higher risk. See Exhibit 163.2 for different types of potential targets.

Government Agencies

There are many terrorists who hate the U.S. government and those of many Western countries. In the minds of terrorists and their sympathizers, governments create the policies and represent the values with which they vehemently disagree. It does not take a rocket scientist, or an information security practitioner for that matter, to realize that agencies of the U.S. government are prime targets for terrorists. This, of course, also includes the U.S. military. Other Western countries, especially those supporting the United States in the War on Terrorism, may also find themselves targets of terrorists. State and local governments may also be at risk.

- *Infrastructure companies.* Companies that comprise the infrastructure also face an increased risk of terrorism. Not only may terrorists want to hurt the U.S. and Western governments, but they may also want to disrupt normal life and the economies of the Western world. Disrupting the flow of energy, travel, finance, and information is one such way to accomplish this. The medical sector is also included here. One has to now consider the previously unthinkable, look beyond our usual mindsets, and recognize that, because medical facilities have not previously been targeted, it is conceivable they could be targeted in the future.
- *Location-based targets.* There are also those targets that by their location or function are at risk. Just as the towers of the World Trade Center represented the power of the American economy to the September 11 terrorists, other landmarks can be interpreted as representing things uniquely American to those with hostile intent. Such landmarks can include skyscrapers in major cities or any of the various landmarks that represent American or Western interests. Popular tourist destinations or events with large numbers of people in attendance can also be at risk because they are either uniquely American/Western or simply because they are heavily populated.
- *Things that mean America.* There is another category to consider. This category has some overlap with the above categories but still deserves mention. Large corporations that represent America or the West to the rest of the world can also be targeted. This also includes companies whose products are sold around the world and represent America to the people of the world.

EXHIBIT 163.2 Potential Terrorist Targets

Government Agencies
U.S. federal agencies
U.S. military facilities
State governments
County governments
Local governments
Infrastructure
Energy
Transportation
Financial
Water
Internet
Medical
Location Based
Tall office buildings
National landmarks
Popular tourist destinations
Large events
Associated with America
Large corporations synonymous with the Western world
American or *U.S.* in the name
Companies that produce famous American brand products

If an organization falls into one of the above categories, it may face a greater risk from terrorism than previously thought. If an organization does not fit one of the above categories, information security practitioners are still well-advised to take as many antiterrorism precautions as feasible.

Paradigm Shift: Deterrence to Prevention

> Business more than any other occupation is a continual dealing with the future; it is a continual calculation, an instinctive exercise in foresight.
>
> **— Henry R. Luce**

The operating paradigm of physical security has been deterrence. The idea of a perpetrator not wanting to be caught, arrested, or even killed has become so ingrained in the way we think that we take it for granted. As we probably all know by now, there are people motivated by fervent religious beliefs or political causes that do not share this perspective; they may be willing or even desiring to die to commit an act they believe will further their cause.

Most security protections considered industry standard today are based on the deterrence paradigm. Security devices such as cameras, alarms, x-ray, or infrared detection are all used with the intent to deter a perpetrator who does not want to be caught. Although deterrence-based measures will provide adequate security for the overwhelming majority of physical security threats, these measures may be largely ineffective against someone who plans to die committing an act of terrorism.

On the morning of September 11, 2001, we learned a painful lesson: that deterrence does not deter those who are willing to die to perpetrate whatever act they have in mind. Unfortunately, this makes physical security much more difficult and expensive. Information security practitioners need to realize that commonly accepted standards such as having security cameras, cipher-lock doors, and ID badges may only slow down a potential terrorist. Instead of working to deter intruders, we now have to also consider the previously unconsidered — the suicidal terrorist. This means considering what measures it will take to prevent someone who is willing to die to commit a terrorist act.

The airline industry appears to have learned that much more stringent security measures are required to prevent a recurrence of what happened on September 11. Previously, an airline's worst nightmare was either a bombing of an aircraft or a hijacking followed by tense negotiations to release hostage passengers. No one had considered the threat of using an airliner as a weapon of mass destruction. Anyone who has flown since then is familiar with the additional delays, searches, and ID checks. They are inconvenient and slow down the traveler; however, this is a small price to pay for having better security.

Although there is still much more to be done, this serves as an example of using the prevention paradigm. The airlines have taken many security measures to prevent another such occurrence. Unfortunately, as with information security, there is no such thing as absolute physical security. There is always the possibility that something not previously considered will occur. Information security practitioners will also likely have to work within corporate/governmental budget constraints, risk assessments, etc. that may limit their ability to implement the needed physical security changes.

Reducing The Risk of Terrorism

> The determination of these terrorists will not deter the determination of the American people. We are survivors and freedom is a survivor.
>
> **— Attorney General John Ashcroft**
>
> **Press conference on September 11, 2001**

Now that we have a better understanding of why we face a greater risk of terrorism and who may be a target, the issue becomes how to better protect our organizations and our fellow employees. There are many methods to reduce the risk of terrorism. These methods include reviewing and increasing the physical security of an organization using the previously discussed prevention paradigm; controlling sensitive information through operational security; developing terrorism incident handling procedures; and building security procedures and antiterrorism procedures for employees. Several of these methods rely on employee training and periodic drills to be successful.

Physical Security Assessments

The first step in reducing risk is to control the physical environment. In this section we use the term *standard* to imply industry-standard practices for physical security. The term *enhanced* will refer to enhanced procedures that incorporate the prevention paradigm.

Verify Standard Physical Security Practices Are in Place

Conduct a standard physical security assessment and implement changes as required. It is important to have physical security practices at least at current standards. Doing this will also minimize the risk from most standard physical security threats. As the trend toward holding organizations liable continues to emerge in information security, it is also likely to occur with physical security in the foreseeable future.

Conduct an Enhanced Physical Security Assessment

Once the standard physical security is in place, conduct another assessment that is much more stringent. This assessment should include enhanced physical security methods. Unfortunately, there is not yet a set of industry standards to protect against the enhanced threat. Many excellent resources are available from the U.S. government. Although they are designed for protecting military or other government facilities, many of these standards can also be successfully implemented in the private sector. At this point, information security practitioners are essentially left to their own initiative to implement standards. Perhaps, in the near future, a set of standards will be developed that include the enhanced threat.

Currently, there are many excellent resources available on the Internet from the U.S. government. However, at the time of this writing, the U.S. government is becoming more selective about what information is available to the public via the Internet for security reasons. It is quite possible that these resources may disappear from the Internet at some point in the near future. Information security practitioners may wish to locate these valuable resources before they disappear. A listing of Internet resources can be found in Exhibit 163.3.

EXHIBIT 163.3 Internet Resources

Professional Organizations
DRI International — http://www.drii.org
International Security Management Association — http://www.ismanet.com
The Terrorism Research Center — http://www.terrorism.com/index.shtml
Infosyssec.com's physical security resource listing — http://www.infosyssec.com/infosyssec/physfac1.htm
Infosyssec.com's Business Continuity Planning Resource Listing — http://www.infosyssec.net/infosyssec/buscon1.htm

Government Agencies
National Infrastructure Protection Center (NIPC) — http://www.nipc.gov
Federal Bureau of Investigation (FBI) — http://www.fbi.gov
Office of Homeland Security Critical Infrastructure Assurance Office (CIAO) — http://www.ciao.gov
Office of Homeland Security — http://www.whitehouse.gov/homeland/
FBI's "War on Terrorism" page — http://www.fbi.gov/terrorinfo/terrorism.htm
Canadian Security Intelligence Service (CSIS) Fighting Terrorism Page — http://canada.gc.ca/wire/2001/09/110901-US_e.html
Bureau of Alcohol, Tobacco & Firearms Bomb Threat Checklist — http://www.atf.treas.gov/explarson/information/bombthreat/checklist.htm

Military Agencies
Department of Defense — http://www.defenselink.mil/
Department of Defense's "Defend America" site — http://www.defendamerica.mil/
U.S. Army Physical Security Field Manual — http://www.adtdl.army.mil/cgi-bin/atdl.dll /fm/3-19.30/toc.htm

Implement Recommended Changes

Again, because there is no uniform set of standards for enhanced physical security for the private sector, we are left to our own devices for enhancing our physical security. Because we are not likely to have unlimited budgets for improving physical security, information security practitioners will have to assess the risk for their organizations, including the potential threat of terrorism, and make recommended changes based on the assessed risk. Ideally, these changes should be implemented in the most expeditious manner possible.

Controlling Sensitive Information through Operational Security (OPSec)

We have now successfully "circled the wagons" and improved physical access controls to our facilities. The next step is to better control our sensitive information. As illustrated by the famous World War II security poster depicted in Exhibit 163.4, the successful control of information can win or lose wars. The Allied capture of the Enigma encryption device proved a critical blow to the Germans during World War II. The Allies were then able to decipher critical codes, which gave them an insurmountable advantage. Again, during the Gulf War, the vast technical advantage enjoyed by the Allied Coalition gave them information supremacy that translated into air supremacy.

These lessons of history illustrate the importance of keeping sensitive information out of the hands of those who wish to do harm. In the days since September 11, this means keeping sensitive information from all who do not need access. First, we need to define exactly what information is sensitive. Then we need to determine how to best control the sensitive information.

- *Defining sensitive information.* Sensitive information can easily be defined as information that, if available to an unauthorized party, can disclose vulnerabilities or can be combined with other information to be used against an organization. For example, seemingly innocuous information on a public Web site can provide a hostile party with enough information to target that organization. Information such as addresses of facilities, maps to facilities, officer and employee names, and names and addresses of customers or clients can all be combined to build a roadmap. This roadmap can tell the potential terrorist not only where the organization is and what it does, but also who is part of the organization and where it is vulnerable.

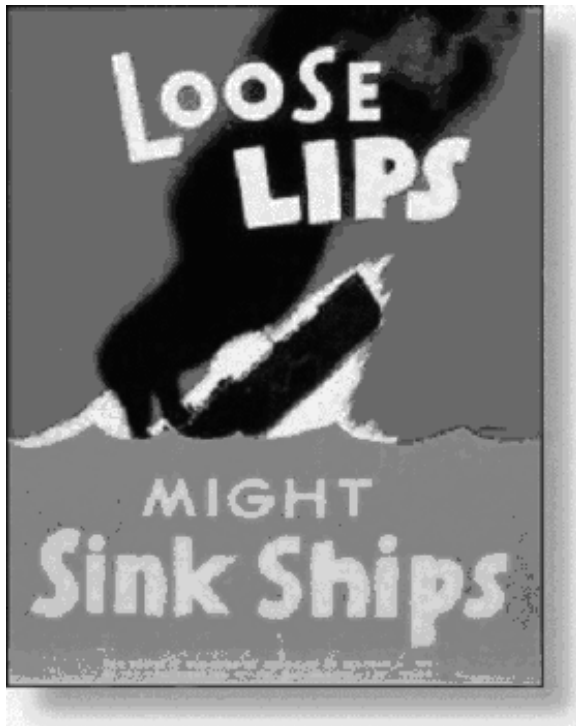


EXHIBIT 163.4 Famous World War II security poster.

- *Controlling sensitive information.* Prudent information security practitioners will first want to control the information source that leaves them the most vulnerable. There are several methods security practitioners can use to maintain control of their sensitive information: removing sensitive information from Web sites and corporate communications; destroying trash with sensitive information; having a clean desk policy; and limiting contractor/vendor access to sensitive information.
  - *Remove sensitive information from publicly available Web sites.* Removing physical addresses, maps, officer/employee names, etc. from these Web sites is highly advisable. They can either be removed entirely from the site or moved into a secured section of the site where access to this information is verified and logged.
  - On January 17, 2002, the National Infrastructure Protection Center released NIPC Advisory 02-001: Internet Content Advisory: Considering the Unintended Audience. See Exhibit 163.5 for a reprint of the advisory. This advisory can function as a set of standards for deciding what and what not to place on publicly available Internet sites. When bringing up the issue with management of removing information from Web sites, the information security practitioner may receive a response that echoes item number seven in the advisory: "Because the information is publicly available in many places, it is not worth an effort to remove it from our site." Although the information does exist elsewhere, the most likely and easiest place for terrorists to find it is on the target organization's Web site. This is also probably the first place they will look. Responsible information security practitioners, or corporate officers for that matter, should make it as difficult as possible for those with hostile intent to gain useful information from their Internet site.
  - *Remove sensitive information from all corporate communications.* No corporate communications should contain any sensitive information. If an organization already has an information clas-

EXHIBIT 163.5 NIPC Advisory 02-001

Internet Content Advisory: Considering the Unintended Audience

January 17, 2002

As worldwide usage of the Internet has increased, so too have the vast resources available to anyone online. Among the information available to Internet users are details on critical infrastructures, emergency response plans and other data of potential use to persons with criminal intent. Search engines and similar technologies have made arcane and seemingly isolated information quickly and easily retrievable by anyone with access to the Internet. The National Infrastructure Protection Center (NIPC) has received reporting that infrastructure related information, available on the Internet, is being accessed from sites around the world. Although in and of itself this information is not significant, it highlights a potential vulnerability.

The NIPC is issuing this advisory to heighten community awareness of this potential problem and to encourage Internet content providers to review the data they make available online. A related information piece on "Terrorists and the Internet: Publicly Available Data should be Carefully Reviewed" was published in the NIPC's *Highlights* 11-01 on December 07, 2001, and is available at the NIPC web site http://www.nipc.gov/. Of course, the NIPC remains mindful that, when viewing information access from a security point of view, the advantages of posting certain information could outweigh the risks of doing so. For safety and security information that requires wide dissemination and for which the Internet remains the preferred means, security officers are encouraged to include in corporate security plans mechanisms for risk management and crisis response that pertain to malicious use of open source information.

When evaluating Internet content from a security perspective, some points to consider include:

1. Has the information been cleared and authorized for public release?
2. Does the information provide details concerning enterprise safety and security? Are there alternative means of delivering sensitive security information to the intended audience?
3. Is any personal data posted (such as biographical data, addresses, etc.)?
4. How could someone intent on causing harm misuse this information?
5. Could this information be dangerous if it were used in conjunction with other publicly available data?
6. Could someone use the information to target your personnel or resources?
7. Many archival sites exist on the Internet, and that information removed from an official site might nevertheless remain publicly available elsewhere.

The NIPC encourages the Internet community to apply common sense in deciding what to publish on the Internet. This advisory serves as a reminder to the community of how the events of September 11, 2001, have shed new light on our security considerations.

The NIPC encourages recipients of this advisory to report computer intrusions to their local FBI office http://www.fbi.gov/contact/fo/fo.htm or the NIPC, and to other appropriate authorities. Recipients may report incidents online at http://www.nipc.gov/ incident/cirr.htm, and can reach the NIPC Watch and Warning Unit at (202) 323-3205, 1-888-585-9078, or nipc.watch@fbi.gov

sification structure in place, this vulnerability should already be resolved. However, if there is no information classification structure in place, this is excellent justification for implementing such a program. And, with such a program, the need for marking documents also exists.

- *Shred/destroy trash with sensitive information.* Do you really know who goes through your trash? Do you know your janitorial staff? Dumpster diving is a widely practiced social engineering method. Shredding is an excellent way to avoid this vulnerability and is already widely practiced. Many organizations have either on-site shredders or bins to collect sensitive documents, which are later shredded by contracted shredding companies.
- *Create a clean desk policy.* Information left unattended on a desktop is a favorite of social engineers. It is easier than dumpster diving (cleaner, too!) and will likely yield better results. Although the definition of clean desk may vary, the intent of such a policy is to keep sensitive information from being left unattended on desktops.
- *Limit contractor/vendor access to sensitive information.* This is a standard physical security practice, but it deserves special mention within the OPSec category because it is fairly easy to implement controls on contractor/vendor access. Restricting access to proprietary information is also a good practice.

EXHIBIT 163.6 Safe Mail-Handling Checklist

Suspicious Packages or Mail
Suspicious characteristics to look for include:
- An unusual or unknown place of origin
- No return address
- An excessive amount of postage
- Abnormal or unusual size
- Oily stains on the package
- Wires or strings protruding from or attached to an item
- Incorrect spelling on the package label
- Differing return address and postmark
- Appearance of foreign style handwriting
- Peculiar odor (many explosives used by terrorists smell like shoe polish or almonds)
- Unusual heaviness or lightness
- Uneven balance or shape
- Springiness in the top, bottom, or sides

Never cut tape, strings, or other wrappings on a suspect package or immerse a suspected letter or package in water; either action could cause an explosive device to detonate
Never touch or move a suspicious package or letter
Report any suspicious packages or mail to security officials immediately

- *Verify identity of all building/office visitors.* Many large organizations and office buildings are verifying the identity of all visitors. Some organizations and buildings are checking identification for everyone who enters. This is an excellent practice because it greatly reduces the risk of unauthorized access.
- *Report unusual visitors or activity to law enforcement agencies (LEA).* Visitors behaving in a suspicious or unusual manner should be reported to building security, if possible, and then to law enforcement authorities. Quick reporting may prevent undesired activities.
- *Exercise safe mail handling procedures.* Mail-handling procedures became of greater importance during the anthrax scare in the autumn of 2001. See Exhibit 163.6 for a list of safe mail handling procedures.

Develop Terrorism Incident Handling Procedures

Security Working Group

Many organizations have established security working groups. These groups may be composed of management, information security practitioners, other security specialists, and safety and facilities management people. Members of the group can also serve as focal points for networking with local, state, and federal authorities and professional organizations to receive intelligence/threat information. The group may meet regularly to review the organization's security posture and act as a body for implementing upgraded security procedures. It may also conduct security evaluations.

Establish Terrorism Incident Procedures

Just as it is important to have incident response plans and procedures for computer security incidents, it is also highly advisable to have incident response plans and procedures for terrorist threats or incidents.

An integral part of any terrorism incident response is checklists for bomb threats and other terrorist threats. These checklists should contain numerous questions to ask the individual making the threatening call: where is the bomb, when is it going to go explode, what does it look like, etc. The checklists should also contain blanks to fill in descriptions of the caller's voice — foreign accent, male or female, tone of voice, background noise, etc. Checklists should be located near all phones or, at a minimum, in company telephone directories. Many federal and state agencies have such checklists available for the general public. The Bureau of Alcohol, Tobacco & Firearms has an excellent checklist that is used by many agencies and is shown in Exhibit 163.7.

Again, as with computer incident response teams, training is quite important. Employees need to know how to respond in these types of high-stress situations. Recurring training on how to respond to threatening phone calls and to complete the checklist all contribute to reduced risk.

EXHIBIT 163.7 BATF Bomb Threat Checklist

ATF BOMB THREAT CHECKLIST
Exact time of call:

_____

Exact words of caller:

_____

QUESTIONS TO ASK

1. When is bomb going to explode?
2. Where is the bomb?
3. What does it look like?
4. What kind of bomb is it?
5. What will cause it to explode?
6. Did you place the bomb?
7. Why?
8. Where are you calling from?
9. What is your address?
10. What is your name?

CALLER'S VOICE (circle)

Calm	Slow	Crying	Slurred
Stutter	Deep	Loud	Broken
Giggling	Accent	Angry	Rapid
Stressed	Nasal	Lisp	Excited
Disguised	Sincere	Squeaky	Normal

If voice is familiar, whom did it sound like?
Were there any background noises?
Remarks:
Person receiving call:
Telephone number call received at:
Date:
Report call immediately to:
(Refer to bomb incident plan)

Safety Practices

Here is an excellent opportunity to involve organizational safety personnel or committees. Some practices to involve them with are:

- *Review building evacuation procedures.* This will provide the current and best method for evacuating buildings should the need arise. Also plan for secondary evacuation routes in the event the primary route is unusable.
- *Conduct building evacuation drills.* Periodic building evacuation drills, such as fire drills, provide training and familiarity with escape routes. In an emergency, it is far better to respond with training. These should be conducted without prior notification on all shifts. Drills should not be the same every time. Periodically, vary the drill by blocking an escape route, forcing evacuees to alter their route.
- *Conduct terrorism event drills.* Other drills, such as responding to various terrorism scenarios, may be beneficial in providing the necessary training to respond quickly and safely in such a situation.
- *Issue protective equipment.* Many of the individuals who survived the World Trade Center disaster suffered smoke inhalation, eye injuries, etc. These types of injuries might be avoided if emergency equipment is issued to employees, such as hardhats, dust masks, goggles, flashlights, gloves, etc.

Building Security Procedures

A determined terrorist can penetrate most office buildings. However, the presence and use of guards and physical security devices (e.g., exterior lights, locks, mirrors, visual devices) create a significant psychological

deterrent. Terrorists are likely to shun risky targets for less protected ones. If terrorists decide to accept the risk, security measures can decrease their chance of success. Of course, if the terrorists are willing to die in the effort, their chance of success increases and the efforts to thwart them are much more complex and expensive. Corporate and government executives should develop comprehensive building security programs and frequently conduct security surveys that provide the basis for an effective building security program. These surveys generate essential information for the proper evaluation of security conditions and problems, available resources, and potential security policy. Only one of the many facets in a complex structure, security policies must be integrated with other important areas such as fire safety, normal police procedures, work environment, and work transactions. The building security checklist found in Exhibit 163.8 provides guidance when developing building security procedures.

Antiterrorism Procedures for Employees

Antiterrorism procedures can be defined as defensive measures used to reduce vulnerability to terrorist attacks. These defensive measures, or procedures, although originated by the U.S. government, are certainly applicable to those living in a high terrorist threat condition. To some security practitioners, many of these procedures may seem on the verge of paranoia; however, they are presented with two intentions: (1) to illustrate the varying dangers that exist and methods to avoid them, and (2) to allow readers to determine for themselves which procedures to use.

Many of the procedures are simply common sense. Others are procedures that are generally only known to those who live and work in high terrorist threat environments. See Exhibit 163.9 for the personnel antiterrorism checklist.

Lessons Learned from September 11

> Our plan worked and did what it was supposed to do. Our employees were evacuated safely.
>
> **— Paul Honey**
>
> **Director of Global Contingency Planning for Merrill Lynch**

Many well-prepared organizations weathered the disaster of September 11. However, there were also many businesses caught unprepared; of those, many no longer exist. Organizations from around the United States and the world are benefiting from the lessons learned on that fateful day. One large and quite well-known organization that was well prepared and survived the event was Merrill Lynch.

When Paul Honey, director of global contingency planning for Merrill Lynch, arrived for work on the morning of September 11, he was met by the disaster of the collapsed World Trade Center. Honey then went to one of the company's emergency command centers, where his contingency planning staff was hard at work. Within an hour of the disaster, the crisis management team had already established communication with key representatives, and emergency procedures were well underway.

Honey's team was able to facilitate the resumption of critical operations within one day and, within a week, the relocation of 8000 employees. This effort required the activation of a well-documented and robust business continuity program, an enormous communications effort, and a lot of teamwork.

Business Continuity Plans

Honey has business continuity planning responsibility for all of Merrill Lynch's businesses. He runs a team of 19 planners who verify that the business
follows the business continuity plan, or BCP. His team is not responsible for the technology recovery planning, and they do not write the plans. They are the subject matter experts in program management and set the standards through a complete BCP program life cycle. Planning involves many different departments within the company because of the comprehensive nature of the program. Each business and support group (i.e., the trading floor, operations, finance, etc.) assigns a planning manager who is responsible for that area.

EXHIBIT 163.8 Building Security Checklist

Office Accessibility

- Buildings most likely to be terrorist targets should not be directly accessible to the public.
- Executive offices should not be located on the ground floor.
- Place ingress door within view of the person responsible for screening personnel and objects passing through the door.
- Doors may be remotely controlled by installing an electromagnetic door lock.
- The most effective physical security configuration is to have doors locked from within and have only one visitor access door into the executive office area. Locked doors should also have panic bars.
- Depending on the nature of the organization's activities, deception measures such as a large waiting area controlling access to several offices can be taken to draw attention away from the location and function of a particular office.

Physical Security Measures

- Consider installing the following security devices: burglar alarm systems (preferably connected to a central security facility), sonic warning devices or other intrusion systems, exterior floodlights, deadbolt locks on doors, locks on windows, and iron grills or heavy screens for windows.
- Depending on the nature of the facility, consider installing a 15 to 20-foot fence or wall and a comprehensive external lighting system. External lighting is one of the cheapest and most effective deterrents to unlawful entry.
- Position light fixtures to make tampering difficult and noticeable.
- Check grounds to ensure that there are no covered or concealed avenues of approach for terrorists and other intruders, especially near entrances.
- Deny exterior access to fire escapes, stairway, and roofs.
- Manhole covers near the building should be secured or locked.
- Cover, lock, or screen outdoor openings (e.g., coal bins, air vents, utility access points).
- Screen windows (particularly near the ground or accessible from adjacent buildings.
- Consider adding a thin, clear plastic sheet to windows to degrade the effects of flying glass in case of explosion.
- Periodically inspect the interior of the entire building, including the basement and other infrequently used areas.
- Locate outdoor trash containers, storage bins, and bicycle racks away from the building.
- Book depositories or mail slots should not be adjacent to or in the building.
- Mailboxes should not be close to the building.
- Seal the top of voids and open spaces above cabinets, bookcases, and display cases.
- Keep janitorial closets, service openings, telephone closets, and electrical closets locked at all times. Protect communications closets and utility areas with an alarm system.
- Remove names from reserved parking spaces.
- Empty trash receptacles daily (preferably twice daily).
- Periodically check all fire extinguishers to ensure that they are in working order and readily available. Periodically check all smoke alarms to ensure that they are in working order.

Personnel Procedures

- Stress heightened awareness of personnel working in the building, because effective building security depends largely on the actions and awareness of people.
- Develop and disseminate clear instructions on personnel security procedures.
- Hold regular security briefings for building occupants.
- Personnel should understand security measures, appropriate responses, and should know whom to contact in an emergency.
- Conduct drills if appropriate.
- Senior personnel should not work late on a routine basis. No one should ever work alone.
- Give all personnel, particularly secretaries, special training in handling bomb threats and extortion telephone calls. Ensure a bomb threat checklist and a pen or pencil is located at each telephone.
- Ensure the existence of secure communications systems between senior personnel, secretaries, and security personnel with intercoms, telephones, and duress alarm systems.
- Develop an alternate means of communications (e.g., two-way radio) in case the primary communications systems fail.
- Do not open packages or large envelopes in buildings unless the sender or source is positively known. Notify security personnel of a suspicious package.
- Have mail room personnel trained in bomb detection handling and inspection.

EXHIBIT 163.8 Building Security Checklist (continued)

- Lock all doors at night, on weekends, and when the building is unattended.
- Maintain tight control of keys. Lock cabinets and closets when not in use.
- When feasible, lock all building rest rooms when not in use.
- Escort visitors in the building and maintain complete control of strangers who seek entrance.
- Check janitors and their equipment before admitting them and observe while they are performing their functions.
- Secure official papers from unauthorized viewing.
- Do not reveal the location of building personnel to callers unless they are positively identified and have a need for this information.
- Use extreme care when providing information over the telephone.
- Do not give the names, positions, and especially the home addresses or phone numbers of office personnel to strangers or telephone callers.
- Do not list the addresses and telephone numbers of potential terrorist targets in books and rosters.
- Avoid discussing travel plans or timetables in the presence of visitors.
- Be alert to people disguised as public utility crews who might station themselves near the building to observe activities and gather information.
- Note parked or abandoned vehicles, especially trucks, near the entrance to the building or near the walls.
- Note the license plate number, make, model, year, and color of suspicious vehicles and the occupant's description, and report that information to your supervisor, security officer, or law enforcement agency.

Controlling Entry

- Consider installing a peephole, intercom, interview grill, or small aperture in entry doorways to screen visitors before the door is opened.
- Use a reception room to handle visitors, thereby restricting their access to interior offices.
- Consider installing metal detection devices at controlled entrances. Prohibit non-organization members from bringing boxes and parcels into the building.
- Arrange building space so that unescorted visitors are under the receptionist's visual observation and to ensure that the visitors follow stringent access control procedures.
- Do not make exceptions to the building's access control system.
- Upgrade access control systems to provide better security through the use of intercoms, access control badges or cards, and closed-circuit television.

Public Areas

- Remove all potted plants and ornamental objects from public areas.
- Empty trash receptacles frequently.
- Lock doors to service areas.
- Lock trapdoors in the ceiling or floor, including skylights.
- Ensure that construction or placement of furniture and other items would not conceal explosive devices or weapons.
- Keep furniture away from walls or corners.
- Modify curtains, drapes, or cloth covers so that concealed items can be seen easily.
- Box in the tops of high cabinets, shelves, or other fixtures.
- Exercise particular precautions in public rest rooms.
- Install springs on stall doors in rest rooms so they stand open when not locked. Equip stalls with an inside latch to prevent someone from hiding a device in a locked stall.
- Install a fixed covering over the tops on commode water tanks.
- Use open mesh baskets for soiled towels. Empty frequently.
- Guards in public areas should have a way to silently alert the office of danger and to summon assistance (e.g., foot-activated buzzer).

Discovery of a Suspected Explosive Device

- Do not touch or move a suspicious object. If it is possible for someone to account for the presence of the object, then ask the person to identify it with a verbal description. This should not be done if it entails bringing evacuated personnel back into the area. Take the following actions if an object's presence remains inexplicable:
- Evacuate buildings and surrounding areas, including the search team.
- Evacuated areas must be at least 100 meters from the suspicious object.
- Establish a cordon and incident control point, or ICP.

EXHIBIT 163.8 Building Security Checklist (continued)

- Inform the ICP that an object has been found.
- Keep person who located the object at the ICP until questioned.
- Cordon suspicious objects to a distance of at least 100 meters and cordon suspicious vehicles to a distance of at least 200 meters. Ensure that no one enters the cordoned area. Establish an ICP on the cordon to control access and relinquish ICP responsibility to law enforcement authorities upon their arrival. Maintain the cordon until law enforcement authorities have completed their examination or state that the cordon may stand down. The decision to allow reoccupation of an evacuated facility rests with the individual in charge of the facility.

Honey's team responds to nearly 70 emergencies, on average, during the course of a year. Facilities and retail branch offices around the globe experience a variety of incidents such as earthquakes, storms, power outages, floods, or bomb threats.

When Honey's team plans for business interruption, the team instructs the business groups to plan for a worst-case scenario of six weeks without access to their facility and, naturally, at the worst possible time for an outage.

The planning also includes having absolutely no access to anything from any building — computers, files, papers, etc. "That's how we force people to think about alternate sites, vital records, physical relocation of staff, and so on, as well as obviously making sure the technology is available at another site," says Honey.

Upgraded Plans and Procedures after Y2K

Merrill Lynch must comply with standards mandated by regulatory agencies such as the Federal Reserve and the Federal Financial Institutions Examination Council. Honey says, "There's a market expectation that companies such as Merrill Lynch would have very robust contingency plans, so we probably attack it over and above any regulatory requirements that are out there." The BCP team's recent efforts to exceed regulatory standards placed Merrill Lynch in a good position to recover successfully from the September 11 attacks.

Extensive Testing of Contingency Plans

All plans are tested twice annually, and once a year the large-scale, corporatewide plans are tested. Honey's team overhauled the headquarters evacuation plan earlier in the year. They distributed nearly 8000 placards with the new procedures. These placards proved quite useful on the day of the attacks. Furthermore, the company's human resources database is downloaded monthly into the team's business continuity planning software program. This ensures that the BCP team has a frequently updated list of all current employees within each building. All this preparation resulted in effective execution of the business continuity plans on September 11.

Recent Test Using Scenario Similar to Terrorist Attacks

In May 2001, Honey's team conducted a two-day planning scenario for the headquarters' key staff. The scenario, although different from September 11, covered an event of devastating impact — a major hurricane in New York City. "While the hurricane scenario doesn't compare to the tragedies of 9/11 in terms of loss of life, we actually put our company through a fairly extensive two-day scenario, which had more impact to the firm in terms of difficulties in transportation and actual damage in the region," says Honey. "So, we were really very well prepared; we had a lot of people who already thought through a lot of the logistical, technology, and HR-type issues."

The Evacuation

The corporate response team was activated at about 8:55 a.m., while Honey was en route to Canal Street. The team, comprised of representatives from all business support groups, is instrumental in assessing the situation, such as building management, physical security personnel, media relations, key technology resources, and key business units. Despite a multitude of telecommunications troubles in the area, the team was finally able to

EXHIBIT 163.9 Personnel antiterrorism checklist

General Security Procedures

- Instruct your family and associates not to provide strangers with information about you or your family.
- Avoid giving unnecessary personal details to information collectors.
- Report all suspicious persons loitering near your residence or office; attempt to provide a complete description of the person and/or vehicle to police or security.
- Vary daily routines to avoid habitual patterns.
- If possible, fluctuate travel times and routes to and from work.
- Refuse to meet with strangers outside your workplace.
- Always advise associates or family members of your destination when leaving the office or home and the anticipated time of arrival.
- Do not open doors to strangers.
- Memorize key phone numbers — office, home, police, etc. Be cautious about giving out information regarding family travel plans or security measures and procedures.
- If you travel overseas, learn and practice a few key phrases in the native language, such as "I need a policeman, doctor," etc.

Business Travel

- Airport Procedures
  - — Arrive early; watch for suspicious activity.
  - — Notice nervous passengers who maintain eye contact with others from a distance. Observe what people are carrying. Note behavior not consistent with that of others in the area.
  - — No matter where you are in the terminal, identify objects suitable for cover in the event of attack; pillars, trash cans, luggage, large planters, counters, and furniture can provide protection.
  - — Do not linger near open public areas. Quickly transit waiting rooms, commercial shops, and restaurants.
  - — Proceed through security checkpoints as soon as possible.
  - — Avoid secluded areas that provide concealment for attackers.
  - — Be aware of unattended baggage anywhere in the terminal.
  - — Be extremely observant of personal carry-on luggage. Thefts of briefcases designed for laptop computers are increasing at airports worldwide; likewise, luggage not properly guarded provides an opportunity for a terrorist to place an unwanted object or device in your carry-on bag. As much as possible, do not pack anything you cannot afford to lose; if the documents are important, make a copy and carry the copy.
  - — Observe the baggage claim area from a distance. Do not retrieve your bags until the crowd clears. Proceed to the customs lines at the edge of the crowd.
  - — Report suspicious activity to the airport security personnel.
- On-Board Procedures
  - — Select window seats; they offer more protection because aisle seats are closer to the hijackers' movements up and down the aisle.
  - — Rear seats also offer more protection because they are farther from the center of hostile action, which is often near the cockpit.
  - — Seats at an emergency exit may provide an opportunity to escape.
- Hotel Procedures
  - — Keep your room key on your person at all times.
  - — Be observant for suspicious persons loitering in the area.
  - — Do not give your room number to strangers.
  - — Keep your room and personal effects neat and orderly so you will recognize tampering or strange out-of-place objects.
  - — Know the locations of emergency exits and fire extinguishers.
  - — Do not admit strangers to your room.
  - — Know how to locate hotel security guards.

Keep a Low Profile

- Your dress, conduct, and mannerisms should not attract attention.
- Make an effort to blend into the local environment.

EXHIBIT 163.9 Personnel Antiterrorism Checklist (continued)

- Avoid publicity and do not go out in large groups.
- Stay away from civil disturbances and demonstrations.

Tips for the Family at Home

- Restrict the possession of house keys.
- Change locks if keys are lost or stolen and when moving into a previously occupied residence.
- Lock all entrances at night, including the garage.
- Keep the house locked, even if you are at home.
- Develop friendly relations with your neighbors.
- Do not draw attention to yourself; be considerate of neighbors.
- Avoid frequent exposure on balconies and near windows.

Be Suspicious

- Be alert to public works crews requesting access to residence; check their identities through a peephole before allowing entry.
- Be alert to peddlers and strangers.
- Write down license numbers of suspicious vehicles; note descriptions of occupants.
- Treat with suspicion any inquiries about the whereabouts or activities of other family members.
- Report all suspicious activity to police or local law enforcement.

Security Precautions when You Are Away

- Leave the house with a lived-in look.
- Stop deliveries or forward mail to a neighbor's home.
- Do not leave notes on doors.
- Do not hide keys outside house.
- Use a timer (appropriate to local electricity) to turn lights on and off at varying times and locations.
- Leave radio on (best with a timer).
- Hide valuables.
- Notify the police or a trusted neighbor of your absence.

Residential Security

- Exterior grounds:
  - — Do not put your name on the outside of your residence or mailbox.
  - — Have good lighting.
  - — Control vegetation to eliminate hiding places.
- Entrances and exits should have:
  - — Solid doors with deadbolt locks
  - — One-way peepholes in door
  - — Bars and locks on skylights
  - — Metal grating on glass doors, and ground-floor windows, with interior release mechanisms that are not reachable from outside
- Interior features:
  - — Alarm and intercom systems
  - — Fire extinguishers
  - — Medical and first-aid equipment
- Other desirable features:
  - — A clear view of approaches
  - — More than one access road
  - — Off-street parking
  - — High (six to eight feet) perimeter wall or fence

EXHIBIT 163.9 Personnel Antiterrorism Checklist (continued)

Parking

- Always lock your car.
- Do not leave it on the street overnight, if possible.
- Never get out without checking for suspicious persons. If in doubt, drive away.
- Leave only the ignition key with parking attendant.
- Do not allow entry to the trunk unless you are there to watch.
- Never leave garage doors open or unlocked.
- Use a remote garage door opener if available. Enter and exit your car in the security of the closed garage.

On the Road

- Before leaving buildings to get into your vehicle, check the surrounding area to determine if anything of a suspicious nature exists. Display the same wariness before exiting your vehicle.
- Prior to getting into a vehicle, check beneath it. Look for wires, tape, or anything unusual.
- If possible, vary routes to work and home.
- Avoid late-night travel.
- Travel with companions.
- Avoid isolated roads or dark alleys when possible.
- Habitually ride with seatbelts buckled, doors locked, and windows closed.
- Do not allow your vehicle to be boxed in; maintain a minimum eight-foot interval between you and the vehicle in front; avoid the inner lanes. Be alert while driving or riding.

Know How to React if You Are Being Followed

- Circle the block for confirmation of surveillance.
- Do not stop or take other actions that could lead to confrontation.
- Do not drive home.
- Get description of car and its occupants.
- Go to the nearest safe haven.
- Report incident to police.

Recognize Events that can Signal the Start of an Attack:

- Cyclist falling in front of your car.
- Flagman or workman stopping your car.
- Fake police or government checkpoint.
- Disabled vehicle/accident victims on the road.
- Unusual detours.
- An accident in which your car is struck.
- Cars or pedestrian traffic that box you in.
- Sudden activity or gunfire.

Know What to Do if under Attack in a Vehicle:

- Without subjecting yourself, passengers, or pedestrians to harm, try to draw attention to your car by sounding the horn
- Put another vehicle between you and your pursuer
- Execute immediate turn and escape; jump the curb at 30–45 degree angle, 35 mph maximum
- Ram blocking vehicle if necessary
- Go to closest safe haven
- Report incident to police

Commercial Buses, Trains, and Taxis

- Vary mode of commercial transportation.
- Select busy stops.

EXHIBIT 163.9 Personnel Antiterrorism Checklist (continued)

- Do not always use the same taxi company.
- Do not let someone you do not know direct you to a specific cab.
- Ensure taxi is licensed and has safety equipment (seatbelts at a minimum).
- Ensure face of driver and picture on license are the same.
- Try to travel with a companion.
- If possible, specify the route you want the taxi to follow.

Clothing

- Travel in conservative clothing when using commercial transportation overseas or if you are to connect with a flight at a commercial terminal in a high-risk area.
- Do not wear U.S.-identified items such as cowboy hats or boots, baseball caps, American logo T-shirts, jackets, or sweatshirts.
- Wear a long-sleeved shirt if you have a visible U.S.-affiliated tattoo.

Actions if Attacked

- Dive for cover. Do not run. Running increases the probability of shrapnel hitting vital organs or the head.
- If you must move, belly crawl or roll. Stay low to the ground, using available cover.
- If you see grenades, lay flat on the floor, with feet and knees tightly together with soles toward the grenade. In this position, your shoes, feet, and legs protect the rest of your body. Shrapnel will rise in a cone from the point of detonation, passing over your body.
- Place arms and elbows next to your ribcage to protect your lungs, heart, and chest. Cover your ears and head with your hands to protect neck, arteries, ears, and skull.
- Responding security personnel will not be able to distinguish you from attackers. Do not attempt to assist them in any way. Lay still until told to get up.

Actions if Hijacked

- Remain calm, be polite, and cooperate with your captors.
- Be aware that all hijackers may not reveal themselves at the same time. A lone hijacker may be used to draw out security personnel for neutralization by other hijackers.
- Surrender your tourist passport in response to a general demand for identification.
- Do not offer any information.
- Do not draw attention to yourself with sudden body movements, verbal remarks, or hostile looks.
- Prepare yourself for possible verbal and physical abuse, lack of food and drink, and unsanitary conditions.
- If permitted, read, sleep, or write to occupy your time.
- Discretely observe your captors and memorize their physical descriptions. Include voice patterns and language distinctions as well as clothing and unique physical characteristics.
- Cooperate with any rescue attempt. Lie on the floor until told to rise.

establish a conference call at 9:30 a.m. to communicate with its other command center in Jersey City, New Jersey, to figure out what was happening.

"In hindsight it seems odd, but we really didn't know, apart from the planes hitting the buildings, whether this was an accident or a terrorist attack," says Honey. "So really, the challenge at that time was to account for our employees, and then to try and understand what had happened. The damage to our buildings also was a concern. How were our buildings. Were they still standing? What was the state of the infrastructure in them?"

Call trees were used to contact employees, and employees also knew how to contact their managers to let them know they got out of the area safely. "In a typical evacuation of a building, employees go about 100 yards from the building and wait to get their names ticked off a list," says Honey. "The issue we faced here is that the whole of lower Manhattan was evacuated. So employees were going home or trying to get to other offices — so that was a challenge for us." Honey says the wallet cards key employees carried were extremely beneficial. "Everyone knew who to call and when," he says. "That was a real valuable planning aid to have."

Once the team had the call trees and other communications processes under way, they began to implement the predefined continuity plans and assess what critical business items they wanted to focus on and when.

The Recovery

Critical Management Functions Resumed within Minutes

Many of the company's recovery procedures were based on backup data centers at Merrill Lynch facilities outside the area. The data recovery procedures were followed through without incident. The company has a hot site provider, but they did not have to use that service.

The company's preparedness efforts for Y2K resulted in near-routine recovery of critical data. "We had a very large IT disaster recovery program in place," says Honey, "and we've been working for a couple years now with the businesses to really strengthen the business procedures to use it. So backup data centers, mirroring over fiber channels, etc. — that all worked pretty well." Likewise for the recovery personnel at the command centers: "A lot of people already knew what a command center was, why they had to be there, and what they needed to do because we had gone through that during Y2K, and I'm very grateful that we did."

8000 Employees Back at Work within a Week

A major challenge for the BCP team was getting the displaced employees back to work. First, the company was able to utilize two campus facilities in New Jersey. The company also had its real estate department itemize every available space in the tri-state area and put it onto a roster. Honey's team collected requirements and coordinated the assignment of available space to each business unit. The company operates a fairly comprehensive alternate work arrangement program, so some employees were permitted to work from home. Finally, the team was able to transfer some work abroad or to other Merrill Lynch offices, which relieved some of the workload from the affected employees.

Resuming Normal Operations

By the end of the week, the BCP team's priority shifted to making sure they could communicate with all employees. Workers needed to be assured that the company was handling the crisis and that space was allocated for displaced workers. Messages were sent instructing them on where to go for more information and what human resource hotlines were available for them to call.

Merrill Lynch's chairman, CEO, and senior business and technology managers made prerecorded messages that were sent out automatically to all employees impacted by the incident by use of a special emergency communication system. This accounted for approximately 74,000 phone calls during the first week after the disaster. "That was a very key part," says Honey. "Getting accurate information to our employee base was a real challenge because of a lot of misinformation in the press, which makes the job very challenging. Plus, key business folks made a huge effort to call all our key customers and reassure them with the accurate information that Merrill Lynch was open for business."

A key logistical challenge was getting the thousands of displaced workers to their new work locations. The company ran a series of ferryboats and buses from various points within the city to other points. The company Web site was also used to communicate transportation information to the affected employees.

Lessons Learned

Honey and his team will be reevaluating certain aspects of their plans in the coming months, even after their success in recovering from such a devastating event, "One of the things I think we'll concentrate on a lot more in the future is region-wide disasters. For example, not so much, 'Your building is knocked out and you can't get in,' but maybe, 'The city you're in is impacted in significant ways.' So, we'll be looking to see how we can make the firm a lot more robust in terms of instances where a city is impacted, rather than just the building."

Honey also believes that many companies will reevaluate their real estate strategies. "Do you really want to have all your operations in one building?" he asks. "Fortunately, for a company like Merrill Lynch, we have a number of real estate options we can utilize."

The Work Ahead

The BCP team was busy working on backup plans for the backup facilities by the end of the second week, while primary sites were either cleaned up or acquired. "Many of our operations are in backup mode," says Honey, "so we did a lot of work to try and develop backup plans for the backup plans. That was a big challenge."

Now the team is in the planning stages for reoccupying the primary sites, which presents its own set of challenges. Switching back to primary facilities will have to be undertaken only when it is perfectly safe for employees to reoccupy the damaged facilities.

One of the most important things for Honey and his team was that, by the Monday morning following the attack, everything was back to nearly 95 percent of normal operations. Their efforts over the past few years preparing for a disruption of this magnitude appear to have paid off. "Certainly from my perspective, I was very glad that we put the company through the training exercise in May," says Honey. "It enlightened an awful lot of the key managers on what they would have to do, so we were very prepared for that. Most folks knew what to do, which was very reassuring to me."

Conclusion

Reducing vulnerability to physical security threats became immensely more complex after September 11, 2001. Terrorism now needs to be included in all physical security planning. The events of September 11 showed us that procedures designed to deter those with hostile intent might be ineffective against suicidal terrorists. Physical security now needs to change its operating paradigm from that of deterrence to prevention to reduce the risk from terrorism. Taking the additional precautions to prevent hostile acts rather than deter them is much more difficult and costly, but necessary. Protecting one's organization, co-workers, and family from terrorism is possible with training. Maintaining control of access to sensitive information that could be used by terrorists is paramount. Many government Web sites are awash with information that could be useful in combating terrorism. Unfortunately, many of these Web sites can also provide this information to potential terrorists who could use that information to discover vulnerabilities.

Dedication

This chapter is respectfully dedicated to those whose lives were lost or affected by the events of September 11, 2001. It is the author's deepest hope that information presented in this chapter will aid in reducing the likelihood of another such event.

Bibliography

1. NIPC Advisory 02-001: Internet Content Advisory: Considering the Unintended Audience, National Infrastructure Protection Center, January 17, 2002.
2. *Service Member's Personal Protection Guide: A Self-Help Handbook to Combating Terrorism*, U.S. Joint Chiefs of Staff, Joint Staff Guide 5260, July 1996.
3. *Joint Tactics, Techniques and Procedures for Antiterrorism*, U.S. Joint Chiefs of Staff, Joint Pub 3-07.2, 17 March 1998, Appendix.
4. *ATF Bomb Threat Checklist*, ATF-F 1613.1, Bureau of Alcohol, Tobacco and Firearms, June 1997.
5. Merrill Lynch Resumes Critical Business Functions within Minutes of Attack, Janette Ballman, *Disaster Recovery Journal*, 14, 4, p. 26, Fall 2001.

Appendix A
Glossary

Abend: The abnormal termination of a computer application or job because of a non-system condition or failure that causes a program to halt.

Abstraction: The process of identifying the characteristics that distinguish a collection of similar objects; the result of the process of abstraction is a type.

Acceptable Use Policy (AUP): A definition of what is acceptable online behavior, and what is not.

Acceptance Testing: The formal testing conducted to determine whether a software system satisfies its acceptance criteria, enabling the customer to determine whether to accept the system.

Access: The ability of a subject to view, change, or communicate with an object. Typically, access involves a flow of information between the subject and the object.

Access Control: The process of allowing only authorized users, programs, or other computer system (i.e., networks) to access the resources of a computer system.

Access Control List (ACL): Most network security systems operate by allowing selective use of service. An Access Control List is the usual means by which access to, and denial of, service is controlled. It is simply a list of the services available, each with a list of the hosts permitted to use the services.

Access Control Mechanisms: Hardware, software, or firmware features and operating and management procedures in various combinations designed to detect and prevent unauthorized access and to permit authorized access to a computer system.

Access Period: A segment of time, generally expressed on a daily or weekly basis, during which access rights prevail.

Access Type: The nature of access granted to a particular device, program, or file (e.g., read, write, execute, append, modify, delete, or create).

Accountability: A security principle stating that individuals must be able to be identified. With accountability, violations or attempted violations can be traced to individuals who can be held responsible for their actions.

Accreditation: A program whereby a laboratory demonstrates that something is operating under accepted standards to ensure quality assurance.

Acknowledgment (ACK): A type of message sent to indicate that a block of data arrived at its destination without error. A negative acknowledgment is called a "NAK."

Active Object: An object that has its own process; the process must be ongoing while the active object exists.

Active Wiretapping: The attachment of an unauthorized device (e.g., a computer terminal) to a communications circuit to gain access to data by generating false messages or control signals or by altering the communications of legitimate users.

ActiveX: Microsoft's Windows-specific non-Java technique for writing applets. ActiveX applets take considerably longer to download than the equivalent Java applets; however, they more fully exploit the features of Windows. ActiveX is sometimes said to be a "superset of Java."

Ada: A programming language that allows use of structured techniques for program design; concise but powerful language designed to fill government requirements for real-time applications.

Add-On Security: The retrofitting of protection mechanisms, implemented by hardware, firmware, or software, on a computer system that has become operational.

Address: (1) A sequence of bits or characters that identifies the destination and sometimes the source of a transmission. (2) An identification (e.g., number, name, or label) for a location in which data is stored.

Address Mapping: The process by which an alphabetic Internet address is converted into a numeric IP address, and vice versa.

Address Mask: A bit mask used to identify which bits in an IP address correspond to the network address and subnet portions of the address. This mask is often referred to as the subnet mask because the network portion of the address can be determined by the class inherent in an IP address. The address mask has ones in positions corresponding to the network and subnet numbers and zeros in the host number positions.

Address Resolution: A means for mapping network layer addresses onto media-specific addresses.

Address Resolution Protocol (ARP): The Internet protocol used to dynamically map Internet addresses to physical (hardware) addresses on the local area network. Limited to networks that support hardware broadcast.

Administrative Security: The management constraints, operational procedures, accountability procedures, and supplemental controls established to provide an acceptable level of protection for sensitive data.

Agent: In the client/server model, the part of the system that performs information preparation and exchange on behalf of a client or server application.

Aggregation: A relation, such as CONSISTS OF or CONTAINS between types that defines the composition of a type from other types.

Aging: The identification, by date, of unprocessed or retained items in a file. This is usually done by date of transaction, classifying items according to ranges of data.

Algorithm: A computing procedure designed to perform a task such as encryption, compressing, or hashing.

Aliases: Used to reroute browser requests from one URL to another.

American National Standards Institute (ANSI): The agency that recommends standards for computer hardware, software, and firmware design and use.

American Registry for Internet Numbers (ARIN): A nonprofit organization established for the purpose of administration and registration of Internet Protocol (IP) numbers to the geographical areas currently managed by Network Solutions (InterNIC). Those areas include, but are not limited to North America, South America, South Africa, and the Caribbean.

American Standard Code for Information Interchange (ASCII): A byte-oriented coding system based on an 8-bit code and used primarily to format information for transfer in a data communications environment.

Amplitude Modulation (AM): The technique of varying the amplitude or wavelength of a carrier wave in direct proportion to the strength of the input signal while maintaining a constant frequency and phase.

Analog: A voice transmission mode that is not digital in which information is transmitted in its original form by converting it to a continuously variable electrical signal.

Analysis and Design Phase: The phase of the systems development life cycle in which an existing system is studied in detail and its functional specifications are generated.

Annual Loss Expectancy (ALE): In risk assessment, the average monetary value of losses per year.

Anonymous FTP: A type of FTP that allows a user to log on to a remote host, which the user would otherwise not have access to, to download files.

ANSI: *See* American National Standards Institute.

Applet: A small Java program embedded in an HTML document.

Application: Computer software used to perform a distinct function. Also used to describe the function itself.

Application Layer: The top-most layer in the OSI Reference Model providing such communication service is invoked through a software package.

Application Objects: Applications and their components that are managed within an object-oriented system. Example operations on such objects are OPEN, INSTALL, MOVE, and REMOVE.

Application Program Interface (API): A set of calling conventions defining how a service is invoked through a software package.

Architecture: The structure or ordering of components in a computational or other system. The classes and the interrelation of the classes define the architecture of a particular application. At another level, the architecture of a system is determined by the arrangement of the hardware and software components. The terms "logical architecture" and "physical architecture" are often used to emphasize this distinction.

Array: Consecutive storage areas in memory that are identified by the same name. The elements (or groups) within these storage areas are accessed through subscripts.

Artificial Intelligence (AI): A field of study involving techniques and methods under which computers can simulate such human intellectual activities as learning.

Assembler Language: A computer programming language in which alphanumeric symbols represent computer operations and memory addresses. Each assembler instruction translates into a single machine language instruction.

Assembler Program: A program language translator that converts assembler language into machine code.

Asynchronous: A variable or random time interval between successive characters, blocks, operations, or events. Asynchronous data transmission provides variable intercharacter time but fixed interbit time within characters.

Asynchronous Transfer Mode (ATM): A transfer mode in which data is transmitted in the form of 53-byte units called cells. Each cell consists of a 5-byte header and a 48-byte payload. The term "asynchronous" in this context refers to the fact that cells from any one particular source need not be periodically spaced within the overall cell stream. That is, users are not assigned a set position in a recurring frame as is common in circuit switching.

Atomicity: The assurance that an operation either changes the state of all participating objects consistent with the semantics of the operation or changes none at all.

Attribute: A characteristic defined for a class. Attributes are used to maintain the state of the object of a class. Values can be connected to objects via the attributes of the class. Typically, the connected value is determined by an operation with a single parameter identifying the object. Attributes implement the properties of a type.

Audit: An independent review and examination of system records and activities that test for the adequacy of system controls, ensure

compliance with established policy and operational procedures, and recommend any indicated changes in controls, policy, and procedures.

Audit trail: A chronological record of system activities that is sufficient to enable the reconstruction, review, and examination of each event in a transaction from inception to output of final results.

Authentication: The act of identifying or verifying the eligibility of a station, originator, or individual to access specific categories of information. Typically, a measure designed to protect against fraudulent transmissions by establishing the validity of a transmission, message, station, or originator.

Authorization: The granting of right of access to a user, program, or process.

Backbone: The primary connectivity mechanism of a hierarchical distributed system. All systems that have connectivity to an intermediate system on the backbone are assured of connectivity to each other.

Backoff: The (usually random) retransmission delay enforced by contentious MAC protocols after a network node with data to transmit determines that the physical medium is already in use.

Backup and Recovery: The ability to recreate current master files using appropriate prior master records and transactions.

Backup Procedures: Provisions make for the recovery of data files and program libraries and for the restart or replacement of computer equipment after the occurrence of a system failure or disaster.

Bandwidth: Difference between the highest and lowest frequencies available for network signals. The term is also used to describe the rated throughput capacity of a given network medium or protocol.

Baseband: Characteristic of any network technology that uses a single carrier frequency and requires all stations attached to the network to participate in every transmission. *See* broadband.

BCP: The newest subseries of RFCs that are written to describe Best Current Practices in the Internet. Rather than specify the best ways to use the protocols and the best ways to configure options to ensure interoperability between various vendors' products, BCPs carry the endorsement of the IESG.

Between-the-Lines Entry: Access obtained through the use of active wiretapping by an unauthorized user to a momentarily inactive terminal of a legitimate user assigned to a communications channel.

BIOS: The BIOS is built-in software that determines what a computer can do without accessing programs from a disk. On PCs, the BIOS contains all the code required to control the keyboard, display screen, disk drives, serial communications, and a number of miscellaneous functions.

Bit: A binary value represented by an electronic component that has a value of 0 or 1.

Bit Error Rate (BER): The probability that a particular bit will have the wrong value.

Bit Map: A specialized form of an index indicating the existence or non-existence of a condition for a group of blocks or records. Although they are expensive to build and maintain, they provide very fast comparison and access facilities.

Bit Mask: A pattern of binary values that is combined with some value using bitwise AND with the result that bits in the value in positions where the mask is zero are also set to zero.

Bit Rate: This is the speed at which bits are transmitted on a circuit, usually expressed in bits per second.

Block Cipher: A method of encrypting text to produce ciphertext in which a cryptographic key and algorithm are applied to a block of data as a group instead of one bit at a time.

Body: One of four possible components of a message. Other components are the headings, attachment, and the envelope.

Bounds Checking: The testing of computer program results for access to storage outside of its authorized limits.

Bridge: A device that connects two or more physical networks and forwards packets between them. Bridges can usually be made to filter packets, that is, to forward only certain traffic.

Broadband: Characteristic of any network that multiplexes multiple, independent network carriers onto a single cable. Broadband technology allows several networks to coexist on one single cable; traffic from one network does not interfere with traffic from another because the conversations happen on different frequencies in the "ether," rather like the commercial radio system.

Broadcast: A packet delivery system where a copy of a given packet is given to all hosts attached to the network. Example: Ethernet.

Broadcast Storm: A condition that can occur on broadcast type networks such as Ethernet. This can happen for a number of reasons, ranging from hardware malfunction to configuration error and bandwidth saturation.

Brouter: A concatenation of "bridge" and "router." Used to refer to devices that perform both bridging and routing.

Browser: Short for *Web browser,* a software application used to locate and display Web pages. The two most popular browsers are Netscape Navigator and Microsoft Internet Explorer. Both of these are *graphical browsers,* which means that they can display graphics as well as text. In addition, most modern browsers can present multimedia information, including sound and video, although they require plug-ins for some formats.

Browsing: The searching of computer storage to locate or acquire information, without necessarily knowing whether it exists or in what format.

Buffer *(n):* A temporary storage area, usually in RAM. The purpose of most buffers is to act as a holding area, enabling the CPU to manipulate data before transferring it to a device. Because the processes of reading and writing data to a disk are relatively slow, many programs keep track of data changes in a buffer and then copy the buffer to a disk. For example, word processors employ a buffer to keep track of changes to files. Then when you *save* the file, the word processor updates the disk file with the contents of the buffer. This is much more efficient than accessing the file on the disk each time you make a change to the file. Note that because your changes are initially stored in a buffer, not on the disk, all of them will be lost if the computer fails during an editing session. For this reason, it is a good idea to save your file periodically. Most word processors automatically save files at regular intervals. Another common use of buffers is for printing documents. When you enter a PRINT command, the operating system copies your document to a print buffer (a free area in memory or on a disk) from which the printer can draw characters at its own pace. This frees the computer to perform other tasks while the printer is running in the background. Print buffering is called *spooling*. Most keyboard drivers also contain a buffer so that you can edit typing mistakes before sending your command to a program. Many operating systems, including DOS, also use a *disk buffer* to temporarily hold data that they have read from a disk. The disk buffer is really a cache.

Bug: A coded program statement containing a logical or syntactical error.

Bulletin Board Service (BBS): A computer that allows you to log on and post messages to other subscribers to the service.

Burn Box: A device used to destroy computer data. Usually a box with magnets or electrical current that will degauss disks and tapes.

Bus: A data path that connects the CPU, input, output, and storage devices.

Business Continuity Plan (BCP): A documented and tested plan for responding to an emergency.

Byte: The basic unit of storage for many computers; typically, one configuration consists of 8 bits used to represent data plus a parity bit for checking the accuracy of representation.

C: A third-generation computer language used for programming on microcomputers. Most microcomputer software products such as spreadsheets and DBMS programs are written in C.

Cable: Transmission medium of copper wire or optical fiber wrapped in a protective cover.

Cache: Pronounced *cash*, a special high-speed storage mechanism. It can be either a reserved section of main memory or an independent high-speed storage device. Two types of caching are commonly used in personal computers: *memory caching* and *disk caching*. A memory

cache, sometimes called a *cache store* or *RAM cache*, is a portion of memory made of high-speed static RAM (SRAM) instead of the slower and cheaper dynamic RAM (DRAM) used for main memory. Memory caching is effective because most programs access the same data or instructions over and over. Disk caching works under the same principle as memory caching, but instead of using high-speed SRAM, a disk cache uses conventional main memory. When data is found in the cache, it is called a *cache hit,* and the effectiveness of a cache is judged by its *hit rate.*

Callback: A procedure that identifies a terminal dialing into a computer system or network by disconnecting the calling terminal, verifying the authorized terminal against the automated control table, and then, if authorized, reestablishing the connection by having the computer system dial the telephone number of the calling terminal.

Carrier Sense, Multiple Access (CSMA): A multiple-station access scheme for avoiding contention in packet networks in which each station can sense the presence of carrier signals from other stations and thus avoid transmitting a packet that would result in a collision. *See also* collision detection.

Central Processing Unit (CPU): The part of the computer system containing the control and arithmetic logic units.

CERN: European Laboratory for Particle Physics. Birthplace of the World Wide Web.

Certification: The acceptance of software by an authorized agent, usually after the software has been validated by the agent or its validity has been demonstrated to the agent.

Chain of Custody: The identity of persons who handle evidence between the time of commission of the alleged offense and the ultimate disposition of the case. It is the responsibility of each transferee to ensure that the items are accounted for during the time that it is in their possession, that it is properly protected, and that there is a record of the names of the persons from whom they received it and to whom they delivered it, together with the time and date of such receipt and delivery.

Chain of Evidence: The "sequencing" of the chain of evidence follows this order:

Collection and identification
Analysis
Storage
Preservation
Presentation in court
Return to owner

Chain of evidence shows:

Who obtained the evidence
Where and when the evidence was obtained

Who secured the evidence
Who had control or possession of the evidence

CHAP (Challenge Handshake Authentication Protocol): Applies a three-way handshaking procedure. After the link is established, the server sends a "challenge" message to the originator. The originator responds with a value calculated using a one-way hash function. The server checks the response against its own calculation of the expected hash value. If the values match, the authentication is acknowledged; otherwise, the connection is usually terminated.

Chat Room: An area of a Web chat service that people can "enter" with their Web browsers where the conversations are devoted to a specific topic; equivalent to a channel in IRC.

Check Digit: One digit, usually the last, of an identifying field is a mathematical function of all of the other digits in the field. This value can be calculated from the other digits in the field and compared with the check digit to verify validity of the whole field.

Checksum: A computed value that depends on the contents of a packet. This value is sent along with the packet when it is transmitted. The receiving system computes a new checksum based on receiving data and compares this value with the one sent with the packet. If the two values are the same, the receiver has a high degree of confidence that the data was received correctly.

Ciphertext: Information that has been encrypted, making it unreadable without knowledge of the key.

Circuit Switching: A communications paradigm in which a dedicated communication path is established between two hosts and on which all packets travel. The telephone system is an example of a circuit-switched network.

Class: An implementation of an abstract data type. A definition of the data structures, methods, and interface of software objects. A template for the instantiation (creation) of software objects.

Client: A workstation in a network that is set up to use the resources of a server.

Client/Server: In networking, a network in which several PC-type systems (clients) are connected to one or more powerful, central computers (servers). In databases, refers to a model in which a client system runs a database application (front end) that accesses information in a database management system situated on a server (back end).

Client/Server Architecture: A local area network in which microcomputers, called servers, provide specialized service on behalf of the user's computers, which are called clients.

Cloning: The term given to the operation of creating an exact duplicate of one medium on another like medium. This is also referred to as a Mirror Image or Physical Sector Copy.

Coaxial Cable: A medium used for telecommunications. It is similar to the type of cable used for carrying television signals.

Code Division Multiple Access (CDMA): A technique permitting the use of a single frequency band by a number of users. Users are allocated a sequence that uniquely identifies them.

Collision: This is a condition that is present when two or more terminals are in contention during simultaneous network access attempts.

Collision Detection: An avoidance method for communications channel contention that depends on two stations detecting the simultaneous start of each other's transmission, stopping, and waiting a random period of time before beginning again. *See also* carrier sense, multiple access.

Commit: A condition implemented by the programmer signaling to the DBMS that all update activity that the program conducts be executed against a database. Before the commit, all update activity can be rolled back or canceled without negative impact on the database contents.

Commit Protocol: An algorithm to ensure that a transaction is successfully completed.

Common Business Oriented Language (COBOL): A high-level programming language for business computer applications.

Common Carrier: An organization or company that provides data or other electronic communication services for a fee.

Common Object Request Broker Architecture (CORBA): CORBA is the Object Management Group's (OMG) answer to the need for interoperability among the rapidly proliferating number of hardware and software products available today. Simply stated, CORBA allows applications to communicate with one another no matter where they are located or who has designed them.

Communications Security: The protection that ensures the authenticity of telecommunications and that results from the application of measures taken to deny unauthorized persons access to valuable information that might be derived from the acquisition of telecommunications.

Compartmentalization: The isolation of the operating system, user programs, and data files from one another in main storage to protect them against unauthorized or concurrent access by other users or programs. Also, the division of sensitive data into small, isolated blocks to reduce risk to the data.

Compiler: A program that translates high-level computer language instructions into machine code.

Compromise: Unauthorized disclosure or loss of sensitive information.

Compromising Emanations: Electromagnetic emanations that convey data and that, if intercepted and analyzed, could compromise sensitive information being processed by a computer system.

Computer Emergency Response Team (CERT): The CERT is chartered to work with the Internet community to facilitate its response to computer security events involving Internet hosts, to take proactive steps to raise the community's awareness of computer security issues, and to conduct research targeted at improving the security of existing systems. The U.S. CERT is based at Carnegie Mellon University in Pittsburgh; regional CERTs are like NICs, springing up in different parts of the world.

Computer Evidence: Computer evidence is a copy of a document stored in a computer file that is identical to the original. The legal "best evidence" rules change when it comes to the processing of computer evidence. Another unique aspect of computer evidence is the potential for unauthorized copies to be made of important computer files without leaving behind a trace that the copy was made. This situation creates problems concerning the investigation of the theft of trade secrets (e.g., client lists, research materials, computer-aided design files, formulas, and proprietary software).

Computer Forensics: The term "computer forensics" was coined in 1991 in the first training session held by the International Association of Computer Specialists (IACIS) in Portland, Oregon. Since then, computer forensics has become a popular topic in computer security circles and in the legal community. Like any other forensic science, computer forensics deals with the application of law to a science. In this case, the science involved is computer science and some refer to it as Forensic Computer Science. Computer forensics has also been described as the autopsy of a computer hard disk drive because specialized software tools and techniques are required to analyze the various levels at which computer data is stored after the fact. Computer forensics deals with the preservation, identification, extraction, and documentation of computer evidence. The field is relatively new to the private sector, but it has been the mainstay of technology-related investigations and intelligence gathering in law enforcement and military agencies since the mid-1980s. Like any other forensic science, computer forensics involves the use of sophisticated technology tools and procedures that must be followed to guarantee the accuracy of the preservation of evidence and the accuracy of results concerning computer evidence processing. Typically, computer forensic tools exist in the form of computer software.

Computer Fraud and Abuse Act PL 99-474: Computer Fraud and Abuse Act of 1986. Strengthens and expands the 1984 Federal Computer Crime Legislation. Law extended to computer crimes in private enterprise and anyone who willfully disseminates information for the purpose of committing a computer crime (i.e., distribute phone numbers to hackers from a BBS).

Computer Matching Act Public Law (PL) 100-53: Computer Matching and Privacy Act of 1988. Ensures privacy, integrity, and verification of data disclosed for computer matching; establishes Data Integrity Boards within federal agencies.

Computer Security: The practice of protecting a computer system against internal failures, human error, attacks, and natural catastrophes that might cause improper disclosure, modification, destruction, or denial-of-service.

Computer Security Act PL 100-235: Computer Security Act of 1987 directs the National Bureau of Standards (now the National Institute of Standards and Technology [NIST]) to establish a computer security standards program for federal computer systems.

Computer System: An interacting assembly of elements, including at least computer hardware and usually software, data procedures, and people.

Computer System Security: All of the technological safeguards and managerial procedures established and applied to computers and their networks (including related hardware, firmware, software, and data) to protect organizational assets and individual privacy.

Computer-Aided Software Engineering (CASE): Tools that automate the design, development, operation, and maintenance of software.

Concealment Systems: A method of keeping sensitive information confidential by embedding it in irrelevant data.

Concurrent Processing: The capability of a computer to share memory with several programs and simultaneously execute the instructions provided by each.

Condensation: The process of reducing the volume of data managed without reducing the logical consistency of data. It is essentially different than compaction in that condensation is done at the record level whereas compaction is done at the system level.

Confidentiality: A concept that applies to data that must be held in confidence and describes that status or degree of protection that must be provided for such data about individuals as well as organizations.

Configuration Management: The use of procedures appropriate for controlling changes to a system's hardware, software, or firmware structure to ensure that such changes will not lead to a weakness or fault in the system.

Connection-Oriented: The model of interconnection in which communication proceeds through three well-defined phases: connection establishment, data transfer, and connection release. Examples: X.25, Internet TCP and OSI TP4, ordinary telephone calls.

Connectionless: The model of interconnection in which communication takes place without first establishing a connection. Sometimes (imprecisely) called datagram. Examples: Internet IP and OSI CLNP, UDP, ordinary postcards.

Console Operator: Someone who works at a computer console to monitor operations and initiate instructions for efficient use of computer resources.

Construct: An object; especially a concept that is constructed or synthesized from simple elements.

Contention: Occurs during multiple access to a network in which the network capacity is allocated on a "first come, first served" basis.

Contingency Plans: Plans for emergency response, backup operations, and post-disaster recovery maintained by a computer information processing facility as a part of its security program.

Control: Any protective action, device, procedure, technique, or other measure that reduces exposures.

Control Break: A point during program processing at which some special processing event takes place. A change in the value of a control field within a data record is characteristic of a control break.

Control Totals: Accumulations of numeric data fields that are used to check the accuracy of the input, processing, or output data.

Control Zone: The space surrounding equipment that is used to process sensitive information and that is under sufficient physical and technical control to preclude an unauthorized entry or compromise.

Cookie: A small file stored on your computer by a Web browser that tracks your surfing activity.

Cooperative Processing: The ability to distribute resources (i.e., programs, files, and databases) across the network.

Copy: An accurate reproduction of information contained on an original physical item, independent of the original physical item.

Copyright: The author or artist's right to control the copying of his or her work.

CORBA: Common Object Request Broker Architecture, introduced in 1991 by the OMG, defined the Interface Definition Language (IDL) and the Application Programming Interfaces (APIs) that enable client/server object interaction within a specific implementation of an Object Request Broker (ORB).

Corrective Action: The practice and procedure for reporting, tracking, and resolving identified problems, in both the software product and the development process. Their resolution provides a final solution to the identified problem.

Corrective Maintenance: The identification and removal of code defects.

Cost/Benefit Analysis: Determination of the economic feasibility of developing a system on the basis of a comparison of the projected costs of a proposed system and the expected benefits from its operation.

Cost-Risk Analysis: The assessment of the cost of potential risk of loss or compromise of data in a computer system without data protection versus the cost of providing data protection.

CPU: The central processing unit; the brains of the computer.

Critical Path: A tool used in project management techniques and is the duration based on the sum of the individual tasks and their dependencies. The critical path is the shortest period in which a project can be accomplished.

Crossover Error Rate (CER): A comparison metric for different biometric devices and technologies; the error rate at which FAR equals FRR. The lower the CER, the more accurate and reliable the biometric device.

Cryptanalysis: The study of techniques for attempting to defeat cryptographic techniques and, more generally, information security services.

Cryptanalyst: Someone who engages in cryptanalysis.

Cryptography: The study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication. Cryptography is not the only means of providing information security services, but rather one set of techniques. The word itself comes from the Greek word *kryptos,* which means "hidden" or "covered." Cryptography is a way to hide writing ("-graphy") but yet retain a way to uncover it again.

Cryptology: The field of study that encompasses both cryptography and cryptanalysis.

Cryptosystem: A general term referring to a set of cryptographic primitives used to provide information security services.

Cyberspace: A term coined to denote the online world of the Internet.

Cyclic Redundancy Check (CRC): A number derived from a set of data that will be transmitted.

Data: Raw facts and figures that are meaningless by themselves. Data can be expressed in characters, digits, and symbols, which can represent people, things, and events.

Data Communications: The transmission of data between more than one site through the use of public and private communications channels or lines.

Data Contamination: A deliberate or accidental process or act that compromises the integrity of the original data.

Data Definition Language (DDL): A set of instructions or commands used to define data for the data dictionary. A data definition language (DDL) is used to describe the structure of a database.

Data Dictionary: A document or listing defining all items or processes represented in a data flow diagram or used in a system.

Data Element: The smallest unit of data accessible to a database management system or a field of data within a file processing system.

Data Encryption Standard (DES): A data encryption standard developed by the U.S. National Bureau of Standards (NBS). DES is a symmetric block cipher with a block length of 64 bits and an effective key length of 56 bits.

Data Integrity: The state that exists when automated information or data is the same as that in the source documents and has not been exposed to accidental or malicious modification, alteration, or destruction.

Data Item: A discrete representation having the properties that define the data element to which it belongs. *See also* data element.

Data Link: A serial communications path between nodes or devices without any intermediate switching nodes. Also, the physical two-way connection between such devices.

Data Manipulation Language (DML): A data manipulation language (DML) provides the necessary commands for all database operations, including storing, retrieving, updating, and deleting database records.

Data Objects: Objects or information of potential probative value that are associated with physical items. Data objects may occur in different formats without altering the original information.

Data Record: An identifiable set of data values treated as a unit, an occurrence of a schema in a database, or collection of atomic data items describing a specific object, event, or tuple (e.g., row of a table).

Data Security: The protection of data from accidental or malicious modification, destruction, or disclosure.

Data Set: A named collection of logically related data items, arranged in a prescribed manner and described by control information to which the programming system has access.

Data Warehouse: A collection of integrated subject-oriented databases designed to support the Decision Support function, where each unit of data is relevant to some moment in time. The data warehouse contains atomic data and summarized data.

Database: An integrated aggregation of data usually organized to reflect logical or functional relationships among data elements.

Database Administrator (DBA): (1) A person who is in charge of defining and managing the contents of a database. (2) The individual in an organization who is responsible for the daily monitoring and maintenance of the databases. The database administrator's function is more closely associated with physical database design than the data administrator's function is.

Database Management System (DBMS): The software that directs and controls data resources.

Datagram: Logical grouping of information sent as a network layer unit over a transmission medium without prior establishment of a virtual circuit. IP datagrams are the primary information units in the Internet. The terms "cell," "frame," "message," "packet," and "segment" are also used to describe logical information groupings at various layers of the OSI Reference Model and in various technology circles.

Data-Link Control Layer: Layer 2 in the SNA architectural model. Responsible for the transmission of data over a particular physical link. Corresponds roughly to the data-link layer of the OSI model.

Data-Link Layer: Layer 2 of the OSI reference model. Provides reliable transit of data across a physical link. The data-link layer is concerned with physical addressing, network topology, line discipline, error notification, ordered delivery of frames, and flow control. The IEEE divided this layer into two sublayers: the MAC sublayer and the LLC sublayer. Sometimes simply called the link layer. Roughly corresponds to the data-link control layer of the SNA model.

Deadlock: A condition that occurs when two users invoke conflicting locks in trying to gain access to a specific record or records.

Decipher: The ability to convert, by use of the appropriate key, enciphered text into its equivalent plaintext.

Decrypt: Synonymous with decipher.

Decrypt/Decipher/Decode: Decryption is the opposite of encryption. It is the transformation of encrypted information back into a legible form. Essentially, decryption is about removing disguise and reclaiming the meaning of information.

Decryption: The conversion through mechanisms or procedures of encrypted data into its original form.

Dedicated Lines: Private circuits between two or more stations, switches, or subscribers.

Dedicated Mode: The operation of a computer system such that the central computer facility, connected peripheral devices, communications facilities, and all remote terminals are used and controlled exclusively by the users or groups of users for the processing of particular types and categories of information.

Degauss: To erase or demagnetize magnetic recording media (usually tapes) by applying a variable, alternating current (AC) field.

Degree (of a relation): The number of attributes or columns of a relation.

Delegation: The notation that an object can issue a request to another object in response to a request. The first object therefore delegates the responsibility to the second object. Delegation can be used as an alternative to inheritance.

Delphi: A forecasting method where several knowledgeable individuals make forecasts and a forecast is derived by a trained analyst from a weighted average.

Demodulation: The reconstruction of an original signal from the modulated signal received at a destination device.

Denial-of-Service (DoS): Attacks on systems written to deny users legitimate access to system resources. These attacks take many forms, but are primarily applications or malicious applets that take more processes or memory allocation area than they should use, such as filling up a file system or allocating all of a system's memory.

Denial-of-service (DoS) attacks are among the most commonly encountered Java security concerns and, unfortunately, Java has a weak defense against it.

Design: The aspect of the specification process that involves the prior consideration of the implementation. Design is the process that extends and modifies an analysis specification. It accommodates certain qualities including extensibility, reusability, testability, and maintainability. Design also includes the specification of implementation requirements such as user interface and data persistence.

Design and Implementation: A phase of the systems development life cycle in which a set of functional specifications produced during systems analysis is transformed into an operational system for hardware, software, and firmware.

Design Review: The quality assurance process in which all aspects of a system are reviewed publicly.

Dial-Up: Access to switched network, usually through a dial or push-button telephone.

Digital: A mode of transmission where information is coded in binary form for transmission on the network.

Digital Audio Tape (DAT): A magnetic tape technology. DAT uses 4-mm cassettes capable of backing up anywhere between 26 and 126 bytes of information.

Digital Signature: The act of electronically affixing an encrypted message digest to a computer file or message in which the originator is then authenticated to the recipient.

Digital Signature Standard (DSS): The National Security Administration's standard for verifying an electronic message.

Direct Access: The method of reading and writing specific records without having to process all preceding records in a file.

Direct Access Storage Device (DASD): A data storage unit on which data can be accessed directly without having to progress through a serial file such as a magnetic tape file. A disk unit is a direct access storage device.

Directory: A table specifying the relationships between items of data. Sometimes a table (index) giving the addresses of data.

Discrepancy Reports: A listing of items that have violated some detective control and require further investigation.

Disk Duplexing: This refers to the use of two controllers to drive a disk subsystem. Should one of the controllers fail, the other is still available for disk I/O. Software applications can take advantage of both controllers to simultaneously read and write to different drives.

Disk Mirroring: Disk mirroring protects data against hardware failure. In its simplest form, a two-disk subsystem would be attached to a host controller. One disk serves as the mirror image of the other. When data is written to it, it is also written to the other disk. Both disks

will contain exactly the same information. If one fails, the other can supply the user data without problem.

Disk Operating System (DOS): Software that controls the execution of programs and may provide system services as resource allocation.

Diskette: A flexible disk storage medium most often used with microcomputers; also called a floppy disk.

Distributed Component Object Model (DCOM): A protocol that enables software components to communicate directly over a network. Developed by Microsoft and previously called "Network OLE," DCOM is designed for use across multiple network transports including Internet Protocols such as HTTP.

Distributed Computing: The distribution of processes among computing components that are within the same computer or different computers on a shared network.

Distributed Database: A database management system with the ability to effectively manage data that is distributed across multiple computers on a network.

Distributed Environment: A set of related data processing systems in which each system has its own capacity to operate autonomously but has some applications that are executed at multiple sites. Some of the systems may be connected with teleprocessing links into a network with each system serving as a node.

Domain Name: The name used to identify an Internet host.

Domain Name Server: An Internet host that checks the addresses of incoming and outgoing Internet messages.

Domain Name System (DNS): The distributed name and address mechanism used in the Internet.

Dumb Terminal: A device used to interact directly with the end user where all data is processed on a remote computer. A dumb terminal only gathers and displays data; it has no processing capability.

Dump: The contents of a file or memory that are output as listings. These listing can be formatted.

Duplex: Communications systems or equipment that can simultaneously carry information in both directions between two points. Also used to describe redundant equipment configurations (e.g., duplexed processors).

Early Token Release: Technique used in Token Ring networks that allows a station to release a new token onto the ring immediately after transmitting, instead of waiting for the first frame to return. This feature can increase the total bandwidth on the ring. *See also* Token Ring.

Earth Stations: Ground terminals that use antennas and other related electronic equipment designed to transmit, receive, and process satellite communications.

Eavesdropping: The unauthorized interception of information-bearing emanations through methods other than wiretapping.

Echo: The display of characters on a terminal output device as they are entered into the system.

Edit: The process of inspecting a data field or element to verify the correctness of its content.

Electromagnetic Emanations: Signals transmitted as radiation through the air or conductors.

Electromagnetic Interference (EMI): Electromagnetic waves emitted by a device.

Electronic Code Book (ECB): A basic encryption method that provides privacy but not authentication.

Electronic Communications Privacy Act of 1986 PL 99-508 (ECPA): Electronic Communications Privacy Act of 1986; extends the Privacy Act of 1974 to all forms of electronic communication, including email.

Electronic Data Interchange (EDI): A process whereby such specially formatted documents as an invoice can be transmitted from one organization to another.

Electronic Data Vaulting: Electronic vaulting protects information from loss by providing automatic and transparent backup of valuable data over high-speed phone lines to a secure facility.

Electronic Frontier Foundation: A foundation established to address social and legal issues arising from the impact on society of the increasingly pervasive use of computers as the means of communication and information distribution.

Electronic Funds Transfer (EFT): The process of moving money between accounts via computer.

Electronic Journal: A computerized log file summarizing, in chronological sequence, the processing activities and events performed by a system. The log file is usually maintained on magnetic storage media.

Emanation Security: The protection that results from all measures designed to deny unauthorized persons access to valuable information that might be derived from interception and analysis of compromising emanations.

Encrypt/Encipher/Encode: Encryption is the transformation of information into a form that is impossible to read unless you have a specific piece of information, which is usually referred to as the "key." The purpose is to keep information private from those who are not intended to have access to it. To encrypt is essentially about making information confusing and hiding the meaning of it.

Encryption: The use of algorithms to encode data in order to render a message or other file readable only for the intended recipient.

Encryption Algorithm: A set of mathematically expressed rules for encoding information, thereby rendering it unintelligible to those who do not have the algorithm decoding key.

Encryption Key: A special mathematical code that allows encryption hardware/software to encode and then decipher an encrypted message.

End System: An OSI system that contains application processes capable of communication through all seven layers of OSI protocols. Equivalent to Internet host.

End-to-End Encryption: The encryption of information at the point of origin within the communications network and postponing of decryption to the final destination point.

Enrollment: The initial process of collecting biometric data from a user and then storing it in a template for later comparison.

Entrapment: The deliberate planting of apparent flows in a system to invite penetrations.

Error: A discrepancy between actual values or conditions and those expected.

Error Rate: A measure of the quality of circuits or equipment. The ratio of erroneously transmitted information to the total sent (generally computed per million characters sent).

Espionage: The practice or employment of spies; the practice of watching the words and conduct of others, to make discoveries, as spies or secret emissaries; secret watching. This category of computer crime includes international spies and their contractors who steal secrets from defense, academic, and laboratory research facility computer systems. It includes criminals who steal information and intelligence from law enforcement computers, and industrial espionage agents who operate for competitive companies or for foreign governments who are willing to pay for the information. What has generally been known as industrial espionage is now being called competitive intelligence. A lot of information can be gained through "open source" collection and analysis without ever having to break into a competitor's computer. This information gathering is also competitive intelligence, although it is not as ethically questionable as other techniques.

Ethernet: A 10-Mbps standard for LANs, initially developed by Xerox and later refined by Digital, Intel, and Xerox (DIX). All hosts are connected to a coaxial cable where they contend for network access using a Carrier Sense Multiple Access with Collision Detection (CSMA/CD) paradigm.

Exception Report: A manager report that highlights abnormal business conditions. Usually, such reports prompt management action or inquiry.

Expert System: The application of computer-based artificial intelligence in areas of specialized knowledge.

Extensibility: A property of software such that new kinds of object or functionality can be added to it with little or no effect to the existing system.

eXtensible Markup Language (XML): Designed to enable the use of SGML on the World Wide Web, XML is a regular markup language that defines what you can do (or what you have done) in the way of describing information for a fixed class of documents (like HTML). XML goes beyond this and allows you to define your own customized markup language. It can do this because it is an application profile of SGML. XML is a metalanguage, a language for describing languages.

Fail Safe: The automatic termination and protection of programs or other processing operations when a hardware, software, or firmware failure is detected in a computer system.

Fail Soft: The selective termination of nonessential processing affected by a hardware, software, or firmware failure in a computer system.

Fallback Procedures: Predefined operations (manual or automatic) invoked when a fault or failure is detected in a system.

False Acceptance Rate (FAR): The percentage of imposters incorrectly matched to a valid user's biometric. False rejection rate (FRR) is the percentage of incorrectly rejected valid users.

Fast Ethernet: Any of a number of 100-Mbps Ethernet specifications. Fast Ethernet offers a speed increase ten times that of the 10BaseT Ethernet specification, while preserving such qualities as frame format, MAC mechanisms, and MTU. Such similarities allow the use of existing 10BaseT applications and network management tools on Fast Ethernet networks. Based on an extension to the IEEE 802.3 specification. *Compare with* Ethernet.

Fetch Protection: A system-provided restriction to prevent a program from accessing data in another user's segment of storage.

Fiber Distributed Data Interface (FDDI): LAN standard, defined by ANSI X3T9.5, specifying a 100-Mbps token-passing network using fiber-optic cable, with transmission distances of up to 2 km. FDDI uses a dual-ring architecture to provide redundancy.

Field Definition Record (FDR): A record of field definition. A list of the attributes that define the type of information that can be entered into a data field.

File Protection: The aggregate of all processes and procedures established in a computer system and designed to inhibit unauthorized access, contamination, or elimination of a file.

File Transfer: The process of copying a file from one computer to another over a network.

File Transfer Protocol (FTP): The Internet protocol (and program) used to transfer files between hosts.

Filter: A process or device that screens incoming information for definite characteristics and allows a subset of that information to pass through.

Finger: A program (and a protocol) that displays information about a particular user, or all users, logged on a local system or on a remote system. It typically shows full-time name, last login time, idle time, terminal line, and terminal location (where applicable). It may also display plan and project files left by the user.

Firewall: A system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both. Firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially *intranets*. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria. There are several types of firewall techniques:

- *Packet filter:* Looks at each packet entering or leaving the network and accepts or rejects it based on user-defined rules. Packet filtering is fairly effective and transparent to users, but it is difficult to configure. In addition, it is susceptible to IP spoofing.
- *Application gateway:* Applies security mechanisms to specific applications, such as FTP and Telnet servers. This is very effective, but can impose performance degradation.
- *Circuit-level gateway:* Applies security mechanisms when a TCP or UDP connection is established. Once the connection has been made, packets can flow between the hosts without further checking.
- *Proxy server:* Intercepts all messages entering and leaving the network. The proxy server effectively hides the true network addresses.

Firmware: Software or computer instructions that have been permanently encoded into the circuits of semiconductor chips.

Flame: To express strong opinion or criticism of something, usually as a frank inflammatory statement in an electronic message.

Flat File: A collection of records containing no data aggregates, nested, or repeated data items, or groups of data items.

Foreign Corrupt Practices Act: The act covers an organization's system of internal accounting control and requires public companies to make and keep books, records, and accounts that, in reasonable detail, accurately and fairly reflect the transactions and disposition of company assets and to devise and maintain a system of sufficient internal accounting controls. This act was amended in 1988.

Formal Review: A type of review typically scheduled at the end of each activity or stage of development to review a component of a deliverable or, in some cases, a complete deliverable or the software product and its supporting documentation.

Format: The physical arrangement of data characters, fields, records, and files.

Forum of Incident Response and Security Teams (FIRST): A unit of the Internet Society that coordinates the activities of worldwide Computer Emergency Response Teams, regarding security-related incidents and information sharing on Internet security risks.

Fragment: A piece of a packet. When a router is forwarding an IP packet to a network with a Maximum Transmission Unit smaller than the packet size, it is forced to break up that packet into multiple fragments. These fragments will be reassembled by the IP layer at the destination host.

Fragmentation: The process in which an IP datagram is broken into smaller pieces to fit the requirements of a given physical network. The reverse process is termed "reassembly."

Frame Relay: A switching interface that operates in packet mode. Generally regarded as the replacement for X.25.

Front-End Computer: A computer that offloads input and output activities from the central computer so it can operate primarily in a processing mode; sometimes called a front-end processor.

Front-End Processor (FEP): (1) A communications computer associated with a host computer can perform line control, message handling, code conversion, error control, and application functions. (2) A teleprocessing concentrator and router, as opposed to a back-end processor or a database machine.

Front Porch: The access point to a secure network environment; also known as a firewall.

Full-Duplex (FDX): An asynchronous communications protocol that allows the communications channel to transmit and receive signals simultaneously.

Fully Qualified Domain Name (FQDN): A complete Internet address, including the complete host and domain name.

Function: In computer programming, a processing activity that performs a single identifiable task.

Functional Specification: The main product of systems analysis, which presents a detailed logical description of the new system. It contains sets of input, processing, storage, and output requirements specifying what the new system can do.

Garbage Collection: A language mechanism that automatically deallocates memory for objects that are not accessible or referenced.

Gateway: A product that enables two dissimilar networks to communicate or interface with each other. In the IP community, an older term referring to a routing device. Today, the term "router" is used to describe nodes that perform this function, and "gateway" refers to a special-purpose device that performs an application layer conversion

of information from one protocol stack to another. *Compare with* router.

General-Purpose Computer: A computer that can be programmed to perform a wide variety of processing tests.

Government OSI Profile (GOSIP): A U.S. Government procurement specification for OSI protocols.

Granularity: The level of detail contained in a unit of data. The more there is, the lower the level of granularity; the less detail, the higher the level of granularity.

Graphical User Interface (GUI): An interface in which the user can manipulate icons, windows, pop-down menus, or other related constructs. A graphical user interface uses graphics such as a window, box, and menu to allow the user to communicate with the system. Allows users to move in and out of programs and manipulate their commands using a pointing device (usually a mouse). *Synonymous with* user interface.

Groupware: Software designed to function over a network to allow several people to work together on documents and files.

Hacker: A person who attempts to break into computers that he or she is not authorized to use.

Hacking: A computer crime in which a person breaks into an information system simply for the challenge of doing so.

Half-Duplex: Capability for data transmission in only one direction at a time between a sending station and a receiving station.

Handprint Character Recognition (HCR): One of several pattern recognition technologies used by digital imaging systems to interpret handprinted characters.

Handshake: Sequence of messages exchanged between two or more network devices to ensure transmission synchronization.

Handshaking Procedure: Dialogue between a user and a computer, two computers, or two programs to identify a user and authenticate his or her identity. This is done through a sequence of questions and answers that are based on information either previously stored in the computer or supplied to the computer by the initiator of the dialogue.

Hard Disk: A fixed or removable disk mass storage system permitting rapid direct access to data, programs, or information.

Hash Total: A total of the values on one or more fields, used for the purpose of auditability and control.

HDLC (High-Level Data-Link Control): Bit-oriented synchronous data-link layer protocol developed by ISO. Derived from SDLC, HDLC specifies a data encapsulation method on synchronous serial links using frame characters and checksums.

HDSL: High-data-rate digital subscriber line. One of four DSL technologies. HDSL delivers 1.544 Mbps of bandwidth each way over two copper

twisted pairs. Because HDSL provides T1 speed, telephone companies have been using HDSL to provision local access to T1 services whenever possible. The operating range of HDSL is limited to 12,000 feet (3658.5 meters), so signal repeaters are installed to extend the service. HDSL requires two twisted pairs, so it is deployed primarily for PBX network connections, digital loop carrier systems, interexchange POPs, Internet servers, and private data networks. *Compare with* ADSL, SDSL, and VDSL.

Header: The beginning of a message sent over the Internet; typically contains addressing information to route the message or packet to its destination.

Hertz (Hz): One cycle per second.

Heuristics The mode of analysis in which the next step is determined by the results of the current step of analysis. Used for decision support processing.

Hexadecimal: A number system with a base of 16.

Hierarchical Database: In a hierarchical database, data is organized like a family tree or organization chart with branches of parent records and child records.

High-Level Data-Link Control (HDLC): A protocol used at the data-link layer that provides point-to-point communications over a physical transmission medium by creating and recognizing frame boundaries.

High-Level Language: The class of procedure-oriented language.

Home Page: The initial screen of information displayed to the user when initiating the client or browser software or when connecting to a remote computer. The home page resides at the top of the directory tree.

Hop: A term used in routing. A hop is one data link. A path from source to destination in a network is a series of hops.

Host: A remote computer that provides a variety of services, typically to multiple users concurrently.

Host Address: The IP address of the host computer.

Host Computer: A computer that, in addition to providing a local service, acts as a central processor for a communications network.

Hostname: The name of the user computer on the network.

HTML: *See* HyperText Markup Language.

HTTP: *See* HyperText Transport Protocol.

Hub: A device connected to several other devices. In ARCnet, a hub is used to connect several computers together. In a message-handling service, a hub is used for transfer of messages across the network. An Ethernet hub is basically a "collapsed network-in-a-box" with a number of ports for the connected devices.

Hypertext: Text that is held in frames and authors develop or define the linkage between frames.

HyperText Markup Language (HTML): The specialized language used to insert formatting commands and links in a hypertext document.

HyperText Transfer Protocol (HTTP): The protocol used to transport hypertext files across the Internet.

IAB: Internet Architecture Board. Board of internetwork researchers who discuss issues pertinent to Internet architecture. Responsible for appointing a variety of Internet-related groups such as the IANA, IESG, and IRSG. The IAB is appointed by the trustees of the ISOC.

ICMP: Internet Control Message Protocol. Network layer Internet protocol that reports errors and provides other information relevant to IP packet processing. Documented in RFC 792.

Icon: A pictorial symbol used to represent data, information, or a program on a GUI screen.

Identification: The process, generally employing unique machine-readable names, that enables recognition of users or resources as identical to those previously described to the computer system.

Impersonation: An attempt to gain access to a system by posing as an authorized user.

Implementation: The specific activities within the systems development life cycle through which the software portion of the system is developed, coded, debugged, tested, and integrated with existing or new software.

Incident: An event that has actual or potentially adverse effects on an information system. A computer security incident can result from a computer virus, other malicious code, intruder, terrorist, unauthorized insider act, malfunction, etc.

Incomplete Parameter Checking: A system fault that exists when all parameters have not been fully checked for correctness and consistency by the operating system, thus leaving the system vulnerable to penetration.

Inference Engine: A system of computer programs in an expert systems application that uses expert experience as a basis for conclusions.

Infobots: Software agents that perform specified tasks for a user or application.

Information Security Service: A method to provide some specific aspect of security. For example, integrity of transmitted data is a security objective, and a method that would achieve that is considered an information security service.

Inheritance: The language mechanism that allows the definition of a class to include the attributes and methods for another more general class. Inheritance is an implementation construct for the specialization relation. The general class is the superclass and the specific class is the subclass in the inheritance relation. Inheritance is a relation between classes that enables the reuse of code and the definition of generalized interface to one or more subclasses.

Input Controls: Techniques and methods for verifying, validating, and editing data to ensure that only correct data enters a system.

Instance: A set of values representing a specific entity belonging to a particular entity type. A single value is also the instance of a data item.

Integrated Data Dictionary (IDD): A database technology that facilitates functional communication among system components.

Integrated Services Digital Network (ISDN): An emerging technology that is beginning to be offered by the telephone carriers of the world. ISDN combines voice and digital network services in a single medium, making it possible to offer customers digital data services as well as voice connections through a single wire. The standards that define ISDN are specified by ITU-TSS.

Integrity: *See also* data integrity. A security service that allows verification that an unauthorized modification (including changes, insertions, deletions and duplications) has not occurred either maliciously or accidentally.

Integrity Checking: The testing of programs to verify the soundness of a software product at each phase of development.

Interactive: A mode of processing that combines some aspects of online processing and some aspects of batch processing. In interactive processing, the user can directly interact with data over which he or she has exclusive control. In addition, the user can cause sequential activity to initiate background activity to be run against the data.

Interface: A shared boundary between devices, equipment, or software components defined by common interconnection characteristics.

Interleaving: The alternating execution of programs residing in the memory of a multiprogramming environment.

Internal Control: The method of safeguarding business assets, including verifying the accuracy and reliability of accounting data, promoting operational efficiency, and encouraging adherence to prescribed organizational policies and procedures.

Internet: The Internet consists of large national backbone networks (such as MILNET, NSFNET, and CREN) and a myriad of regional and local campus networks all over the world. The Internet uses the Internet Protocol suite. To be on the Internet, you must have IP connectivity (i.e., be able to Telnet to — or ping — other systems). Networks with only email connectivity are not actually classified as being on the Internet.

Internet Address: A 32-bit address assigned to hosts using TCP/IP.

Internet Architecture Board (IAB): Formally called the Internet Activities Board. The technical body that oversees the development of the Internet suite of protocols (commonly referred to as "TCP/IP"). It has two task forces (the IRTF and the IETF), each charged with investigating a particular area.

Internet Assigned Numbers Authority (IANA): A largely government-funded overseer of IP allocations chartered by the FNC and the ISOC.

Internet Control Message Protocol (ICMP): The protocol used to handle errors and control messages at the IP layer. ICMP is actually part of the IP.

Internet Engineering Task Force (IETF): An open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet's architecture; established by the IAB.

Internet Layer: The stack in the TCP/IP protocols that addresses a packet and sends the packets to the network access layer.

Internet Message Access Protocol (IMAP): A method of accessing electronic mail or bulletin board messages that are kept on a (possibly shared) mail server. IMAP permits a "client" email program to access remote message stores as if they were local. For example, email stored on an IMAP server can be manipulated from a desktop computer at home, a workstation at the office, and a notebook computer while traveling, without the need to transfer messages of files back and forth between these computers. IMAP can be regarded as the next-generation POP.

Internet Protocol (IP, IPv4): The Internet Protocol (version 4), defined in RFC 791, is the network layer for the TCP/IP suite. It is a connectionless, best-effort, packet-switching protocol.

Internet Protocol (Ping, IPv6): IPv6 is a new version of the Internet Protocol that is designed to be evolutionary.

Internet Service Provider (ISP): An organization that provides direct access to the Internet, such as the provider that links your college or university to the Net.

Interoperability: The ability to exchange requests between entities. Objects interoperate if the methods that apply to one object can request services of another object.

Investigation: The phase of the systems development life cycle in which the problem or need is identified and a decision is made on whether to proceed with a full-scale study.

IP Address: A unique number assigned to each computer on the Internet, consisting of four numbers, each less than 256, and each separated by a period, such as 129.16.255.0.

IP Datagram: The fundamental unit of information passed across the Internet. Contains source and destination addresses, along with data and a number of fields that define such things as the length of the datagram, the header checksum, and flags to say whether the datagram can be (or has been) fragmented.

ISO 9000: A certification program that demonstrates an organization adheres to steps that ensure quality of goods and services. A quality

series that comprises a set of five documents and was developed in 1987 by the International Standards Organization (ISO).

Isolation: The separation of users and processes in a computer system from one another, as well as from the protection controls of the operating system.

Iterative Development Life Cycle: A strategy for developing systems that allows for the controlled reworking of parts of a system to remove mistakes or to make improvements based on feedback.

Java: Object-oriented programming language developed at Sun Microsystems to solve a number of problems in modern programming practice. The Java language is used extensively on the World Wide Web, particularly for applets.

Join: An operation that takes two relations as operand and produces a new relation by concealing the tuples and matching the corresponding columns when a stated condition holds between the two.

Jukebox: Hardware that houses, reads, and writes to many optical disks using a variety of mechanical methods for operation.

Kerberos: Developing standard for authenticating network users. Kerberos offers two key benefits: it functions in a multi-vendor network, and it does not transmit passwords over the network.

Key: In cryptography, a sequence of symbols that controls encryption and decryption.

Key/Cryptovariable: Encryption and decryption generally require the use of some secret information, referred to as a *key*. For some encryption mechanisms, the same key is used for both encryption and decryption; for other mechanisms, the keys used for encryption and decryption are different.

Key Generation: The origination of a key or set of distinct keys.

Key, Primary: A unique attribute used to identify a class of records in a database.

Knowledge Base: The part of an expert system that contains specific information and facts about the expert area. Rules that the expert system uses to make decisions are derived from this source.

L2F Protocol: Layer 2 Forwarding Protocol. Protocol that supports the creation of secure virtual private dial-up networks over the Internet.

Label: A set of symbols used to identify or describe an item, record, message, or file.

LAN: Local Area Network. High-speed, low-error data network covering a relatively small geographic area (up to a few thousand meters). LANs connect workstations, peripherals, terminals, and other devices in a single building or other geographically limited area. LAN standards specify cabling and signaling at the physical and data-link layers of the OSI model. Ethernet, FDDI, and Token Ring are widely used LAN technologies. *Compare with* MAN and WAN.

LAN Switch: High-speed switch that forwards packets between data-link segments. Most LAN switches forward traffic based on MAC addresses. This variety of LAN switch is sometimes called a frame switch. LAN switches are often categorized according to the method they use to forward traffic: cut-through packet switching or store-and-forward packet switching. Multi-layer switches are an intelligent subset of LAN switches. *Compare with* multi-layer switch. *See also* cut-through packet switching and store-and-forward packet switching.

Language Translator: Systems software that converts programs written in assembler or a higher-level language into machine code.

Laser: Light Amplification by Stimulated Emission of Radiation. Analog transmission device in which a suitable active material is excited by an external stimulus to produce a narrow beam of coherent light that can be modulated into pulses to carry data. Networks based on laser technology are sometimes run over SONET.

Laser Printer: An output unit that uses intensified light beams to form an image on an electrically charged drum and then transfers the image to paper.

Latency: In local networking, the time (measured in bits at the transmission rate) for a signal to propagate around or throughput the network. The time taken by a DASD device to position a storage location to reach the read arm over the physical storage medium. For general purposes, average latency time is used. Delay between the time a device requests access to a network and the time it is granted permission to transmit.

Layer 3 Switching: The emerging layer 3 switching technology integrates routing with switching to yield very high routing throughput rates in the millions-of-packets-per-second range. The movement to layer 3 switching is designed to address the downsides of the current generation of layer 2 switches, which are functionally equivalent to bridges. These downsides for a large, flat network include being subject to broadcast storms, spanning tree loops, and address limitations that drove the injection of routers into bridged networks in the late 1980s. Currently, layer 3 switching is represented by a number of approaches in the industry.

LDAP: Lightweight Directory Access Protocol. Protocol that provides access for management and browser applications that provide read/write interactive access to the X.500 Directory.

Leased Line: An un-switched telecommunications channel leased to an organization for its exclusive use.

Least Recently Used (LRU): A replacement strategy in which new data must replace existing data in an area of storage; the least recently used items are replaced.

Lightweight Directory Access Protocol (LDAP): This protocol provides access for management and browser application that provide read/write interactive access to the X.500 Directory.

Limit Check: An input control text that assesses the value of a data field to determine whether values fall within set limits.

Line Conditioning: A service offered by common carriers to reduce delay, noise, and amplitude distortion to produce transmission of higher data speeds.

Line Printer: An output unit that prints alphanumeric characters one line at a time.

Line Speed: The transmission rate of signals over a circuit, usually expressed in bits per second.

Load Sharing: A multiple-computer system that shares the load during peak hours. During non-peak periods or standard operation, one system can handle the entire load with the others acting as fallback units.

Logging: The automatic recording of data for the purpose of accessing and updating it.

MAC (Media Access Control): Lower of the two sub-layers of the data-link layer defined by the IEEE. The MAC sub-layer handles access to shared media, such as whether token passing or contention will be used.

MAC Address: Standardized data-link layer address that is required for every port or device that connects to a LAN. Other devices in the network use these addresses to locate specific ports in the network and to create and update routing tables and data structures. MAC addresses are 6 bytes long and are controlled by the IEEE. Also known as a hardware address, MAC-layer address, and physical address. *Compare with* network address.

Machine Language: Computer instructions or code representing computer operations and memory addresses in a numeric form that is executable by the computer without translation.

Maintenance: Tasks associated with the modification or enhancement of production software.

Maintenance Programmer: An applications programmer responsible for making authorized changes to one or more computer programs and ensuring that the changes are tested, documented, and verified.

Masquerade: A type of security threat that occurs when an entity successfully pretends to be a different entity.

Media: The various physical forms (e.g., disk, tape, and diskette) on which data is recorded in machine-readable formats.

Media Access Control (MAC): A local network control protocol that governs station access to a shared transmission medium. Examples are token passing and CSMA. *See* carrier sense, multiple access.

Megabyte (Mbyte, MB): The equivalent of 1,048,576 bytes.

Memory: The area in a computer that serves as temporary storage for programs and data during program execution.

Memory Address: The location of a byte or word of storage in computer memory.

Memory Bounds: The limits in the range of storage addresses for a protected region in memory.

Memory Chips: A small integrated circuit chip with a semiconductor matrix used as computer memory.

Menu: A section of the computer program — usually the top-level module — that controls the order of execution of other program modules. Also, online options displayed to a user, prompting the user for specific input.

Message: The data input by the user in the online environment that is used to drive a transaction. The output of transaction.

Message Address: The information contained in the message header that indicates the destination of the message.

Metadata: The description of such things as the structure, content, keys, and indexes of data.

Metalanguage: A language used to specify other languages.

Metropolitan Area Network (MAN): A data network intended to serve an area approximating that of a large city. Such networks are being implemented by innovative techniques, such as running fiber cables through subway tunnels.

Microprocessor: A single small chip containing circuitry and components for arithmetic, logical, and control operations.

Middleware: The distributed software needed to support interactions between client and servers.

Minicomputer: Typically, a word-oriented computer whose memory size and processing speed falls between that of a microcomputer and a medium-sized computer.

Mirror Image Backup: Mirror image backups (also referred to as bitstream backups) involve the backup of all areas of a computer hard disk drive or another type of storage media (e.g., Zip disks, floppy disks, Jazz disks, etc.). Such mirror image backups exactly replicate all sectors on a given storage device. Thus, all files and ambient data storage areas are copied. Such backups are sometimes referred to as "evidence-grade" backups and they differ substantially from standard file backups and network server backups. The making of a mirror image backup is simple in theory, but the accuracy of the backup must meet evidence standards. Accuracy is essential and to guarantee accuracy, mirror image backup programs typically rely on mathematical CRC computations in the validation process. These mathematical validation processes compare the original source data with the restored data. When computer evidence is involved, accuracy is extremely important, and the making of a mirror image backup

is typically described as the preservation of the "electronic crime scene."

Mode of Operation: A classification for systems that execute in a similar fashion and share distinctive operational characteristics (e.g., Production, DSS, online, and Interactive).

Model: A representation of a problem or subject area that uses abstraction to express concepts.

Modem (Modulator/Demodulator): A device that converts the digital language of the PC to a series of high- and low-pitched tones for transmission over analog telephone lines.

Modification: A type of security threat that occurs when its content is modified in an unanticipated manner by a non-authorized entity.

Multiple Inheritance: The language mechanism that allows the definition of a class to include the attributes and methods defined for more than one superclass.

Multiprocessing: A computer operating method in which two or more processors are linked and execute multiple programs simultaneously.

Multiprogramming: A computer operating environment in which several programs can be placed in memory and executed concurrently.

Multi-purpose Internet Mail Extension (MIME): The standard for multimedia mail contents in the Internet suite of protocols.

NAK: Negative acknowledgment. Response sent from a receiving device to a sending device indicating that the information received contained errors. *Compare with* acknowledgment.

NAK Attack: A penetration technique that capitalizes on an operating system's inability to properly handle asynchronous interrupts.

Name Resolution: The process of mapping a name into the corresponding address.

NAT (Network Address Translation): Mechanism for reducing the need for globally unique IP addresses. NAT allows an organization with addresses that are not globally unique to connect to the Internet by translating those addresses into globally routable address space. *Also known as* Network Address Translator.

Need-to-Know: A security principle stating that an individual should have access only to that needed to perform a particular function.

Negative Acknowledgment (NAK): A response sent by the receiver to indicate that the previous block was unacceptable and the receiver is ready to accept a retransmission.

Network: An integrated, communicating aggregation of computers and peripherals linked through communications facilities.

Network Access Layer: The layer of the TCP/IP stack that sends the message out through the physical network onto the Internet.

Network Access Points (NAPs): (1) Nodes providing entry to the high-speed Internet backbone system. (2) Another name for an Internet Exchange Point.

Network Address: The network portion of an IP address. For a class A network, the network address is the first byte of the IP address. For a class B network, the network address is the first two bytes of the IP address. For a class C network, the network address is the first three bytes of the IP address. In the Internet, assigned network addresses are globally unique.

Network Administrator: The person who maintains user accounts, password files, and system software on your campus network.

Network Information Center (NIC): Originally, there was only one, located at SRI International and tasked to serve the ARPANET (and later DDN) community. Today, there are many NICs, operated by local, regional, and national networks all over the world. Such centers provided user assistance, document service, training, and much more.

Network Layer: The OSI layer that is responsible for routing, switching, and subnetwork access across the entire OSI environment.

Neural Network: A type of system developed by artificial intelligence researchers used for processing logic.

Node: A device attached to a network.

Noise: Random electrical signals introduced by circuit components or natural disturbances that tend to degrade the performance of a communications channel.

Object-Oriented: Any method, language, or system that supports object identity, classification, and encapsulation and specialization. C++, Smalltalk, Objective-C, and Eiffel are examples of object-oriented implementation languages.

Object-Oriented Analysis (OOA): The specification of requirements in terms of objects with identity that encapsulate properties and operations, messaging, inheritance, polymorphism, and binding.

Object-Oriented Database Management System (OODBMS): A database that stores, retrieves, and updates objects using transaction control, queries, locking, and versioning.

Object-Oriented Design (OOD): The development activity that specifies the implementation of a system using the conceptual model defined during the analysis phase.

Object-Oriented Language: A language that supports objects, method resolution, specialization, encapsulation, polymorphism, and inheritance.

Object Program: A program that has been translated from a higher-level source code into machine language.

Object Request Broker (ORB): A software mechanism by which objects make and receive requests and responses.

OLE: Microsoft's Object Linking and Embedding technology designed to let applications share functionality through live data exchange and embedded data. Embedded objects are packaged statically within the source application, called the "client;" linked objects launch the

"server" applications when instructed by the client application. Linking is the capability to call a program, embedding places data in a foreign program.

Online System: Applications that allow direct interaction of the user with the computer (CPU) via a CRT, thus enabling the user to receive back an immediate response to data entered (i.e., an airline reservation system). Only one root node can be used at the beginning of the hierarchical structure.

Open System: A system whose architecture permits components developed by independent organizations or vendors to be combined.

Open Systems Interconnection (OSI): An international standardization program to facilitate communications among computers from different manufactures. *See* ISO.

Operating System: The various sets of computer programs and other software that monitor and operate the computer hardware and the firmware to facilitate use of the hardware.

Optical Disk: A disk that is written to or read from by optical means.

Optical Fiber: A form of transmission medium that uses light to encode signals and has the highest transmission rate of any medium.

Optical Storage: A medium requiring lasers to permanently alter the physical media to create a permanent record. The storage also requires lasers to read stored information from this medium.

OSI Reference Model: The seven-layer architecture designed by OSI for open data communications network.

Overwriting: The obliteration of recorded data by recording different data on the same surface.

PABX: Private Automatic Branch Exchange. Telephone switch for use inside a corporation. PABX is the preferred term in Europe, while PBX is used in the United States.

Packet: Logical grouping of information that includes a header containing control information and (usually) user data. Packets are most often used to refer to network layer units of data. The terms "datagram," "frame," "message," and "segment" are also used to describe logical information groupings at various layers of the OSI Reference Model and in various technology circles.

Packet Internet Grouper (PING): A program used to test reachability of destinations by sending them an ICMP echo request and waiting for a reply. The term is used as a verb: "Ping host X to see if it is up."

Packet Switch: WAN device that routes packets along the most efficient path and allows a communications channel to be shared by multiple connections. Formerly called an Interface Message Processor (IMP).

Packet Switching: A switching procedure that breaks up messages into fixed-length units (called packets) at the message source. These units may travel along different routes before reaching their intended destination.

Padding: A technique used to fill a field, record, or block with default information (e.g., blanks or zeros).

Page: A basic unit of storage in main memory.

Page Fault: A program interruption that occurs when a page that is referred to is not in main memory and must be read from external storage.

Paging: A method of dividing a program into parts called pages and introducing a given page into memory as the processing on the page is required for program execution.

PAP (Password Authentication Protocol): Authentication protocol that allows PPP peers to authenticate one another. The remote router attempting to connect to the local router is required to send an authentication request. Unlike CHAP, PAP passes the password and hostname or username in the clear (unencrypted). PAP does not itself prevent unauthorized access, but merely identifies the remote end. The router or access server then determines if that user is allowed access. PAP is supported only on PPP lines. *Compare with* CHAP.

Parallel Port: The computer's printer port, which in a pinch, allows user access to notebooks and computers that cannot be opened.

Parent: A unit of data in a 1:n relationship with another unit of data called a child, where the parent can exist independently but the child cannot.

Parity: A bit or series of bits appended to a character or block of characters to ensure that the information received is the same as the information that was sent. Parity is used for error detection.

Parity Bit: A bit attached to a byte that is used to check the accuracy of data storage.

Partition: A memory area assigned to a computer program during its execution.

Passive Wiretapping: The monitoring or recording of data while it is being transmitted over a communications link.

Password: A word or string of characters that authenticates a user, a specific resource, or an access type.

Patent: Exclusive right granted to an inventor to produce, sell, and distribute the invention for a specified number of years.

Penetration: A successful unauthorized access to a computer system.

Penetration Testing: The use of special programmer or analyst teams to attempt to penetrate a system to identify security weaknesses.

Persistent Object: An object that can survive the process that created it. A persistent object exists until it is explicitly deleted.

Physical Layer: The OSI layer that provides the means to activate and use physical connections for bit transmission. In plain terms, the physical layer provides the procedures for transferring a single bit across a physical medium.

Piggyback Entry: Unauthorized access to a computer system that is gained through another user's legitimate connection.

Plain Old Telephone System (POTS): What we consider to be the "normal" phone system used with modems. Does not include leased lines or digital lines.

Plaintext: Intelligible text or signals that have meaning and can be read or acted on without being decrypted.

Platform: Foundation upon which processes and systems are built and which can include hardware, software, firmware, etc.

Point-of-Presence (POP): A site where there exists a collection of telecommunications equipment, usually digital leased lines and multi-protocol routers.

Point-to-Point: A network configuration interconnecting only two points. The connection can be dedicated or switched.

Point-to-Point Protocol (PPP): The successor to SLIP, PPP provides router-to-router and host-to-network connections over both synchronous and asynchronous circuits.

Pointer: The address of a record (or other data grouping) contained in another record so that a program may access the former record when it has retrieved the latter record. The address can be absolute, relative, or symbolic, and hence the pointer is referred to as absolute, relative, or symbolic.

Polling: A procedure by which a computer controller unit asks terminals and other peripheral devices in a serial fashion if they have any messages to send.

Polymorphism: A request-handling mechanism that selects a method based on the type of target object. This allows the specification of one request that can result in invocation of different methods depending on the type of the target object. Most object-oriented languages support the selection of the appropriate method based on the class of the object (classical polymorphism). A few languages or systems support characteristics of the object, including values and user-defined defaults (generalized polymorphism).

Port: An outlet, usually on the exterior of a computer system, that enables peripheral devices to be connected and interfaced with the computer.

Portability: The ability to implement and execute software in one type of computing space and have it execute in a different computing space with little or no changes.

Presentation Layer: The OSI layer that determines how application information is represented (i.e., encoded) while in transit between two end systems.

Pretty Good Privacy (PGP): PGP provides confidentiality and authentication services for electronic mail and file storage applications. Devel-

oped by Phil Zimmerman and distributed for free on the Internet. Widely used by the Internet technical community.

Primary Key: An attribute that contains values that uniquely identifies the record in which the key exists.

Principle of Least Privilege: A security procedure under which users are granted only the minimum access authorization they need to perform required tasks.

Privacy: The prevention of unauthorized access and manipulation of data.

Privacy Act of 1974: The federal law that allows individuals to know what information about them is on file and how it is used by all government agencies and their contractors. The 1986 Electronic Communication Act is an extension of the Privacy Act.

Privacy Enhanced Mail (PEM): Internet email standard that provides confidentiality, authentication, and message integrity using various encryption methods. Not widely deployed in the Internet.

Privacy Protection: The establishment of appropriate administrative, technical, and physical safeguards to protect the security and confidentiality of data records against anticipated threats or hazards that could result in substantial harm, embarrassment, inconvenience, or unfairness to any individual about whom such information is maintained.

Private Network: A network established and operated by a private organization for the benefit of members of the organization.

Privilege: A right granted to an individual, a program, or a process.

Privileged Instructions: A set of instructions generally executable only when the computer system is operating in the executive state (e.g., while handling interrupts). These special instructions are typically designed to control such protection features as the storage protection features.

Problem: Any deviation from predefined standards.

Problem Reporting: The method of identifying, tracking, and assigning attributes to problems detected within the software product, deliverables, or within the development processes.

Procedure: Required "how-to" instructions that support some part of a policy or standard.

Processor: The hardware unit containing the functions of memory and the central processing unit.

Program Development Process: The activities involved in developing computer programs, including problem analysis, program design, process design, program coding, debugging, and testing.

Program Maintenance: The process of altering program code or instructions to meet new or changing requirements.

Programmable Read-Only Memory (PROM): Computer memory chips that can be programmed permanently to carry out a defined process.

Programmer: The individual who designs and develops computer programs.

Programmer/Analyst: The individual who analyzes processing requirements and then designs and develops computer programs to direct processing.

Programming Language: A language with special syntax and style conventions for coding computer programs.

Programming Specifications: The complete description of input, processing, output, and storage requirements necessary to code a computer program.

Protection Ring: A hierarchy of access modes through which a computer system enforces the access rights granted to each user, program, and process, ensuring that each operates only within its authorized access mode.

Protocol: A set of instructions required to initiate and maintain communication between sender and receiver devices.

Protocol Analyzer: A data communications testing unit set that enables a network engineer to observe bit patterns and simulate network elements.

Prototype: A usable system or subcomponent that is built inexpensively or quickly with the intention of modifying or replacing it.

Public Key Encryption: An encryption scheme where two pairs of algorithmic keys (one private and one public) are used to encrypt and decrypt messages, files, etc.

Purging: The orderly review of storage and removal of inactive or obsolete data files.

Quality: The totality of features and characteristics of a product or service that bear on its ability to meet stated or implied needs.

Quality Assurance: An overview process that entails planning and systematic actions to ensure that a project is following good quality management practices.

Quality Control: Process by which product quality is compared with standards.

Quality of Service (QoS): The service level defined by a service agreement between a network user and a network provider, which guarantees a certain level of bandwidth and data flow rates.

Query Language: A language that enables a user to interact indirectly with a DBMS to retrieve and possibly modify data held under the DBMS.

RADIUS (Remote Dial-In User Service): Database for authenticating modem and ISDN connections and for tracking connection time.

RAID (Redundant Arrays of Inexpensive Disks): Instead of using one large disk to store data, you use many smaller disks (because they are cheaper). *See* disk mirroring and duplexing. An approach to using many low-cost drives as a group to improve performance, yet also

provides a degree of redundancy that makes the chance of data loss remote.

RAM: A type of computer memory that can be accessed randomly; that is, any byte of memory can be accessed without touching the preceding bytes. RAM is the most common type of memory found in computers and other devices, such as printers. There are two basic types of RAM: dynamic RAM (DRAM) and static RAM (SRAM).

RARP (Reverse Address Resolution Protocol): Protocol in the TCP/IP stack that provides a method for finding IP addresses based on MAC addresses. *Compare with* Address Resolution Protocol (ARP).

Read-Only Memory (ROM): Computer memory chips with preprogrammed circuits for storing such software as word processors and spreadsheets.

Reassembly: The process by which an IP datagram is "put back together" at the receiving hosts after having been fragmented in transit.

Recovery: The restoration of the information processing facility or other related assets following physical destruction or damage.

Recovery Procedures: The action necessary to restore a system's computational capability and data files after system failure or penetration.

Recursion: The definition of something in terms of itself. For example, a bill of material is usually defined in terms of itself.

Referential Integrity: The assurance that an object handle identifies a single object. The facility of a DBMS that ensures the validity of predefined relationships.

Regression Testing: The rerunning of test cases that a program has previously executed correctly to detect errors created during software correction or modification. Tests used to verify a previously tested system whenever it is modified.

Relational Database: In a relational database, data is organized in two-dimensional tables or relations.

Remanence: The residual magnetism that remains on magnetic storage media after degaussing.

Remote Access: The ability to dial into a computer over a local telephone number using a number of digital access techniques.

Remote Authentication Dial-In User Service (RADIUS): A security and authentication mechanism for remote access.

Remote Procedure Call (RPC): An easy and popular paradigm for implementing the client/server model of distributed computing. A request is sent to a remote system to execute a designated procedure, using arguments supplied, and the result returned to the caller.

Replay: A type of security threat that occurs when an exchange is captured and resent at a later time to confuse the original recipients

Replication: The process of keeping a copy of data through either shadowing or caching.

Report: Printed or displayed output that communicates the content of files and other activities. The output is typically organized and easily read.

Request for Comments (RFC): The document series, begun in 1969, that describes the Internet suite of protocols and related experiments. Not all (in fact, very few) RFCs describe Internet standards, but all Internet standards are written up as RFCs.

Residue: Data left in storage after processing operations and before degaussing or rewriting has occurred.

Resource: In a computer system, any function, device, or data collection that can be allocated to users or programs.

Risk: The probability that a particular security threat will exploit a particular vulnerability.

Risk Analysis: An analysis that examines an organization's information resources, its existing controls, and its remaining organization and computer system vulnerabilities. It combines the loss potential for each resource or combination of resources with an estimated rate of occurrence to establish a potential level of damage in dollars or other assets.

Risk Assessment: Synonymous with risk analysis.

Role: A job type defined in terms of a set of responsibilities.

Rollback: (1) Restoration of a system to its former condition after it has switched to a fallback mode of operation when the cause of the fallback has been removed. (2) The restoration of the database to an original position or condition often after major damage to the physical medium. (3) The restoration of the information processing facility or other related assets following physical destruction or damage.

Router: A system responsible for making decisions about which of several paths network (or Internet) traffic will follow. To do this, it uses a routing protocol to gain information about the network, and algorithms to choose the best route based on several criteria known as "routing metrics."

RSA: A public-key cryptographic system that may be used for encryption and authentication. It was invented in 1977 and named for its inventors: Ron Rivest, Adi Shamir, and Leonard Adleman.

Safeguard: Synonymous with control.

Sanitizing: The degaussing or overwriting of sensitive information in magnetic or other storage media.

Scalability: The likelihood that an artifact can be extended to provide additional functionality with little or no additional effort.

Scavenging: The searching of residue for the purpose of unauthorized data acquisition.

Scripts: Executable programs used to perform specified tasks for servers and clients.

Search Engine: A program written to allow users to search the Web for documents that match user-specified parameters.

Secrecy: A security principle that keeps information from being disclosed to anyone not authorized to access it.

Secure Electronic Transaction (SET): The SET specification has been developed to allow for secure credit card and offline debit card (check card) transactions over the World Wide Web.

Secure Operating System: An operating system that effectively controls hardware, software, and firmware functions to provide the level of protection appropriate to the value of the data resources managed by this operating system.

Secure Socket Layer (SSL): An encryption technology for the Web used to secure transactions such as the transmission of credit card numbers for e-commerce.

Security Audit: An examination of data security procedures and measures to evaluate their adequacy and compliance with established policy.

Security Controls: Techniques and methods to ensure that only authorized users can access the computer information system and its resources.

Security Filter: A set of software or firmware routines and techniques employed in a computer system to prevent automatic forwarding of specified data over unprotected links or to unauthorized persons.

Security Kernel: The central part of a computer system (hardware, software, or firmware) that implements the fundamental security procedures for controlling access to system resources.

Security Program: A systems program that controls access to data in files and permits only authorized use of terminals and other related equipment. Control is usually exercised through various levels of safeguards assigned on the basis of the user's need-to-know.

Seepage: The accidental flow, to unauthorized individuals, of data or information that is presumed to be protected by computer security safeguards.

Sensitive Data: Data that is considered confidential or proprietary. The kind of data that, if disclosed to a competitor, might give away an advantage.

Sensitive Information: Any information that requires protection and that should not be made generally available.

Serial Line Internet Protocol (SLIP): An Internet protocol used to run IP over serial lines such as telephone circuits or RS-232 cables interconnecting two systems. SLIP is now being replaced by Point-to-Point Protocol. *See* Point-to-Point Protocol.

Server: A computer that provides a service to another computer, such as a mail server, a file server, or a news server.

Session: A completed connection to an Internet service, and the ensuing connect time.

Session Layer: The OSI layer that provides means for dialogue control between end systems.

Shareware: Software available on the Internet that may be downloaded to your machine for evaluation and for which you are generally expected to pay a fee to the originator of the software if you decide to keep it.

Simple Network Management Protocol (SNMP): Provides remote administration of network device; "simple" because the agent requires minimal software.

Simulation: The use of an executable model to represent the behavior of an object. During testing, the computational hardware, the external environment, and even the coding segments may be simulated.

Simultaneous Processing: The execution of two or more computer program instructions at the same time in a multiprocessing environment.

Single Inheritance: The language mechanism that allows the definition of a class to include the attributes and methods defined for, at most, one superclass.

Social Engineering: An attack based on deceiving users or administrators at the target site.

Socket: A paring of an IP address and a port number. *See* port.

Softlifting: Illegal copying of licensed software for personal use.

Software: Computer programs, procedures, rules, and possibly documentation and data pertaining to the operation of the computer system.

Software Life Cycle: The period of time beginning when a software product is conceived and ending when the product is no longer available for use. The software life cycle is typically broken into phases (e.g., requirements, design, programming, testing, conversion, operations, and maintenance).

Software Maintenance: All changes, corrections, and enhancements that occur after an application has been placed into production.

Software Piracy: To illegally copy software.

Source Document: The form that is used for the initial recording of data prior to system input.

Source Program: The computer program that is coded in an assembler or higher-level programming language.

Spam: The act of posting the same information repeatedly on inappropriate places or too many places so as to overburden the network.

Specification: A description of a problem or subject that will be implemented in a computational or other system. The specification includes both a description of the subject and aspects of the implementation that affect its representation. Also, the process and analysis and design that results in a description of a problem or subject that can be implemented in a computation or other system.

Spoofing: The deliberate inducement of a user or resource to take incorrect action.

Spooling: A technique that maximizes processing speed through the temporary use of high-speed storage devices. Input files are transferred from slower, permanent storage and queued in the high-speed devices to await processing, or output files are queued in high-speed devices to await transfer to slower storage devices.

Standard: Mandatory statement of minimum requirements that support some part of a policy.

Standard Generalized Markup Language (SGML): An international standard for encoding textual information that specifies particular ways to annotate text documents separating the structure of the document from the information content. HTML is a generalized form of SGML.

State Transition: A change of state for an object; something that can be signaled by an event.

State Variable: A property or type that is part of an identified state of a given type.

Static Data: Data that, once established, remains constant.

Stream Cipher: An encryption method in which a cryptographic key and an algorithm are applied to each bit in a datastream, one bit at a time.

Structured Design: A methodology for designing systems and programs through a top-down, hierarchical segmentation.

Structured Programming: The process of writing computer programs using logical, hierarchical control structures to carry out processing.

Structured Query Language (SQL): The international standard language for defining and accessing a relational database.

Subnet: A portion of a network, which may be a physically independent network segment, that shares a network address with other portions of the network and is distinguished by a subnet number. A subnet is to a network what a network is to the Internet.

Subnet Address: The subnet portion of an IP address. In a subnetted network, the host portion of an IP address is split into a subnet and a host portion using an address (subnet) mask.

Subroutine: A segment of code that can be called up by a program and executed at any time from any point.

Superclass: A class from which another class inherits attributes and methods.

Swapping: A method of computer processing in which programs not actively being processed are held on special storage devices and alternated in and out of memory with other programs according to priority.

Synchronous: A protocol of transmitting data over a network where the sending and receiving terminals are kept in synchronization with each other by a clock signal embedded in the data.

Synchronous Optical NETwork (SONET): SONET is an international standard for high-speed data communications over fiber-optic media. The transmission rates range from 51.84 Mbps to 2.5 Gbps.

System: A series of related procedures designed to perform a specific task.

System Analysis: The process of studying information requirements and preparing a set of functional specifications that identify what a new or replacement system should accomplish.

System Design: The development of a plan for implementing a set of functional requirements as an operational system.

System Integrity Procedures: Procedures established to ensure that hardware, software, firmware, and data in a computer system maintain their state of original integrity and are not tampered with by unauthorized personnel.

System Log: An audit trail of relevant system happenings (e.g., transaction entries, database changes).

Systems Analysis: The process of studying information requirements and preparing a set of functional specifications that identify what a new or replacement system should accomplish.

Systems Design: The development of a plan for implementing a set of functional requirements as an operational system.

Systems Development Life Cycle (SDLC): (1) The classical operational development methodology that typically includes the phases of requirements gathering, analysis, design, programming, testing, integration, and implementation. (2) The systematic systems building process consisting of specific phases; for example, preliminary investigation, requirements determination, systems analysis, systems design, systems development, and systems implementation.

TACACS (Terminal Access Controller Access Control System): Authentication protocol, developed by the DDN community, that provides remote access authentication and related services, such as event logging. User passwords are administered in a central database rather than in individual routers, providing an easily scalable network security solution.

Technological Attack: An attack that can be perpetrated by circumventing or nullifying hardware, software, and firmware access control mechanisms rather than by subverting system personnel or other users.

Telecommunications: Any transmission, emission, or reception of signs, signals, writing, images, sounds, or other information by wire, radio, visual, satellite, or electromagnetic systems.

Teleprocessing: Information processing and transmission performed by an integrated system of telecommunications, computers, and person-to-machine interface equipment.

Teleprocessing Security: The protection that results from all measures designed to prevent deliberate, inadvertent, or unauthorized disclosure or acquisition of information stored in or transmitted by a teleprocessing system.

TEMPEST: The study and control of spurious electronic signals emitted from electronic equipment. TEMPEST is a classification of technology designed to minimize the electromagnetic emanations generated by computing devices. TEMPEST technology makes it difficult, if not impossible, to compromise confidentiality by capturing emanated information.

Test Data: Data that simulates actual data to form and content and is used to evaluate a system or program before it is put into operation.

Threat Monitoring: The analysis assessment and review of audit trails and other data collected to search out system events that may constitute violations or precipitate incidents involving data privacy.

Three-Way Handshake: The process whereby two protocol entities synchronize during connection establishment.

Throughput: The process of measuring the amount of work a computer system can handle within a specified timeframe.

Timestamping: The practice of tagging each record with some moment in time, usually when the record was created or when the record was passed from one environment to another.

Time-Dependent Password: A password that is valid only at a certain time of day or during a specified timeframe.

Token Passing: A network access method that uses a distinctive character sequence as a symbol (token), which is passed from node to node, indicating when to begin transmission. Any node can remove the token, begin transmission, and replace the token when it is finished.

Token Ring: A type of area network in which the devices are arranged in a virtual ring in which the devices use a particular type of message called a token to communicate with one another.

Traceroute: A program available on many systems that traces the path a packet takes to a destination. It is mostly used to debug routing problems between hosts. There is also a traceroute protocol defined in RFC 1393.

Trademark: A registered word, letter, or device granting the owner exclusive rights to sell/distribute the goods to which it is applied.

Traffic Analysis: A type of security threat that occurs when an outside entity is able to monitor and analyze traffic patterns on a network.

Traffic Flow Security: The protection that results from those features in some cryptography equipment that conceal the presence of valid messages on a communications circuit, usually by causing the circuit to appear busy at all times or by encrypting the source and destination addresses of valid messages.

Transaction: A transaction is an activity or request to a computer. Purchase orders, changes, additions, and deletions are examples of transactions that are recorded in a business information environment.

Transmission Control Protocol (TCP): The major transport protocol in the Internet suite of protocols providing reliable, connection-oriented, full-duplex streams.

Transport Layer: The OSI layer that is responsible for reliable end-to-end data transfer between end systems.

Trojan Horse: A computer program that is apparently or actually useful and contains a trapdoor or unexpected code.

Trust: Reliance on the ability of a system or process to meet its specifications.

Trusted Computer Security Evaluation Criteria (TCSEC): A security development standard for system manufacturers and a basis for comparing and evaluating different computer systems. Also known as the *Orange Book*.

Tunneling: Tunneling refers to encapsulation of protocol A with protocol B, such that A treats B as though it were a data-link layer. Tunneling is used to get data between administrative domains, which use a protocol that is not supported by the internet connecting those domains.

Twisted Pair: A type of network physical medium made of copper wires twisted around each other. Example: Ordinary telephone cable.

UDP: User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768.

Uniform Resource Locator (URL): The primary means of navigating the web; consists of the means of access, the Web site, the path, and the document name of a Web resource, such as http://www.auerbach-publications.com.

Unshielded Twisted Pair (UTP): A generic term for "telephone" wire used to carry data such as 10Base-T and 100Base-T. Various categories (qualities) of cable exist that are certified for different kinds of networking technologies.

Update: The file processing activity in which master records are altered to reflect the current business activity contained in transactional files.

USENET: A facility of the Internet, also called "the news," that allows users to read and post messages to thousands of discussion groups on various topics.

User Datagram Protocol (UDP): A transport protocol in the Internet suite of protocols. UDP, like TCP, uses IP for delivery; however, unlike TCP, UDP provides for exchange of datagrams without acknowledgments or guaranteed delivery.

Validation: The determination of the correctness, with respect to the user needs and requirements, of the final program or software produced from a development project.

Validation, Verification, and Testing: Used as an entity to define a procedure of review, analysis, and testing throughout the software life cycle to discover errors; the process of validation, verification, and testing determines that functions operate as specified and ensures the production of quality software.

Value-Added Network (VAN): A communications network using existing common carrier networks and providing such additional features as message switching and protocol handling.

Verification: (1) The authentication process by which the biometric system matches a captured biometric against the person's stored template. (2) The demonstration of consistency, completeness, and correctness of the software at and between each stage of the development life cycle.

Virtual Circuit: A network service that provides connection-oriented service, regardless of the underlying network structure.

Virtual Memory: A method of extending computer memory using secondary storage devices to store program pages that are not being executed at the time.

Virtual Private Network (VPN): A VPN allows IP traffic to travel securely over public TCP/IP network by encrypting all traffic from one network to another. A VPN uses "tunneling" to encrypt all information at the IP level.

Virus: A type of malicious software that can destroy the computer's hard drive, files, and programs in memory, and that replicates itself to other disks.

WAN (Wide Area Network): Data communications network that serves users across a broad geographic area and often uses transmission devices provided by common carriers. Frame Relay, SMDS, and X.25 are examples of WANs. *Compare with* LAN and MAN.

Waterfall Life Cycle: A software development process that structures the analysis, design, programming, and testing. Each step is completed before the next step begins.

Web Crawler: A software program that searches the Web for specified purposes such as to find a list of all URLs within a particular site.

Whois: An Internet resource that permits users to initiate queries to a database containing information on users, hosts, networks, and domains.

Wiring Closet: Specially designed room used for wiring a data or voice network. Wiring closets serve as a central junction point for the wiring and wiring equipment that is used for interconnecting devices.

Work Factor: The effort and time required to break a protective measure.

X.400: ITU-T recommendation specifying a standard for email transfer.

X.500: The CITT and ISO standard for electronic directory services.

XDSL: A group term used to refer to ADSL (Asymmetrical Digital Subscriber Line), HDSL (High data rate Digital Subscriber Line), and SDSL (Symmetrical Digital Subscriber Line). All are digital technologies using the existing copper infrastructure provided by the telephone companies. XDSL is a high-speed alternative to ISDN.